



---

**Universidad DE Valladolid**

Escuela Técnica Superior de Ingenieros de Telecomunicación  
Dpto. de Teoría de la Señal y Comunicaciones e Ing. Telemática

**Tesis Doctoral**

**Simulador Web de Resonancia  
Magnética como Herramienta  
Educativa: Diseño, Desarrollo y  
Evaluación.**

Presentada por **Daniel Treceño Fernández** para optar al grado de doctor por la  
Universidad de Valladolid

Dirigida por:  
**Carlos Alberola López y Rodrigo de Luis García**

4 de octubre de 2019  
Valladolid, España









# AGRADECIMIENTOS

---

Me gustaría comenzar dando las gracias a todos los docentes y profesionales que han contribuido al desarrollo de esta tesis. En primer lugar, a mis tutores, los doctores Carlos Alberola y Rodrigo de Luis, que me han mostrado su apoyo, su paciencia y su sacrificio. También a los doctores Eduardo Gómez y Miguel L. Bote por prestarme su consejo y su dirección en los aspectos educativos y experimentales de este trabajo. Así mismo, no debo olvidar a la SERAM, y especialmente a los doctores Juan Calabia y Fátima Matute por su predisposición en el desarrollo y prueba del simulador. Y, de igual modo, mencionar a mi tutor en la CMU, Jonathan Sewall por su dedicación durante la incorporación del ITS a nuestro simulador de resonancia.

Sin embargo, esta labor no habría sido posible sin mis compañeros de laboratorio. Gracias doctor Iñaki Rabanillo por las largas conversaciones, las horas de trabajo dedicadas y por tu ayuda. De la misma manera, gracias al futuro doctor Santiago Sanz junto al que he pasado largo tiempo discutiendo conceptos. Con especial cariño quiero agradecer a Elena Martín, Elisa Moya y Óscar Peña por los buenos momentos vividos. Sin olvidar a todos los demás miembros del LPI que también han formado parte de este proceso Javier Royuela, Diego Martín, Federico Simmross, Santiago Aja, Susana Merino, Álvaro Planchuelo, Alejandro Godino, Rosa María Menchón, Pablo Casaseca, Marcos Martín, Gonzalo Barrio, Ariel Hernán, Gonzalo Vegas y Antonio Tristán.

No menos importantes han sido la personas más cercanas, todos mis amigos y, especialmente Laura, mi mujer, que me han brindado su apoyo, su ánimo y su comprensión en los momentos de mayor dificultad en estos 5 años. De igual modo, gracias a mi familia, a mis padres Candelas y Luis y a mi hermano Álvaro por su presencia, su cariño y su respaldo.

Finalmente, quiero agradecer la financiación de la compañía Giveme5D y del Fondo Social Europeo, Programa Operativo de Castilla y León, y de la Junta de Castilla y León, a través de la Consejería de Educación.



# ABSTRACT

---

Magnetic resonance imaging (MRI) is a medical modality with enormous popularity. This is due to several reasons, such as the high contrast it provides in soft tissues, the fact that it makes use of non-ionizing radiation and its high versatility since different contrasts can be obtained by proper parameter tuning. This has the direct consequence that this modality turns out to be complex and so is the learning process of radiographers.

In response to such a complexity, this thesis focuses on the design, development and evaluation of a magnetic resonance web simulator that serves for training in both theoretical concepts and practice of MRI.

Software engineering is used for its development, where the different steps to be carried out in a software project have been followed. Specifically, the requirements that the tool must fulfill have been compiled by a group of experts. In addition, an intelligent tutoring system (ITS) has been incorporated in the tool. To the best of our knowledge, this is the first time that an ITS is integrated in an MR simulator.

The simulator has been evaluated by means of three educational experiences. We have measured both user experience as well as the actual educational value of the tool. The latter has been measured by means of a one-day actual class-room experience.

In terms of user experience, our results evolved positively during the simulation evaluations. In particular, this feature improved significantly with the incorporation of an ITS. User comments supported the quantitative measurements obtained. Results from the class-room experience showed statistical differences in favor of the participants who used the simulator both in effect size and in inferential tests. We are not aware that an evaluation of the actual educational value of an MR simulator has ever been reported.

---

**Keywords:** Magnetic Resonance Imaging (MRI); Simulator System; Medical training/educational tool; System Usability Scale (SUS); Instructional design; Radiographers training; Intelligent Tutoring System (ITS)



# RESUMEN

---

La imagen por resonancia magnética (MRI) es una modalidad médica con una gran popularidad. Esto se debe a varias causas, tales como la consecución de imágenes de gran contraste en tejidos blandos, el hecho de que utiliza radiación no ionizante y su alta versatilidad, ya que puede dar lugar a muy diversos contrastes mediante el ajuste adecuado de parámetros. Esto hace que la resonancia magnética sea una modalidad compleja y, en consecuencia, el proceso de aprendizaje de los técnicos de imagen para el radiodiagnóstico también lo es.

En respuesta a tal complejidad, esta tesis se centra en el diseño, desarrollo y evaluación de un simulador web de resonancia magnética que sirve tanto para el aprendizaje de conceptos teóricos sobre resonancia magnética como para la capacitación en la adquisición de esta modalidad.

Para su desarrollo se hace uso de ingeniería del software, donde se establecen los pasos que deben llevarse a cabo en una implementación de este carácter. Específicamente, se reúne a un grupo de expertos para recopilar el conjunto de requisitos que debe cumplir esta herramienta y que serán claves para su desempeño. Además, se ha incorporado un sistema de tutoría inteligente (ITS) a esta herramienta. Hasta donde conocemos, ésta es la primera vez que se integra un ITS en un simulador de resonancia magnética.

El simulador ha sido evaluado a través de tres experiencias educativas, donde se ha medido tanto la experiencia del usuario como el valor educativo real de la herramienta. Esta última medición se llevó a cabo durante la realización de un experiencia real en una clase y que tuvo una duración de un día.

En términos de la experiencia del usuario, nuestros resultados evolucionaron positivamente durante las evaluaciones del simulador. En particular, esta característica mejoró notablemente con la incorporación de un ITS. Los comentarios de los usuarios respaldaron las mediciones cuantitativas obtenidas. Los resultados de la experiencia en el aula mostraron diferencias estadísticas significativas a favor de los participantes que usaron el simulador, tanto en el tamaño del efecto como en las pruebas inferenciales. No somos conscientes de publicación científica alguna en la que se haya descrito la medición del valor educativo real de un simulador de resonancia magnética.

---

**Palabras clave:** Imagen por resonancia magnética (MRI); Sistema de simulación; Herramienta educativa y para la instrucción médica; System Usability Scale (SUS); Diseño experimental; Formación de técnicos en imagen para el diagnóstico y medicina nuclear; Intelligent Tutoring System (ITS)



# TABLA DE CONTENIDO

---

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.1.1. Planteamiento del problema . . . . .	2
1.2. Objetivos . . . . .	3
1.3. Metodología . . . . .	5
1.4. Publicaciones . . . . .	8
1.5. Estructura de la tesis . . . . .	8
<b>2. Introducción a la resonancia magnética</b>	<b>11</b>
2.1. Imagen por resonancia magnética . . . . .	11
2.1.1. Escáner de MRI . . . . .	13
2.1.2. Proceso de adquisición . . . . .	14
2.2. Fundamentos de MRI . . . . .	16
2.2.1. Localización de la señal . . . . .	20
2.2.2. Secuencia de pulsos . . . . .	22
2.2.3. Artefactos . . . . .	25
<b>3. Simuladores de resonancia magnética</b>	<b>29</b>
3.1. Introducción . . . . .	29
3.2. Simuladores educativos . . . . .	30
3.2.1. Bloch Simulator . . . . .	30
3.2.2. Simplified MRI . . . . .	31
3.2.3. Virtual MRI . . . . .	32
3.2.4. Simulador de Torheim . . . . .	34
3.3. Simuladores en la investigación . . . . .	35
3.3.1. MRiLab . . . . .	35
3.3.2. JEMRIS . . . . .	37
3.3.3. SIMRI . . . . .	39
3.3.4. MRISIMUL . . . . .	39
3.3.5. PSUdoMRI . . . . .	40
3.3.6. SpinBench . . . . .	42
3.3.7. BlochSolver . . . . .	42
3.3.8. POSSUM . . . . .	43
3.3.9. ODIN . . . . .	45
3.4. Discusión y Conclusión . . . . .	47

<b>4. Sistemas de tutoría inteligente (ITS)</b>	<b>51</b>
4.1. Introducción . . . . .	51
4.2. Arquitectura de un ITS . . . . .	53
4.3. Diseño de un ITS. Modelos cognitivos vs. sistemas expertos . . . . .	55
4.4. Herramientas de creación de un ITS . . . . .	57
4.4.1. Justificación de su necesidad . . . . .	57
4.4.2. Descripción de las principales herramientas . . . . .	57
4.5. Discusión y conclusiones . . . . .	61
<b>5. Desarrollo del simulador</b>	<b>63</b>
5.1. Introducción . . . . .	63
5.2. Análisis . . . . .	63
5.3. Diseño . . . . .	66
5.3.1. Diseño del sistema . . . . .	66
5.3.2. Diseño del programa . . . . .	69
5.4. Implementación . . . . .	78
5.4.1. Paneles desplegables . . . . .	79
5.4.2. Wrapper . . . . .	83
5.4.3. Secuencias de pulsos . . . . .	85
5.4.4. Imágenes generadas . . . . .	87
5.5. Resultados . . . . .	89
5.6. Conclusiones . . . . .	92
<b>6. Inclusión de un ITS en el simulador</b>	<b>93</b>
6.1. Introducción . . . . .	93
6.1.1. Elección de una herramienta de tutoría . . . . .	93
6.1.2. Detalles adicionales sobre CTAT . . . . .	94
6.2. Análisis . . . . .	96
6.3. Diseño . . . . .	97
6.3.1. Diseño del sistema . . . . .	98
6.3.2. Diseño del programa . . . . .	98
6.4. Implementación . . . . .	102
6.4.1. Servicio . . . . .	102
6.4.2. Directivas . . . . .	106
6.5. Resultados . . . . .	111
6.6. Conclusiones . . . . .	114
<b>7. Experimentación</b>	<b>115</b>
7.1. Introducción . . . . .	115
7.2. Evolución del simulador . . . . .	116
7.2.1. Versión 1 . . . . .	116
7.2.2. Versión 2 . . . . .	118
7.2.3. Versión 3 . . . . .	118
7.2.4. Versión 4 . . . . .	119
7.3. Estudio sobre la experiencia del usuario . . . . .	119



7.3.1. Metodología . . . . .	120
7.3.2. Resultados . . . . .	123
7.3.3. Discusión . . . . .	124
7.4. Estudio sobre la utilidad educativa . . . . .	127
7.4.1. Metodología . . . . .	128
7.4.2. Resultados . . . . .	131
7.4.3. Discusión . . . . .	132
7.5. Conclusión . . . . .	133
<b>8. Conclusiones, contribuciones y líneas futuras</b>	<b>135</b>
8.1. Conclusiones . . . . .	135
8.2. Contribuciones . . . . .	137
8.3. Limitaciones . . . . .	137
8.4. Líneas futuras . . . . .	138
<b>A. CASEIB 2016</b>	<b>141</b>
<b>B. AngularJS</b>	<b>147</b>
B.1. Introducción . . . . .	147
B.2. Modularidad y patrón de diseño . . . . .	147
B.2.1. MVC . . . . .	148
B.2.2. Dependencia Inyección . . . . .	148
B.3. Controlador en AngularJS . . . . .	148
B.4. Servicios en AngularJS . . . . .	149
B.5. Directivas en AngularJS . . . . .	151
B.6. Filtros en AngularJS . . . . .	152
<b>C. Planificación espacial: Análisis, diseño e implementación.</b>	<b>155</b>
C.1. Introducción . . . . .	155
C.2. Análisis . . . . .	155
C.3. Diseño . . . . .	157
C.4. Detalles de implementación . . . . .	159
<b>D. Instrumento de medida</b>	<b>163</b>
<b>E. Manual del simulador</b>	<b>169</b>
<b>F. Creación de un ejercicio con CTAT</b>	<b>205</b>



## ACRÓNIMOS Y NOTACIÓN

---

$TE_{eff}$  tiempo de eco efectivo

$T_1$  tiempo de relajación longitudinal

$T_2$  tiempo de decaimiento transversal

$\gamma$  relación giromagnética

$\omega$  frecuencia de resonancia o frecuencia de Larmor

$\omega_0$  frecuencia de resonancia o frecuencia de Larmor ( $\omega$ ) para un campo magnético principal ( $\vec{B}_0$ ) dado

$\tau_p$  tiempo de aplicación de pulso radiofrecuencia (RF)

$\vec{B}_0$  campo magnético principal

$\vec{B}_1$  pulso de RF

$\vec{M}$  vector de magnetización

**AB** basado en el agente o Agent-Based

**ADC** coeficiente de difusión aparente o Apparent Diffusion Coefficient

**AI** inteligencia artificial o Artificial-Intelligence

**AIED** inteligencia artificial en educación o Artificial Intelligence in EDucation

**ALM** modelo de aprendizaje del ejército de EE.UU. o U.S. Army Learning Model

**API** interfaz de programación de aplicaciones o Application Programming Interface

**APLUS** Artificial Peer Learning environment Using SimStudent

**ARL** laboratorio de investigación del ejército de EE.UU. o U.S. Army Research Laboratory

**ASAT** AutoTutor Script Authoring Tool

**ASATA** AutoTutor Script Authoring Tool for Assessment

**ASPIRE** Authoring Software Platform for Intelligent Resources in Education

**AT** herramientas de creación o Authoring Tools

**BLOD** dependiente del nivel de oxigenación sanguínea o Blood Oxygenation Level Dependent

**BRD** Behavior Recorder graph

**BW** ancho de banda o BandWidth

**CAI** *instrucción asistida por ordenador o Computer-Assisted Instruction*

**CAT** *CourseAuthoring Tool*

**CB** *basado en restricciones o Constraint-Based*

**CG** *grupo control*

**CPU** *unidad de procesamiento central o Central Processing Unit*

**CSF** *líquido cefalorraquídeo o CerebroSpinal Fluid*

**CSS** *Cascading Style Sheets*

**CT** *tomografía computerizada o Computed Tomography*

**CTAI** *Cognitive Tutor Algebra I*

**CTAT** *Cognitive Tutor Authoring Tools*

**DAT** *Domain Knowledge File Authoring Tool*

**DB** *basado en el diálogo o Dialogue-Based*

**DI** *inyección de dependencias o Dependency Injection*

**DICOM** *Digital Imaging and Communication On Medicine*

**dll** *bibliotecas de enlaces dinámicos o Dynamic Link Library*

**DOM** *Document Object Model*

**DRY** *no te repitas o Don't Repeat Yourself*

**DSE** *Dual Spin Echo*

**DWI** *diffusion weighted imaging*

**EG** *grupo experimental*

**EPI** *Echo-Planar imaging*

**ESP** *espacio entre ecos*

**ET** *seguimiento de ejemplo o Example-Tracing*

**ETL** *longitud del tren de eco o Echo Train Length*

**FA** *ángulo de giro o Flip Angle*

**FID** *Free Induction Decay*

**FISP** *Fast Imaging with Steady-state Precession*

- fMRI** *resonancia magnética funcional o functional magnetic resonance*
- FOV** *campo de visión o Field Of View*
- FR** *requisitos funcionales*
- FSE** *Fast Spin Echo*
- FSL** *biblioteca de software FMRIB's o FMRIB's Software Library*
- GIFT** *Generalized Intelligent Framework for Tutoring*
- GPU** *unidad de procesamiento gráfico o Graphic Processor Unit*
- GRE** *Gradient Echo*
- GUI** *interfaz gráfica de usuario o Graphical User Interfacel*
- HCI** *interacción persona-ordenador o Human-Computer Interaction*
- HTML** *HyperText Markup Language*
- ICAI** *aprendizaje inteligente asistido por ordenador o Intelligent Computer-Assisted Learning*
- IFT** *transformada inversa de Fourier o Inverse Fourier Transform*
- IJAIED** *revista internacional de inteligencia artificial en educación o International Journal of Artificial Intelligence in EDucation*
- IRSE** *Inversion Recovery Spin-Echo*
- ISO** *Organización Internacional de Normalización o International Standards Organization*
- ITK** *Insight segmentation and registration ToolKit*
- ITS** *sistemas de tutoría inteligente o Intelligent Tutoring Systems*
- JEMRIS** *Jülich Extensible MRI Simulator*
- JRE** *Java Runtime Environment*
- JS** *JavaScript*
- LMS** *sistema de gestión de alumnos o Learner Management System*
- LTR** *probabilidad de recomendar o LiKelihood to Recommend*
- MR** *resonancia magnética o Magnetic Resonance*
- MRI** *imagen por resonancia magnética o Magnetic Resonance Imaging*

- MT** *seguimiento de modelo o Model-Tracing*
- MVC** *modelo vista controlador o Model-View-Controller*
- MVVM** *modelo vista modelo de vista o Model-View-ViewModel*
- NEX** *numero de excitaciones o Number of EXcitations*
- NFR** *requisitos no funcionales*
- NIFTI** *Neuroimaging Informatics Technology Initiative*
- NMR** *resonancia magnética nuclear o Nuclear Magnetic Resonance*
- NPW** *No Phase Wrap*
- ODIN** *interfaz de desarrollo orientada a objetos para NMR o Object-Oriented Development Interface for NMR*
- ORM** *mapeo objeto-relacional o Object-Relational Mapping*
- PACS** *sistema de archivo y comunicación de imágenes o Picture Archiving and Communication System*
- PCAT** *Pedagogy Configuration Authoring Tool*
- PD** *densidad de protones o Proton Density*
- PET** *tomografía de emisión de positrones o Positron Emission Tomography*
- PF** *Partial Fourier*
- POSSUM** *escáner simulado orientado a la física para comprender la resonancia magnética o Physics-Oriented Simulated Scanner for Understanding MRI*
- ppm** *partes por millón*
- px** *pixel*
- RARE** *Rapid Acquisition with Relaxation Enhancement*
- RAS** *derecha-anterior-superior o Right-Anterior-Superior*
- RB** *basado en reglas o Rule-Based*
- REST** *transferencia de estado representacional o Representational State Transfer*
- RF** *radiofrecuencia*
- ROI** *región de interés o Region Of Interest*

**SAI** *selección, acción, entrada o Selection, Action, Input*

**SAR** *tasa de absorción específica o Specific Absorption Rate*

**SAS** *Survey Authoring System*

**SASS** *Syntactically Awesome Style Sheets*

**SDK** *Cognitive Model Software Development Kit*

**SDLC** *ciclo de vida del desarrollo de software o Software Development Life Cycle*

**SE** *Spin Echo*

**SERAM** *Sociedad Española de Radiología Médica*

**SIMILAR** *modelos de información de estudiantes para entornos de aprendizaje inteligentes o Student Information Models for Intelligent Learning Environments*

**SitPed** *Situated Pedagogical*

**SNR** *relación de señal a ruido o signal-to-noise ratio*

**so** *objetos compartidos o Shared Objects*

**SOA** *arquitectura orientada a servicios o Service-Oriented Architecture*

**SOAP** *protocolo simple de acceso a objetos o Simple Object Access Protocol*

**SP** *Scan Percentage*

**SPIR** *Spectral Presaturation with Inversion Recovery*

**Spoiled GRE** *Spoiled Gradient Echo*

**SQL** *Structured Query Language*

**SRS** *especificación de los requisitos del programa o Software Requirements Specification*

**SSFP** *Steady State Free Precession*

**SUS** *escala de usabilidad del sistema o System Usability Scale*

**TDK** *Tutorial Development Kit*

**TE** *tiempo de eco*

**TI** *tiempo de inversión*

**TR** *tiempo de repetición*

**URL** *Uniform Resource Locator*

**UX** *experiencia del usuario o User eXperience*

**VTK** *Visualization ToolKit*

**WBE** *educación basada en la web o Web-Based Education*

**XML** *lenguaje de marcas extensible o Extensible Markup Language*

**xPST** *Extensible Problem-Solving Tutor*

**XTK** *X ToolKit*



## Capítulo 1

# INTRODUCCIÓN

---

EN este capítulo introductorio se expone la motivación que ha dado lugar a la presente tesis así como los objetivos perseguidos. A continuación se describe la metodología empleada, detallando cada uno de los procesos seguidos por el doctorando y los materiales a los que ha tenido acceso. Por último, se incluyen las contribuciones derivadas del trabajo realizado así como un esquema de la estructura general de la tesis.

## 1.1 MOTIVACIÓN

---

La *imagen por resonancia magnética* o *Magnetic Resonance Imaging* (MRI) es una de las modalidades más usadas en el área de la radiología en la actualidad. Está basada en el fenómeno de *resonancia magnética nuclear* o *Nuclear Magnetic Resonance* (NMR), que fue descrito en los años 40 (Bloch, 1946) del siglo pasado. Sin embargo, su aplicación en imagen médica no fue desarrollada hasta principios de los años 70 (Damadian, 1971). Esta tecnología hace uso de fuertes campos magnéticos junto a gradientes de campo y señales de radiofrecuencia para generar imágenes internas del cuerpo. Esto la convierte en una técnica no invasiva. Adicionalmente, a diferencia de la *tomografía computerizada* o *Computed Tomography* (CT) y la *tomografía de emisión de positrones* o *Positron Emission Tomography* (PET), se trata de una modalidad no ionizante, ya que no altera la estructura molecular. Pero, por encima de todo, sus imágenes destacan por aportar un gran nivel de información sobre datos químicos y físicos de los tejidos, convirtiendo esta modalidad en una herramienta diagnóstica de gran versatilidad.

Desafortunadamente, esta flexibilidad conlleva un coste. El manejo adecuado de una máquina de *resonancia magnética* o *Magnetic Resonance* (MR) es una tarea compleja por varias razones. En primer lugar, la física que subyace en el fenómeno de la NMR y las técnicas usadas para la formación de las imágenes no resultan intuitivas y, por tanto, son difíciles de aprender. Esto se ve agravado por el hecho de que los usuarios habituales, los técnicos en imagen para el radiodiagnóstico que operan el escáner de resonancia, generalmente no cuentan con conocimientos avanzados en física y en matemáticas. En segundo lugar, existe una enorme cantidad de parámetros asociados a una adquisición de MR, los cuales repercuten muy directamente en la calidad de la imagen, su contraste o la duración de la exploración. Además, cada fabricante utiliza su propio software y nomenclatura, por lo que hay que añadir las cuestiones relacionadas con enfrentarse a un sistema desconocido. Esta adaptación a las particularidades del sistema ha

demostrado ser problemática incluso para radiólogos experimentados (Jorritsma et al., 2016). En cuarto lugar, el coste de estos equipos es muy elevado y tienen una alta tasa de ocupación dado su uso intensivo en la práctica clínica. Debido a ello, la práctica *in situ* de los estudiantes es muy escasa. Finalmente, a estas incidencias hay que añadir la gran cantidad de experiencia necesaria para analizar la imagen, donde la presencia de artefactos podría afectar críticamente al diagnóstico (Erasmus et al., 2004).

Por estas razones, una alternativa de interés para la formación de técnicos en imagen para el radiodiagnóstico sería la disposición de un sistema informático que reprodujese el comportamiento de una interfaz de una consola real de MR y que simulase la física del proceso en la generación de una imagen. A lo largo de este documento, dichos técnicos va a ser también referidos como técnicos radiólogos, técnicos en imagen o, simplemente, técnicos.

### 1.1.1 PLANTEAMIENTO DEL PROBLEMA

Internet ha supuesto un cambio equiparable a la revolución industrial en nuestra sociedad Leiner et al. (2009). Puede observarse tanto a nivel individual como colectivo y está suponiendo una evolución notable en infraestructuras y equipamiento tanto en el sector privado como en el público. Su acceso generalizado ha democratizado la información, que ha pasado a estar disponible en cualquier momento y en cualquier lugar, creando grandes oportunidades en el mundo del aprendizaje. Como consecuencia, ha surgido una nueva generación de estudiantes que buscan una educación personalizada, accesible, práctica y deslocalizada. La *educación basada en la web o Web-Based Education* (WBE), surge como respuesta a esta demanda (Peredo et al., 2011) y también está presente en la educación aplicada al campo de la medicina (Papachristos and Alafodimos, 2011). Sin embargo, aunque esta tecnología suple ciertas necesidades del estudiante con respecto a la educación tradicional, existen problemas que son comunes a ambas.

Las herramientas educativas sirven para rellenar los espacios existentes entre las técnicas convencionales de educación y las dificultades que ofrece la interacción de un estudiante con un sistema complejo. Dichas técnicas, de forma paralela a la WBE y complementando al aprendizaje puramente teórico, han tenido un crecimiento exponencial gracias al desarrollo tecnológico. En el ámbito médico, un claro ejemplo se observa en los simuladores quirúrgicos, donde los sistemas que utilizan los cirujanos para la realización de operaciones adolecen de un elevado nivel de complejidad. Por otro lado, es interesante distinguir entre un simulador que trata de reproducir con exactitud un fenómeno que ocurre en el mundo real y un dispositivo de entrenamiento, donde el sistema presenta estímulos y oportunidades de práctica necesarios para mejorar el nivel de aprendizaje del alumno. Por tanto, los simuladores son útiles para las prácticas de aprendices y expertos, pero no necesariamente se traducen en el desarrollo de habilidades (Andrews, 1988). En otras palabras, un simulador no tiene necesariamente que ser un dispositivo de entrenamiento. Manteniendo el ejemplo anterior, lo que dota de interés al simulador quirúrgico es el hecho de que la simulación por ordenador se complementa con el aparataje que un cirujano usa realmente durante la operación y que le permite adquirir habilidades prácticas reales.

Una herramienta educativa puede, además, brindar servicios automáticos de tutoría que favorezcan el aprendizaje. Esta idea, de forma análoga a la educación y sus herramientas, ha ido evolucionando notablemente. En los años 70 comienza su concepción, donde a la *instrucción asistida por ordenador o Computer-Assisted Instruction* (CAI) se le suma un sistema de *in-*

*teligencia artificial o Artificial-Intelligence* (AI) (Carbonell, 1971) y forman el precedente al término de *aprendizaje inteligente asistido por ordenador o Intelligent Computer-Assisted Learning* (ICAI). Pero no fue hasta más adelante cuando Sleeman and Brown (1982) acuñaron el término de *sistemas de tutoría inteligente o Intelligent Tutoring Systems* (ITS) tal y como hoy se conocen y que, además, da nombre a la conferencia bianual que se lleva celebrando desde 1988 hasta hoy en día. En suma, este sistema se ha venido utilizando para el aprendizaje en áreas tan diversas como matemáticas, biología, física o incluso en simulaciones para el aprendizaje del pilotaje de helicópteros, con notables resultados en los tamaños de efecto (Torchiano, 2018) ( $>0.5$ ) así como en opiniones favorables de los estudiantes.

En Radiología, y más específicamente en el manejo de instrumental radiológico como un escáner de MRI, el técnico toma parte en la preparación del paciente y también en el mantenimiento y la seguridad. No obstante, su labor principal consiste en interacciones con la consola que opera el escáner. Esta tarea abarca el posicionamiento del paciente en la máquina, la selección de la bobina a usar, la selección de los protocolos de imagen y la determinación de los parámetros de captación. Intuitivamente, la simulación puede dar respuesta a cómo la alteración de dichos parámetros repercute en la calidad final de la imagen. Actualmente existen varios simuladores de resonancia magnética, que pueden clasificarse por el nivel de detalle que sigue el modelo físico que utilizan; existen modelos simples y que han sido utilizados en educación o modelos mucho más detallados que han sido enfocados principalmente al mundo investigador. Sin embargo, la mayor parte de ellos no pueden ser considerados dispositivos de entrenamiento y, dadas sus arquitecturas, no sería inmediato incluir estos simuladores ni en la enseñanza tradicional ni en la WBE.

En resumen, el uso de una herramienta educativa flexible en su disponibilidad y su uso, y que sirva para la adquisición de conocimientos y destrezas sobre adquisición de resonancia magnética, sería de utilidad como ayuda a la superación de las dificultades inherentes al aprendizaje de MR, pudiendo resultar interesante en la formación de técnicos en imagen y médicos radiólogos, tanto en su etapa de estudiantes como una vez titulados.

## 1.2 OBJETIVOS

---

En vista del gran protagonismo que hoy en día tiene la MR en la imagen médica y la complejidad que implica tanto su adquisición como su interpretación, el principal objetivo de esta tesis es *el diseño, desarrollo y evaluación de un simulador de resonancia magnética como herramienta educativa*. Su principal función será posibilitar al alumno la adquisición de destrezas y el desarrollo de competencias de forma similar a como lo haría si estuviese sentado ante la consola del equipo real. Con este fin, los objetivos específicos que se derivan son:

1. Diseñar el simulador como un dispositivo de entrenamiento. El simulador debe posibilitar que el estudiante desarrolle las habilidades necesarias para el manejo de una máquina de MR. En otras palabras, debe permitir que un estudiante pueda reproducir el proceso completo que un técnico realiza de forma habitual sobre estos equipos. A su vez, debe estar dotado de la versatilidad necesaria para su uso en distintos modelos educativos, es decir, debe poderse utilizar como complemento o herramienta educativa tanto en instrucción tradicional como en la WBE.

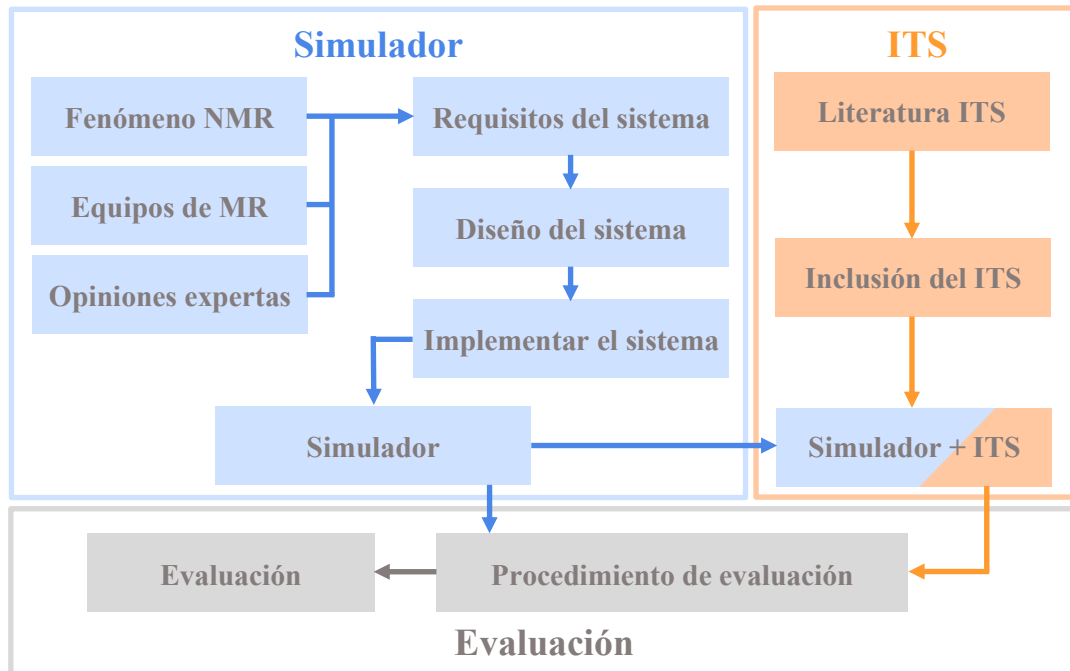


FIGURA 1.1: Objetivos de la tesis

2. Dotar al simulador de un ITS que pueda guiar al alumno y proporcionarle información acorde con los pasos realizados.
3. Demostrar que un simulador diseñado según el primer objetivo, y que incorpore un ITS, es una herramienta de utilidad educativa. Distintos aspectos del simulador deben ser valorados, abarcando ambas versiones, esto es, tanto la versión simple como la versión con sistema de tutoría incorporado.

La consecución de estos objetivos específicos lleva a su vez implícito una serie de hitos u objetivos secundarios a alcanzar:

- Familiarizarse con los equipos de MR.
- Recoger opiniones y comentarios sobre el personal experto que manipula los escáneres de MR; esto incluye tanto a radiólogos como a técnicos.
- Familiarizarse con la física que modela el fenómeno de NMR.
- Establecer los requisitos del sistema usando la información recogida.
- Diseñar el simulador a partir de los requisitos especificados y los datos recogidos.
- Implementar y probar el simulador.
- Crear un procedimiento —o utilizar un procedimiento estándar— para la evaluación del simulador. Dicho procedimiento deberá evaluar las propiedades educativas del simulador

tanto en términos de utilidad como de uso. Las medidas a tener en cuenta podrán ser cualitativas y/o cuantitativas.

- Familiarizarse con los ITS.
- Incorporar un ITS al simulador.
- Crear un procedimiento —o utilizar un procedimiento estándar— para la evaluación del simulador con el ITS. En la evaluación del simulador, las medidas a tener en cuenta podrán ser cualitativas y/o cuantitativas.

De forma abreviada, el diagrama de la Figura 1.1 esquematiza estos puntos. El primer objetivo específico viene representado por el color azul, el segundo hace uso de tonos naranjas, mientras que el tercero utiliza matices neutros. En el interior de cada objetivo específico se desglosan los objetivos secundarios.

## 1.3 METODOLOGÍA

---

La metodología que se ha seguido ha estado encaminada a la consecución de los diversos objetivos planteados en la sección anterior. En primer lugar, se han acometido los estudios necesarios para crear un diseño aproximado del simulador. A continuación, se ha llevado a cabo su desarrollo. Para terminar, se ha hecho un esfuerzo importante para evaluar la herramienta en escenarios educativos reales.

Desde la perspectiva de desarrollo software, debemos destacar el concepto del *ciclo de vida del desarrollo de software* o *Software Development Life Cycle* (SDLC), el cual está ampliamente detallado en la literatura (Mall, 2013). En esta tesis, hemos optado por adaptar el modelo incremental (Larman and Basili, 2003). De igual forma, para su representación hemos hecho uso del esquema del modelo espiral (Boehm, 1986), el cual como puede verse en la Figura 1.2.

El esquema debe leerse desde el interior, siguiendo el sentido de la espiral, hasta su parte más externa, pasando por los distintos sectores:

1. El **estudio inicial** del estado del arte establece el marco teórico que contextualiza la presente tesis. Se divide en las siguientes áreas:
  - **Estudio de los fundamentos de MR y operatividad de una máquina.** Abarca los conceptos generales de la resonancia (Zhi-Pei Liang, 2000), haciendo además hincapié en el proceso que un técnico realiza con este tipo de escáneres (Elmaoglu and Çelik, 2012).
  - **Estudio del estado del arte de simuladores de MR.** Este campo ha avanzado notablemente en los últimos años y abarca desde simuladores básicos (Torheim et al., 1994; Hackländer and Mertens, 2005) a simuladores de mucha mayor complejidad (Benoit-Cattin et al., 2005; Xanthis et al., 2013; Kose and Kose, 2017).
  - **Estudio de los ITS.** Este área es extensa y nos centraremos principalmente en aquellas herramientas que permitan una creación rápida de un ITS.

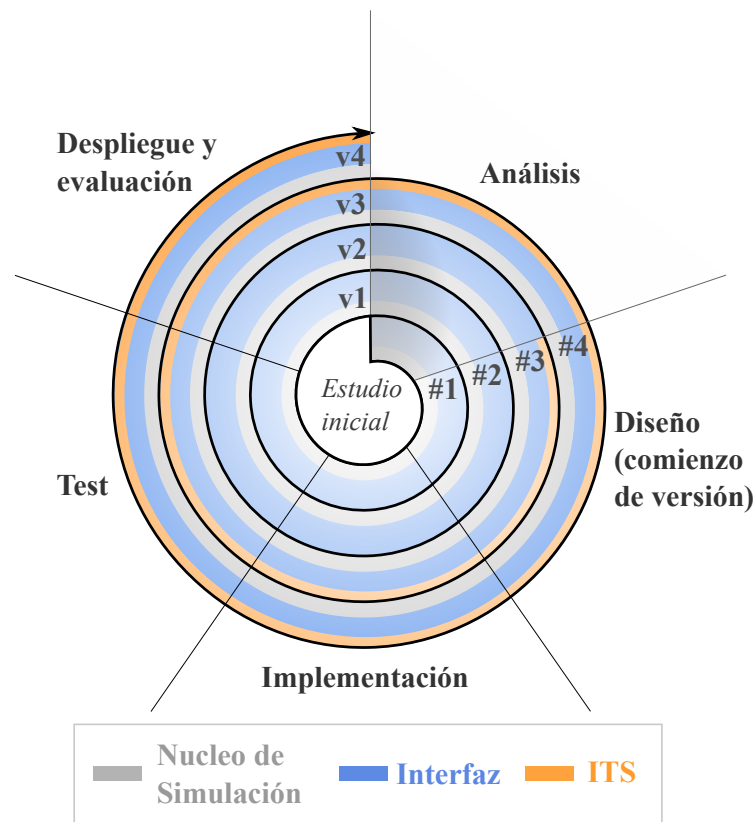


FIGURA 1.2: Esquema de la metodología seguida. Se utiliza un modelo incremental representado como una espiral donde se muestran las diferentes fases del desarrollo del software.

2. **Análisis.** Su objetivo es fijar los requerimientos que el simulador de resonancia debe cumplir. Se divide en dos fases:
  - **Requisitos.** El simulador se ha desarrollado con la colaboración de personal experto perteneciente a la *Sociedad Española de Radiología Médica* (SERAM). En la interacción con estos profesionales se han identificado una serie de requisitos que el simulador debería cumplir y que deben ser analizados para comprobar su viabilidad. Estos requisitos pueden diferenciarse entre requisitos globales, que se mantendrían a lo largo del todo el desarrollo, y requisitos específicos para cada versión que se desarrolle.
  - **Especificaciones.** Se crea un documento que concrete los requisitos que debe cumplir la versión. Éste es conocido como *especificación de los requisitos del programa o Software Requirements Specification* (SRS).
3. **Diseño.**
  - **Diseño del sistema.** Determina los componentes que deben constituir el sistema, es decir, el *hardware*, las herramientas de software que van a utilizarse, la arquitectura del sistema, las *frameworks*, etc.

- **Diseño del programa.** Define los distintos componentes software que van a componer el simulador así como los elementos en que se subdividen. Analizando la Figura 1.2 vemos que la espiral está compuesta por dos bandas en su inicio, aunque finaliza con tres. Cada una de estas bandas representa un componente principal del simulador. Desde dentro hacia fuera de la espiral se observan el núcleo de simulación (banda de color gris), la interfaz de usuario del simulador (banda de color azul) y finalmente el ITS, el cual se incorpora en fases posteriores (banda de color naranja).
4. **Implementación**, acorde con el diseño del programa.
  5. **Test.** Determina si el simulador cumple con los requisitos y trata de encontrar errores que estén presentes en el código.
  6. **Despliegue y evaluación.**
    - **Despliegue.** Incluye todos los procesos necesarios para realizar un despliegue completo de la herramienta y que ésta pueda ser utilizada por los usuarios finales.
    - **Evaluación.** El proceso de evaluación debe ser lo más objetivo posible y por ende debe estar sujeto a una baja variabilidad. Por tanto, la evaluación debe hacerse mediante el procedimiento y con el instrumento adecuados. Esto es aplicable para cualquiera de las versiones, pudiendo cambiar igualmente en función del carácter de la prueba.
  7. **Escritura de la tesis.** Por último, el documento final de la tesis recoge la motivación y la metodología utilizadas, el marco teórico, el desarrollo del simulador, su evaluación y presenta todos los resultados obtenidos.

Una ventaja de usar esta representación en espiral es que la posición angular en la que se encuentre la espiral representa el estado en el que se encuentra el programa; la longitud del radio es representativa del tiempo invertido en el simulador. El número de rotaciones que realiza la espiral proporciona el número de versiones que ha tenido el simulador.

Finalmente, como estudiante del grupo *Laboratorio de Procesado de Imagen* (LPI) de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid, se ha contado con diversos medios para el desempeño del trabajo:

- **Hardware:** El grupo ha proporcionado acceso a diversidad de equipos (ordenadores portátiles, servidores, impresoras, etc.). Por otro lado, para la obtención de imágenes de MR, el grupo cuenta con acceso a un escáner MRI del Laboratorio de Técnicas Instrumentales de la Universidad de Valladolid.
- **Software:** El grupo posee licencias para programas con los que llevar a cabo el prototipado y las tareas de documentación y edición de textos.
- **Conocimiento:** Los integrantes del LPI cuentan con una reconocida experiencia en el análisis de imágenes médicas y, en particular, en la adquisición y procesado de resonancia magnética.

## 1.4 PUBLICACIONES

---

Durante la ejecución de esta tesis se han generado las siguientes publicaciones:

- *Publicación en congreso como exposición oral:* Treceño Fernández, D., Calabia del Campo, J., de Luis García, R., Alberola López, C., 2016. Una implementación eficiente no paralela de secuencias de resonancia magnética mediante matrices sparse. En XXXIV CASEIB. CASEIB, pp. 177–180.
- *Publicación en congreso como exposición oral:* Moya Sáez, E., Treceño Fernández, D., de Luis García, R., Simmross Wattenberg, F., Alberola López, C., 2017. Implementación paralela de secuencias de resonancia magnética mediante programación en gpu. En: XXXV CASEIB. CASEIB, pp. 29–32.
- *Aceptación en congreso como exposición oral:* Treceño Fernández, D., Calabia del Campo, J., Luna Alcalá, A., Bote Lorenzo, M., Gómez Sánchez, E., de Luis García, R., Alberola López, C., 2019. A web-based MRI simulator for radiographer education: Quantitative evaluation of an actual classroom experience. Annual conference of Radiological Society of North America (RSNA).
- *Aceptación en revista:* Treceño-Fernández, Daniel, Calabia-del-Campo, Juan, Bote-Lorenzo, Miguel L., Gómez-Sánchez, Eduardo , de-Luis-García, Rodrigo, Alberola-López, Carlos, 2019. A Web-based Educational Magnetic Resonance Simulator: Design, Implementation and Testing. En: Journal of Medical Systems.

## 1.5 ESTRUCTURA DE LA TESIS

---

El presente documento se estructura en ocho capítulos; del Capítulo 2 al 4 se introduce el marco teórico, los Capítulos 5 y 6 describen los métodos puestos en práctica para el desarrollo de la herramienta, el Capítulo 7 expone los experimentos realizados, y finalmente el capítulo 8 contiene las conclusiones. De forma más detallada:

En el primer capítulo del marco teórico, el Capítulo 2, se hace una breve introducción a la MR, poniendo énfasis en los procesos que el técnico radiólogo realiza. Asimismo, se introducen los fundamentos físicos que soportan esta tecnología, explicando cómo se genera la señal, algunas secuencias básicas o qué artefactos son comunes. A continuación, en el Capítulo 3 se realiza una breve introducción a los simuladores de resonancia magnética actuales. Después se seleccionan algunos de ellos para su análisis más detallado, mostrando sus puntos fuertes y débiles. Finalmente, se realiza una comparativa entre ellos y se extrae una serie de conclusiones. Posteriormente, y para finalizar el marco teórico, en el Capítulo 4 se realiza una introducción a los ITSs, justificando su necesidad y exponiendo las herramientas que permiten un rápido desarrollo de estos sistemas. Para terminar, se realiza una elaborada discusión sobre las herramientas analizadas.

Los métodos utilizados en el desarrollo del programa comienzan con el Capítulo 5 donde se describe el proceso completo de desarrollo, el análisis de requisitos, el diseño y las estrategias de



implementación de cada uno de los elementos principales, tanto de la interfaz como del núcleo que realiza la simulación. El capítulo finaliza analizando de forma retrospectiva las características de la herramienta creada. La incorporación de las *Cognitive Tutor Authoring Tools* (CTAT) se recoge en el Capítulo 6. Aquí se examina detalladamente el diseño de la tecnología y se crea un software intermedio que hace compatible nuestra herramienta con CTAT. Nuevamente, se comienza por un análisis y a continuación se realiza el diseño del módulo que es necesario implementar. Los resultados y las conclusiones de dicha implementación son recogidos a continuación. La evaluación de la herramienta se recoge en el Capítulo 7; el capítulo comienza con una breve recopilación de las versiones lanzadas, sus características y el contexto en el que fueron desplegadas. A continuación se relata la metodología usada para su evaluación, así como los resultados y las conclusiones obtenidas. La tesis concluye con el Capítulo 8, donde se reúnen las principales conclusiones derivadas de los desarrollos y resultados presentados en la tesis.



## Capítulo 2

# INTRODUCCIÓN A LA RESONANCIA MAGNÉTICA

---

**E**STE capítulo pretende realizar una revisión de los conceptos más importantes relacionados con el funcionamiento de la imagen por resonancia magnética, tanto desde el punto de vista del equipamiento como del procedimiento empleado para realizar una adquisición. Se continúa con una explicación de los fundamentos físicos del fenómeno de la resonancia magnética nuclear, y cómo este fenómeno se utiliza para generar y medir una señal que termina convirtiéndose en una imagen. Se describen brevemente los tipos de secuencia más básicos empleados en resonancia, y finalmente se hace una breve revisión de los artefactos de imagen más comunes en esta modalidad de imagen. Este contenido resulta de relevancia porque constituye las bases que serán empleadas para la núcleo de las simulaciones que deben ser realizadas para emular el proceso de adquisición de una imagen de resonancia magnética.

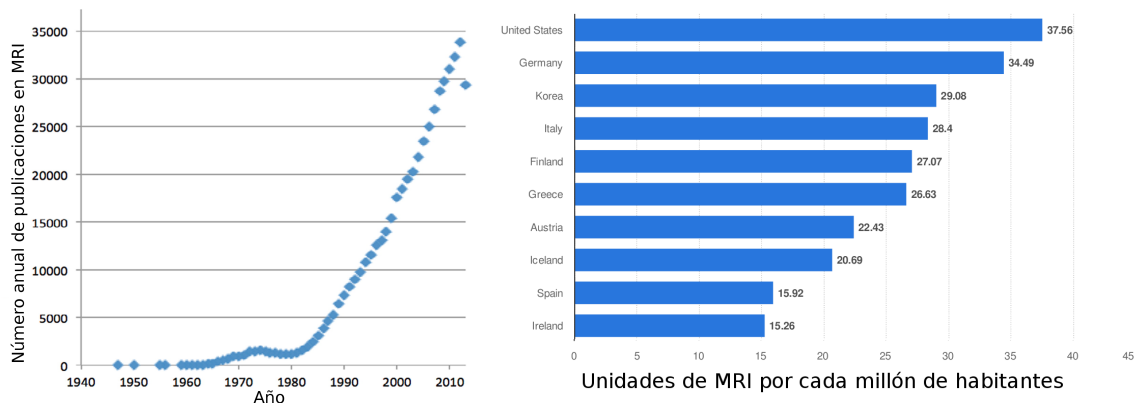
## 2.1 IMAGEN POR RESONANCIA MAGNÉTICA

---

La imagen por resonancia magnética permite la adquisición de imágenes de cualquier zona anatómica del cuerpo humano, pudiéndose obtener una gran cantidad de información sobre sus propiedades físicas y químicas. Ésto, sumado a que se trata de una técnica no invasiva y no ionizante, hace de la resonancia magnética una de las modalidades médicas más utilizadas y con mayor expansión de los últimos años.

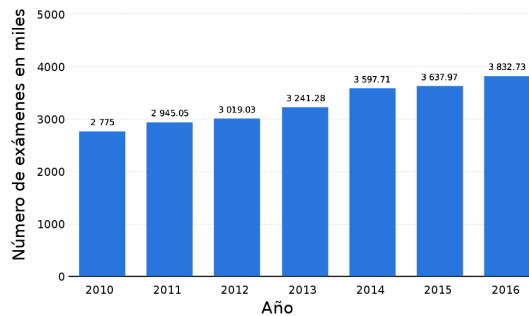
El inicio de esta prometedora técnica fue gracias a Isidor Isaac Rabi, que demostró que una molécula era capaz de emitir ondas de radio a frecuencias específicas ([Rabi et al., 1938](#)). Posteriormente, Felix Bloch y Edward Mills extendieron este trabajo a sólidos y líquidos ([Bloch and Purcell, 1952](#)). No obstante, no fue hasta 1971 que Raymond Damadian descubrió que el tejido cancerígeno tenía un tiempo de relajación distinto al tejido sano en MR ([Damadian, 1971](#)), siendo pionero en el uso de esta técnica a la aplicación clínica. Finalmente, la primera imagen de resonancia fue obtenida en 1973, donde Paul Lauterbur demostró que era posible usar el fenómeno de NMR para la creación de una imagen ([Lauterbur et al., 1973](#)). Cuatro años después, se publicaron las primeras imágenes humanas de MR.

Actualmente, su éxito se ve reflejado en la evolución del número de publicaciones relacionadas con la MRI (Figura 2.1a) en los últimos años a nivel mundial. De la misma forma, este desarrollo es visible también en España. Este país contaba en el año 2017 con 15.92 máquinas de resonancia magnética disponibles por cada millón de personas, situándose entre los primeros del mundo (Figura 2.1b) en este indicador. Este dato es sin duda significativo si lo comparamos con la situación que se producía en 1980, año en el que sólo existían 12 máquinas de resonancia magnética en el mundo. Otro indicador de su desarrollo es el crecimiento del número de exámenes de MR que se hacen en España (Figura 2.1c), donde el crecimiento entre los años 2010 y 2016 ha sido del 38 %.



(A) Publicaciones globales de MR.

(B) Escáneres de MRI por millón.



(C) Exámenes de MR en España.

FIGURA 2.1: Indicadores de la expansión de la resonancia magnética. La Figura (A) ha sido obtenidas del artículo [Edelman \(2014\)](#) y muestran el número de publicaciones totales que contienen o NMR o MRI en su título. La Figura (B) enumera distintos países según el número de escáneres de MRI que tengan disponibles por cada millón de habitantes. Los datos proceden de la [OECD](#) y la gráfica ha sido obtenida de [OECD \(2017\)](#). La Figura (C) ha sido obtenida de [Eurostat \(2016\)](#) y contiene las estadísticas oficiales de Eurostat sobre el número de exámenes anuales de resonancia magnética realizados en España.

A continuación se analizarán las partes que componen un escáner de resonancia magnética y los requerimientos de su instalación (Sección 2.1.1). Descrito el entorno, será más fácil entender los pasos que técnico en imagen debe realizar durante una adquisición rutinaria. Éstos serán enumerados seguidamente (Sección 2.1.2)

### 2.1.1 ESCÁNER DE MRI

Un escáner de MR está compuesto principalmente por tres componentes (Zhi-Pei Liang, 2000) que son enumerados brevemente a continuación:

- **Imán principal.** Es el imán encargado de crear un campo magnético estático. Puede ser de tres tipos: permanente, resistivo y superconductor. En el uso clínico, los imanes superconductores son los más comunes, puesto que proporcionan un campo magnético estático más intenso, más homogéneo y más estable que los otros tipos. La intensidad del campo magnético generado en las máquinas de uso clínico general va desde 1,5T a 7T, siendo los más comunes los imanes de 1,5T y de 3T. Este campo, en comparación con el de la tierra, cuyas unidades son del orden de  $\mu T$ , es enorme, por lo que hay que poner especial atención a la presencia accidental de objetos metálicos cuando se está cerca de uno de estos sistemas.
- **Sistema de gradientes.** Es un sistema formado habitualmente por 3 bobinas colocadas de forma ortogonal entre sí. Estas bobinas, cuando son activadas, están diseñadas para crear gradientes en el campo magnético en la dirección deseada. Además son utilizados para la corrección de inhomogeneidades de campo, por lo que la intensidad mínima del gradiente debe ser mayor que la inhomogeneidad máxima del campo magnético principal. También son diseñados para que puedan pasar del estado apagado a la máxima potencia en el menor tiempo posible, puesto que la rapidez de aplicación de estos gradientes es de gran importancia en esta modalidad de imagen.
- **Sistema *radiofrecuencia* (RF)** Está compuesto por un sistema transmisor capaz de crear campos magnéticos giratorios y un sistema receptor capaz de convertir la señal procedente del cuerpo en señales eléctricas legibles. En ocasiones, ambas tareas son realizadas por la misma bobina, a la que se le denominaría como transceptor. Como añadido al sistema incorporado en la máquina, con el fin de conseguir campos más homogéneos y una mejor lectura, se utilizan bobinas externas, cuyos diseños están optimizados para una zona anatómica concreta. Existen bobinas para la zona de la cabeza, el torso y las articulaciones, entre otras.

La instalación de un escáner superconductor es compleja, ya que para su funcionamiento requieren sistemas de refrigeración, habitualmente con helio líquido, que mantengan sus circuitos cerámicos a baja temperatura y así mantengan su superconductividad. Además, para que las mediciones sean precisas, éstas no pueden ser perturbadas por señales de RF externas. En consecuencia, el escáner debe situarse en el interior de una habitación aislada con una jaula de Faraday.

El plano de una instalación típica de un equipo en un entorno diagnóstico, puede verse en la Figura 2.2, en la que el escáner está situado en la parte central de la sala. La estructura alargada que se desprende de la máquina es la camilla donde el paciente es situado. Las bobinas utilizadas durante la adquisición, además del equipo para el acomodamiento del paciente, son almacenadas en el interior de la sala del escáner. Junto a la sala del escáner se encuentra la sala de control, la cual debe situarse de forma que: en primer lugar, tenga una línea de visión directa a la puerta que da acceso al escáner; en segundo lugar, también debe de poderse ver en todo momento al

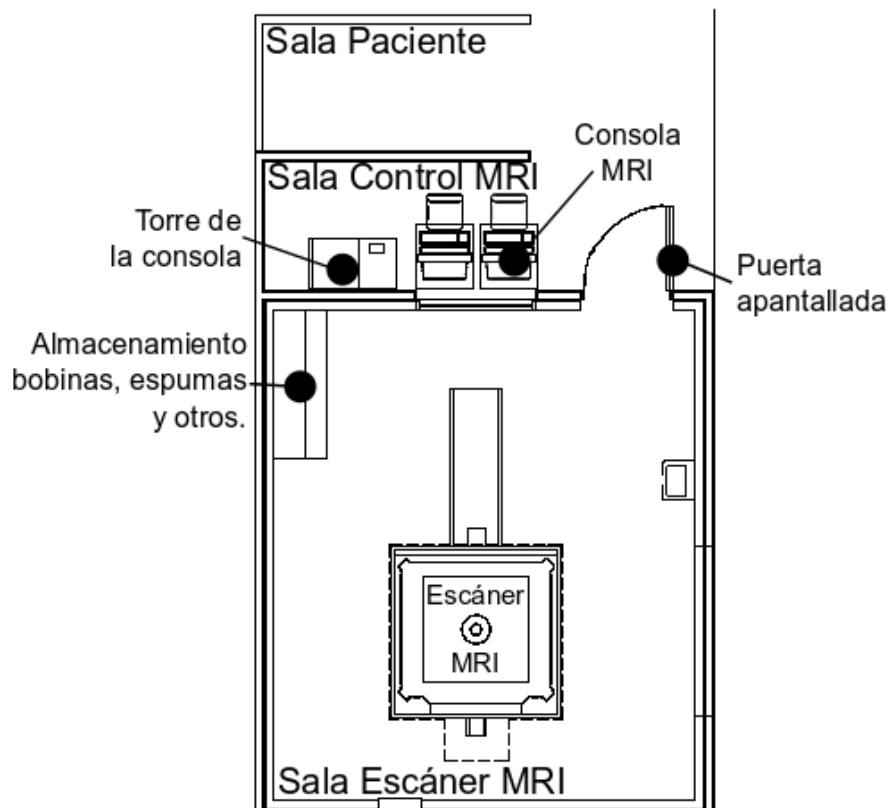


FIGURA 2.2: Plano con la disposición de las 3 salas habituales en una instalación de uso médico de un escáner MRI, a saber: sala del paciente, sala de control de la máquina y sala donde está situado escáner de MRI.

paciente desde la posición de operación del escáner, lo cual es posible mediante el uso de cristal apantallado. Finalmente, suele existir una tercera sala donde se realiza la interacción con el paciente antes de entrar a la sala del escáner.

### 2.1.2 PROCESO DE ADQUISICIÓN

El proceso completo para la realización de una adquisición de MRI está formado por diversos pasos que son ejecutados en las tres salas mostradas en la Figura 2.2. Dichos pasos son descritos a continuación (Elmaoglu and Çelik, 2012):

1. Preparación del paciente. El operador o técnico explica los riesgos que implica la prueba, realiza un cuestionario para asegurar que el paciente lo ha entendido y comprueba que no presente ninguna contraindicación o alguna condición de salud incompatible con el proceso, obteniendo el consentimiento informado del paciente. Además, debe realizar un examen en busca de cuerpos o dispositivos metálicos. Un procedimiento estándar para evitar la introducción de elementos metálicos en la sala del escáner es pedir al paciente

que se desvista y se ponga la ropa suministrada. En el caso de que se lleve sujetador armado, éste también debe retirarse. Esta operación se realiza habitualmente en una sala donde el paciente pueda tener intimidad, y alejada de la sala del escáner. En la Figura 2.2 esto se correspondería con la Sala Paciente.

2. Posicionamiento del paciente. El paciente es dirigido o trasladado a la Sala del Escáner (Figura 2.2) donde se le coloca en la camilla, habitualmente en posición supina. A continuación se colocan las bobinas externas sobre el paciente y se utilizan espumas para inmovilizarlo. Las espumas también pueden ser utilizadas para colocar la parte que va a ser escaneada lo más cerca del isocentro de la bobina, donde el campo es más homogéneo. Una vez colocado cualquier dispositivo externo necesario, incluido un botón que permita al paciente alertar al técnico durante la exploración, el paciente pasa a ser introducido en el imán de la máquina. Un aspecto clave durante su introducción es la localización de un punto de referencia mediante una guía laser emitida por la máquina. Este punto suele estar marcado en las bobinas empleadas durante la adquisición, pero si no suelen emplearse estructuras anatómicas como el puente de la nariz o el ombligo. Una vez situado al paciente correctamente en el interior de la máquina, el operario se mueve a la sala de control, asegurándose de que deja la puerta de la sala del escáner correctamente cerrada. A partir de este punto, el resto de operaciones se llevan a cabo desde la consola de control del escáner.
3. Introducción de los datos. El técnico debe introducir los datos personales del paciente así como los datos que identifiquen al técnico que va a llevar a cabo el procedimiento. Algunos de estos datos pueden tener relevancia para la máquina, como es el caso del peso del paciente, que es empleado por la máquina para el cálculo del *tasa de absorción específica o Specific Absorption Rate (SAR)*, o pueden ser destinados a un uso interno del hospital.
4. Posición del paciente y bobina. A continuación se le indica al escáner la posición en la que ha sido introducido el paciente y la bobina que está siendo empleada, que debe corresponderse con la colocada al paciente. En los equipos más modernos la detección de la bobina suele ser automática.
5. Elección del protocolo. En la mayoría de las instalaciones de resonancia magnética se ha creado una extensa lista de protocolos, los cuales se adaptan tanto a la patología que trate de diagnosticarse como al escenario clínico que presente el paciente. De esta manera, el técnico seleccionaría el protocolo que mejor se adapte a su fin y comenzaría la adquisición. Sin embargo, en caso de no existir un protocolo adecuado, contaría con la posibilidad de seleccionar una secuencia genérica y, posteriormente, adaptar los parámetros por defecto a las necesidades concretas de la situación.
6. Obtención del survey. Habitualmente, en un protocolo, la primera secuencia que lo conforma es llamada localizador o survey. Esta adquisición resulta en un conjunto de imágenes obtenidas a lo largo de 3 planos y que cuentan con una baja resolución y un *campo de visión o Field Of View (FOV)* grande. El survey es empleado para guiar la planificación geométrica de las secuencias posteriores.

7. Colocación del localizador espacial de los cortes. Este paso es conocido también como planificación de los cortes o planificación espacial. Consiste en determinar la posición y la angulación de los cortes de forma gráfica. También deben variarse los parámetros del protocolo para su adaptación a la zona anatómica. Dichos parámetros serían el FOV, el grosor de los cortes, la separación entre ellos, la dirección de codificación de fase o la dirección de codificación de frecuencia.
8. Fijación de parámetros de la secuencia, lo que determinará las características de la imagen. Para ello es necesario conocer con precisión la finalidad de cada parámetro, así como su influencia en la calidad de la imagen, el tiempo de adquisición, la aparición de artefactos...
9. Preescaneo. Antes de comenzar la adquisición de la imagen, se realiza un breve procedimiento de calibración que dura unos segundos. Este proceso está normalmente automatizado, ejecutándose inmediatamente antes de realizar la adquisición propiamente dicha. Sin embargo, en ocasiones podría necesitarse un ajuste manual.
10. Adquisición de la imagen. Una vez el pulso de secuencias es ejecutado, la imagen es adquirida y mostrada en la consola del escáner. Es importante que ésta sea visualizada y su calidad sea evaluada para que, en caso de que sea defectuosa, pueda repetirse. El proceso visto en entre los puntos 7 y 11 debe repetirse para cada una de las secuencias que forme el protocolo.
11. Post-procesado de la imagen. Algunas técnicas requieren que la imagen resultante sea procesada. Esto puede realizarse en la consola del escáner o en un equipo separado.
12. Archivo de datos. Cuando todas las imágenes hayan sido adquiridas, los datos deben ser almacenados en un *sistema de archivo y comunicación de imágenes o Picture Archiving and Communication System (PACS)*, utilizando el estándar de almacenamiento *Digital Imaging and Communication On Medicine (DICOM)*.

## 2.2 FUNDAMENTOS DE MRI

---

La adquisición de imágenes por MR está basada en el fenómeno NMR, el cual ocurre cuando los núcleos de cierto tipo de átomos son, primero, sometidos a un campo magnético estático y, posteriormente, sometidos a un campo magnético oscilante.

El cuerpo humano está formado principalmente por agua, que, como bien es sabido, es una molécula compuesta por dos átomos de hidrógeno y un átomo de oxígeno. Específicamente, el núcleo del átomo de hidrógeno posee un momento cinético fijo o spin, que lo hace girar en torno a su propio eje (Figura 2.3a). Este movimiento, dado que su núcleo posee carga eléctrica no nula, produce un campo magnético asociado.

Si a este campo le añadimos un campo magnético externo, la interacción entre ambos producirá un nuevo movimiento llamado movimiento de precesión (Figura 2.3b). Ambos hechos, que el núcleo atómico en cuestión sea susceptible de producir el fenómeno de la resonancia magnética nuclear, y que exista una cantidad suficiente de estos átomos como para que la señal pueda ser percibida, hace que el fenómeno NMR sea posible en nuestro cuerpo. Además, aunque



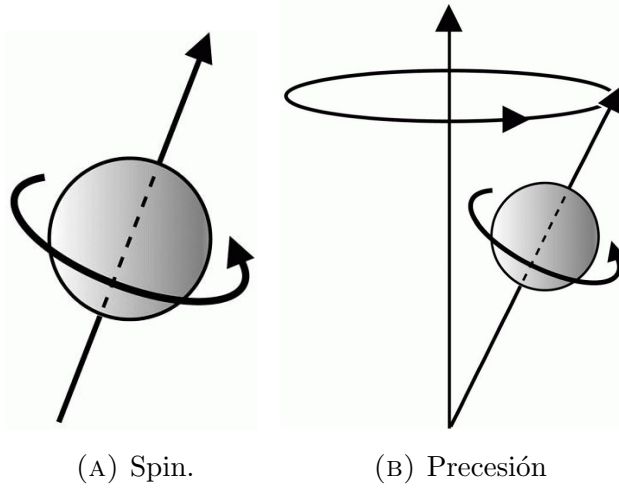


FIGURA 2.3: La figura (A) muestra el movimiento del núcleo de un átomo con spin no nulo. La figura (B) muestra el movimiento de precesión que describe el núcleo del átomo cuando un campo magnético externo es introducido, donde dicho campo viene representado por el vector vertical. Ambas figuras han sido obtenidas de [Hashemi et al. \(2012\)](#)

dicho fenómeno tenga en sentido estricto un origen cuántico, puede ser, sin embargo, explicado utilizando solamente la mecánica clásica ([Hanson, 2008](#)).

En MRI, el campo estático externo suele denominarse *campo magnético principal* ( $\vec{B}_0$ ), que es producido por el imán principal de la máquina y que por convenio se sitúa en la dirección  $z$ .

$$\vec{B}_0 = B_0 \vec{z} \quad (2.1)$$

La velocidad a la que precesiona el núcleo cuando es sometido a un campo estático es denominada *frecuencia de resonancia o frecuencia de Larmor* ( $\omega$ ). Esta frecuencia depende de la intensidad del campo magnético y también de un factor llamado *relación giromagnética* ( $\gamma$ ), que para el caso del hidrógeno es  $\gamma = 2,675 \times 10^8 \text{ rad/s/T}$ . Por tanto,  $\omega$  para un  $\vec{B}_0$  dado ( $\omega_0$ ) será:

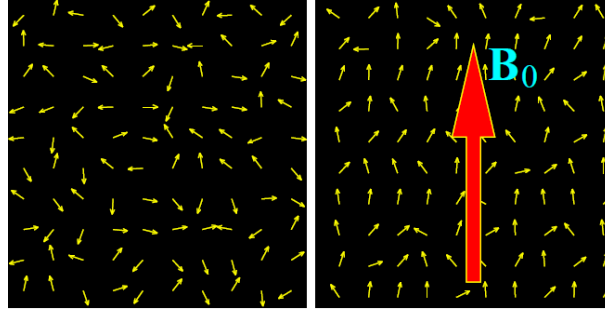
$$\omega_0 = \gamma \|\vec{B}_0\|_2 \quad (2.2)$$

Sin embargo, para poder generar una señal medible aún es necesario aplicar un campo magnético oscilante. En MRI, éste es producido por los llamados pulsos de radiofrecuencia (RF). Este nuevo campo magnético se denomina habitualmente  $B_1$ . La aplicación de dichos pulsos es perpendicular al campo  $\vec{B}_0$  y su frecuencia de oscilación, en el caso ideal, vendría dada por  $\omega_0$ . La expresión viene descrita en ([Zhi-Pei Liang, 2000](#)):

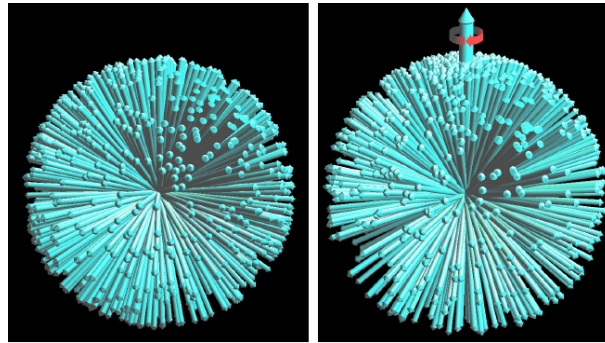
$$\vec{B}_1(t) = B_1^e(t) \cdot [\cos(\omega_{RF} \cdot t) \vec{x} - \sin(\omega_{RF} \cdot t) \vec{y}] \quad (2.3)$$

donde  $B_1^e(t)$  define la envolvente del pulso.

Considerados individualmente, cuando no existe un campo magnético estático, el spin de cada uno de los átomos de hidrógeno estaría apuntado a una dirección cualquiera (todas las posibles direcciones serían equiprobables) y, por tanto, la suma vectorial de todos ellos sería nula (parte izquierda de la Figura 2.4). Sin embargo, cuando un campo  $\vec{B}_0$  es introducido, los



(A) Red de magnetización.



(B) Vector de magnetización.

FIGURA 2.4: En la figura (A), cada una de las flechas representa un spin, mientras que en la figura (b) estos spines vienen representados por cada uno de las flechas finas azules. La parte izquierda de la figura muestran la red de magnetización sin campo externo. Cuando el campo externo es introducido, en la parte derecha de la figura (A) vemos cómo los spines tienden a alinearse con dicho campo. En la parte derecha de la figura (B) vemos como aparece una nueva flecha con un grosor mayor, que representa el vector de magnetización producido por  $\vec{B}_0$  y cuya frecuencia de giro se corresponde con  $\omega_0$ . Ambas figuras han sido obtenidas de [Hanson \(2008\)](#)

spines tienden a alinearse con dicho campo, formando una red de magnetización (parte derecha de la Figura 2.4), y dando lugar al *vector de magnetización* ( $\vec{M}$ ).

$$\vec{M} = [M_x, M_y, M_z] \quad (2.4)$$

En este momento, cuando el campo oscilante  $\vec{B}_1$  es introducido, los spines entrarían en resonancia con  $\vec{B}_1$  y  $\vec{M}$  comenzaría a desplazarse de la forma descrita en la Figura 2.5. Este fenómeno es descrito por la ecuación de Bloch:

$$\frac{d\vec{M}}{dt} = \gamma \vec{M} \times \vec{B} - \frac{M_x \vec{x} + M_y \vec{y}}{T_2} - \frac{(M_z - M_z^0) \vec{z}}{T_1}, \quad (2.5)$$

donde  $\vec{B}$  sería el campo magnético total, en este caso  $\vec{B} = \vec{B}_0 + \vec{B}_1(t)$ ,  $M_z^0$  sería el valor de  $\vec{M}$  cuando sólo está presente  $\vec{B}_0$ , y  $T_1$  y  $T_2$  son constantes que caracterizan el proceso de relajación de la señal y que veremos más adelante.

Acabamos de observar el comportamiento de  $\vec{M}$  en torno al marco de referencia del laboratorio  $[\vec{x}, \vec{y}, \vec{z}]$  (ver Figura 2.6a). Ahora bien, es conveniente, por simplicidad, definir un marco

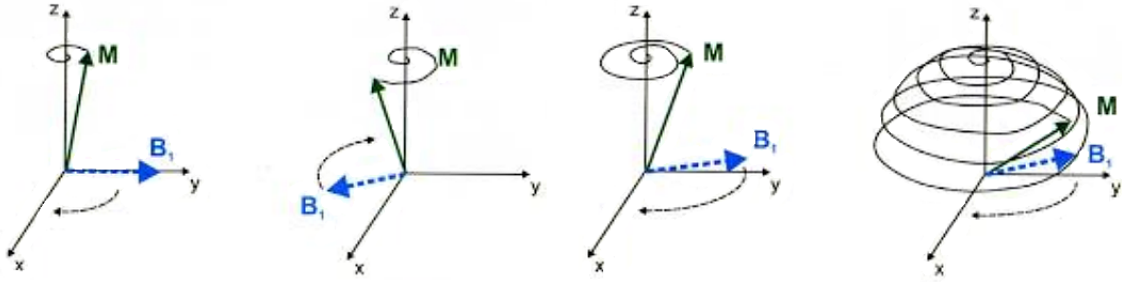


FIGURA 2.5: Esta figura muestra cómo  $\vec{M}$  comienza a girar en torno al eje  $z$ , que es la dirección de aplicación de  $\vec{B}_0$ . Dicho efecto es debida a la aplicación trasversal de *pulso de RF* ( $\vec{B}_1$ ). De esta forma, vemos como  $\vec{M}$  entra en resonancia con  $\vec{B}_1$  cuya frecuencia es  $\omega_0$ . *Figura obtenida de Moratal Pérez et al. (2006)*

de referencia rotatorio, el cual gire alrededor del eje  $z$  a la frecuencia de Larmor. De esta forma, obtenemos un nuevo sistema de coordenadas  $[\vec{x}', \vec{y}', \vec{z}']$  (ver Figura 2.6b y Figura 2.6c).

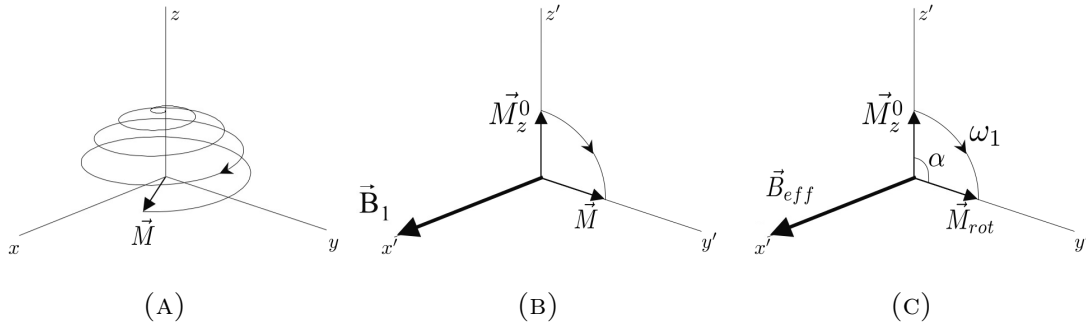


FIGURA 2.6: Estas figuras muestran el desplazamiento de  $\vec{M}$  debido a  $\vec{B}_1$ . La figura (A) utiliza el marco de referencia del laboratorio para ilustrarlo, mientras que las figuras (B) y (C) utiliza el marco de referencia rotatorio. *Las figuras han sido obtenidas de Brown et al. (2014).*

En este marco rotatorio, la ecuación general de Bloch (Ecuación 2.5) puede ser expresada en términos de coordenadas relativas (Zhi-Pei Liang, 2000):

$$\frac{d\vec{M}_{rot}}{dt} = \gamma \vec{M}_{rot} \times \vec{B}_{eff} - \frac{M_{x'} \vec{x}' + M_{y'} \vec{y}'}{T_2} - \frac{(M_{z'} + M_z^0) \vec{z}'}{T_1}, \quad (2.6)$$

donde  $\vec{M}_{rot}$  es el vector de magnetización en el marco de referencia rotatorio y  $\vec{B}_{eff}$  es el campo efectivo aplicado, para este caso  $\vec{B}_{eff} = (B_0 - \omega_{RF}/\gamma) \vec{z}' + B_1 \vec{x}'$ . Observando la Figura 2.6c, la frecuencia a la que precesiona  $\vec{M}_{rot}$  en torno a  $\vec{B}_{eff}$  vendría dada, para este caso, por:

$$\omega_1 = \gamma \|\vec{B}_{eff}\|_2 = \gamma \|\vec{B}_1\|_2 \quad (2.7)$$

y el *ángulo de giro* o *Flip Angle* (FA) ( $\alpha$ ) dependería del *tiempo de aplicación de pulso RF* ( $\tau_p$ ):

$$\alpha = \omega_1 \tau_p \quad (2.8)$$

En el caso que se cumpla la Ecuación 2.7 se daría el fenómeno llamado *on-resonance* y el FA sería el deseado. Sin embargo, cuando  $\vec{B}_{eff} \neq \vec{B}_1$  debido a las inhomogeneidades de campo, el FA pasa a ser distinto del esperado y se produce el fenómeno de *off-resonance*.

Una vez que  $\vec{M}_{rot}$  ha sido perturbado, el sistema de magnetización, en ausencia de  $\vec{B}_{eff}$ , retornaría a su estado de reposo (Ecuación 2.6). Este proceso es conocido como relajación, y está dominado por dos propiedades del tejido: el *tiempo de relajación longitudinal* ( $T_1$ ) y el *tiempo de decaimiento transversal* ( $T_2$ ) (Figura 2.7). En la componente longitudinal,  $\vec{M}_{rot}$  vuelve a realinearse con el campo  $\vec{B}_0$ , mientras que la componente transversal se pierde al cabo de poco tiempo debido a las interacciones entre los spins.

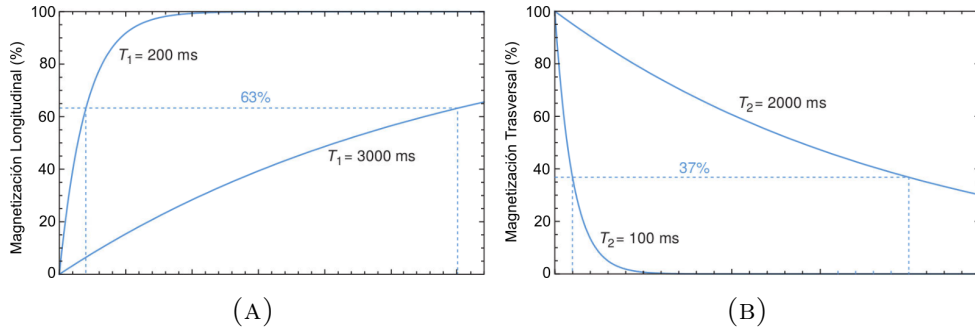


FIGURA 2.7: La figura (A) muestra la recuperación de la magnitud longitudinal modelada por  $T_1$  para dos tejidos distintos. La figura (B) muestra la pérdida de la magnitud transversal modelada por  $T_2$  para dos tejidos distintos. En el caso (A), el valor de  $T_1$  se corresponde con la recuperación del 63% de dicha componente, mientras que en el caso (B), el valor  $T_2$  se da cuando se ha perdido el 37% de la componente transversal. *Las figuras han sido obtenidas de Suetens (2017).*

## 2.2.1 LOCALIZACIÓN DE LA SEÑAL

Hasta ahora, hemos obtenido una señal medible que proviene de todos los espines excitados. Sin embargo, en estas condiciones el receptor no es capaz de distinguir su origen espacial. En consecuencia, es necesario localizar la posición espacial de las señales. Para ello, en primer lugar se realiza una excitación selectiva, que consiste en la creación de un gradiente en la dirección  $\vec{B}_0$ , pero cuya amplitud  $G_s$  varía linealmente a lo largo de la dirección de selección  $\vec{u}_G$ , de forma que:

$$\vec{G}_s = G_x \vec{u}_x + G_y \vec{u}_y + G_z \vec{u}_z \quad (2.9)$$

La introducción de un campo magnético estático nuevo hace que la frecuencia de precesión cambie (Ecuación 2.10):

$$\omega_{RF} = \gamma \|\vec{B}_0 + \vec{G}_s\|_2 \quad (2.10)$$

Un ejemplo numérico de ello puede verse en la Figura 2.8, donde  $\vec{G}_s = G_z \vec{u}_z$ . Además, como puede apreciarse, el ancho de banda del pulso RF ( $\Delta_z$ ) determina el grosor del corte en función de la pendiente del gradiente aplicado, que suele ser del orden de  $mT/m$ .

De esta manera es posible aislar selectivamente la señal procedente únicamente de un corte, donde todos los vectores de las redes de magnetización  $M_{x'y'}$  estarían apuntando en la misma dirección. La Figura 2.9 representa nueve  $M_{x'y'}$ , uno por cada voxel.

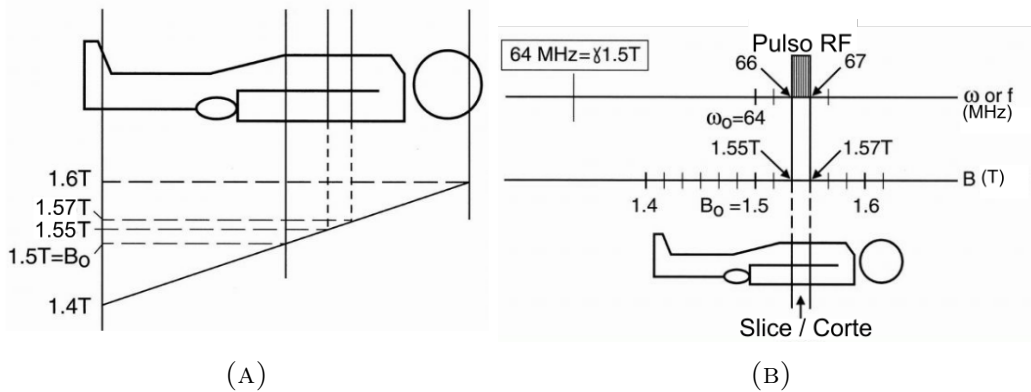


FIGURA 2.8: La figura (A) representa la variación en amplitud del gradiente a lo largo del campo principal. La figura (B) representa lo mismo, de forma que la frecuencia de Larmor que produciría la resonancia sea diferente a lo largo de los posibles cortes axiales. *Las figuras han sido obtenidas de Hashemi et al. (2012).*

Una vez conseguida la selección de corte, en segundo lugar se procede a la codificación en fase y frecuencia de ese corte.

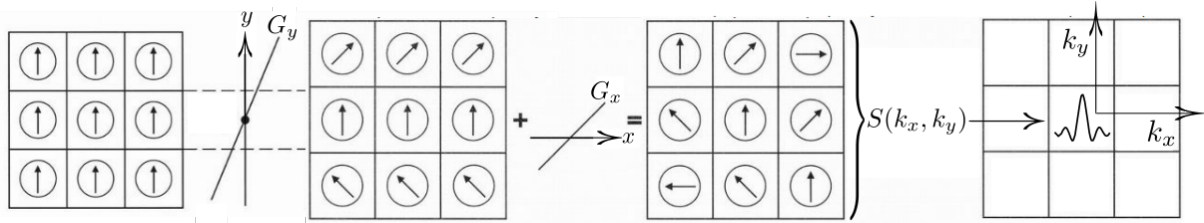


FIGURA 2.9: Codificación en fase y frecuencia de una imagen 2D. La figura ilustra, de izquierda a derecha, el proceso de desfase de las diferentes redes de magnetización producidas por los gradientes en fase y frecuencia, así como la adquisición de la señal de un determinado punto del espacio k. *Figura obtenida obtenida de Hashemi et al. (2012).*

En la Figura 2.9, primero se aplica el gradiente de fase que, por conveniencia, varía su amplitud en la dirección  $y$ . Esto produce que cada  $M_{x'y'}$  en dicha dirección experimente una velocidad  $\omega$  diferente, por lo que, después de un determinado tiempo, los  $M_{x'y'}$  de cada fila se encuentren en una posición diferente.

Seguidamente, se aplica la codificación de frecuencia, donde el gradiente de frecuencia varía su amplitud en la dirección  $x$ . Su aplicación durante un determinado tiempo hace que, nuevamente, los  $M_{x'y'}$  varíen su posición para cada columna.

De esta forma, la aplicación de los gradientes consigue que cada posición dentro del corte seleccionado tenga una  $M_{x'y'}$  distinta. Esto suele describirse como el recorrido de una trayectoria en el denominado espacio k, donde el valor de la señal viene dado por la integral de la imagen de la imagen en un determinado instante. Para un tiempo fijo, el valor de la señal sería el siguiente:

$$S(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_{x'y'}(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy \quad (2.11)$$

Donde  $k_x, k_y$  determinan las coordenadas en el espacio k y dependen del gradiente aplicado y de su duración:

$$\vec{k}_{x,y}(t) = \frac{\gamma}{2\pi} \int_0^t \vec{G}_{x,y}(t') dt' \quad (2.12)$$

## 2.2.2 SECUENCIA DE PULSOS

Las secuencias de pulsos están formadas por una combinación de los pulsos de RF y los gradientes vistos anteriormente. Los pulsos de RF son utilizados para la excitación de los protones, mientras que gradientes son utilizados para la codificación de la señal, la localización del corte y la saturación de la señal, aunque también tienen otras aplicaciones.

Existen una gran cantidad de secuencias de pulsos, cuyos propósitos generales suelen ser la obtención de imágenes con un cierto contraste y la reducción en lo posible del tiempo de adquisición. Sin embargo, en esta sección vamos a centrarnos en dos secuencias básicas en las que se basa la mayor parte del resto de secuencias: *Gradient Echo* (GRE) y *Spin Echo* (SE). Más adelante se explicarán los factores que intervienen en los tiempos de adquisición de una secuencia, y se mencionarán algunas de las formas que hay de reducirlos.

### GRE

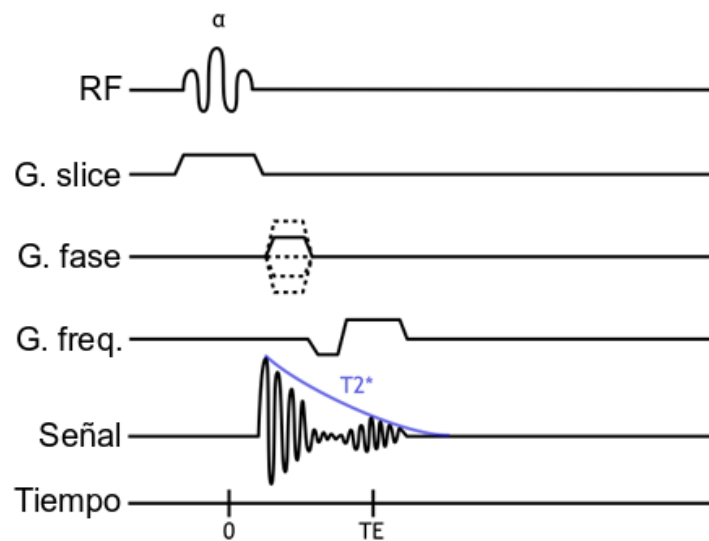


FIGURA 2.10: Secuencia de pulsos GRE. Está compuesta por un pulso de 90 grados, que se aplica junto al gradiente que selecciona el corte. A continuación se aplican los gradientes de codificación de fase y frecuencia para su adquisición. *Figura obtenida de Hashemi et al. (2012).*

GRE es la secuencia de pulsos más sencilla que existe, y su aplicación se muestra en la Figura 2.10. En primer lugar, se aplica pulso de RF, que ocasionaría un giro de  $\vec{M}_{rot}$ . El número de grados lo determina el parámetro FA y se corresponde con la Ecuación 2.8. Además, durante la aplicación de este pulso, el gradiente de selección de corte es activado, de forma que sólo sea excitado el slice deseado.

A continuación, se introduce el gradiente en la dirección de codificación de fase, que nos sitúa en la línea adecuada del espacio k. Finalmente, es aplicado el gradiente de codificación de frecuencia. El valor inicial de este gradiente es negativo, y sirve para situarse al comienzo de la línea del espacio k. Una vez ubicados en la posición correcta, el espacio k comienza a muestrearse.

Recordemos que la señal muestreada se corresponde con el vector  $M_{x'y'}$ , cuya amplitud decrece en función del parámetro del tejido  $T_2$ . Sin embargo, debido a las inhomogeneidades de campo ( $\Delta B$ ), los spines que conforman la red de magnetización comienzan a desfasarse más rápido, a razón de  $T_2^*$ :

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{\gamma \Delta B} \quad (2.13)$$

Finalmente, el tiempo en el que se adquiere la parte central de cada línea del espacio k es llamado *tiempo de eco* (TE). Cada vez que se desea adquirir una línea de este espacio k, se aplica la secuencia completa. El tiempo que transcurre entre los pulsos de RF iniciales de la cada secuencia es llamado *tiempo de repetición* (TR).

Por tanto, una estimación aproximada del tiempo que tardaría en adquirirse una imagen completa, consistiría en multiplicar el TR por el número de líneas del espacio k que se muestrean para construir la imagen.

## SE

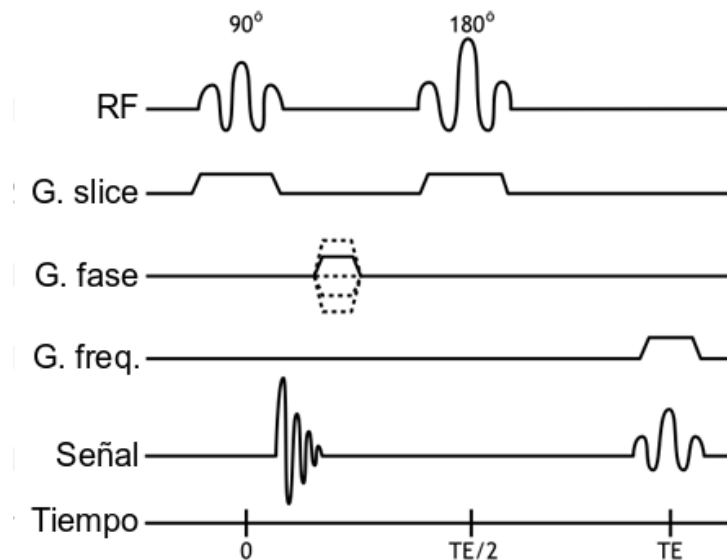


FIGURA 2.11: Secuencia de pulsos SE. En esta secuencia se aplican dos pulsos sucesivos junto al gradiente encargado de seleccionar el corte. Primero se aplica un pulso de  $90^\circ$  y posteriormente un pulso de  $180^\circ$ . El tiempo entre ambos pulsos se denomina  $TE/2$ . Para la adquisición de la señal son necesarios también el uso de los gradientes de fase y de frecuencia. *Figura obtenida de Hashemi et al. (2012).*

El propósito principal de esta secuencia es conseguir que la atenuación de la señal dependa del parámetro  $T_2$  en lugar de depender del parámetro  $T_2^*$  (atenuación más rápida y más sensible



a las inhomogeneidades de campo) como ocurre en las secuencias GRE. El método para hacerlo fue propuesto por [Hahn \(1950\)](#) y consiste, como se ve en la [Figura 2.11](#), en: primero, aplicar la excitación  $\vec{M}_{rot}$  con un ángulo de  $90^\circ$ ; a continuación, se aplican los gradientes de codificación de fase y de frecuencia para un correcto posicionamiento en el espacio  $k$ ; posteriormente, se aplica un pulso de  $180^\circ$ , llamado pulso de inversión, y finalmente se realiza la lectura del espacio  $k$ .

Durante el tiempo trascurrido entre el pulso de  $90^\circ$  y el pulso de  $180^\circ$ , los spines del voxel se han estado desfasando entre sí debido a las inhomogeneidades del campo, al igual que como ocurría en la secuencia GRE. Sin embargo, la aplicación del pulso de  $180^\circ$  hace que el eje de giro de los spines se invierta. En consecuencia, mientras antes se estaban desfasando, ahora, el sentido del giro hace que comiencen a ponerse en fase. Si se espera un tiempo igual al trascurrido entre el pulso de  $90^\circ$  y de  $180^\circ$  para la lectura del espacio  $k$ , se consigue una resincronización completa de los spines, de forma que la constante de tiempo con la que decrece la señal vendría dada únicamente por  $T_2$ .

De esta forma, TE es el momento en el que se adquiere la parte central de la línea del espacio  $k$ , y el momento del pulso de inversión, en consecuencia, será la mitad de TE. El tiempo entre la adquisición de líneas estaría marcado por TR.

## ADQUISICIÓN ACELERADA

Como se ha visto, el tiempo de una adquisición es proporcional al TR, al número de líneas de la imagen y al número de imágenes o cortes que desee adquirirse. El resultado es que los tiempos de adquisición que el paciente debería pasar de forma estática en el interior de la máquina serían extremadamente elevados.

Para reducir estos tiempos existen varias técnicas. Mencionamos algunas de ellas brevemente a continuación:

- Adquisición en paralelo. Se basa en la adquisición de una cantidad reducida de los datos del espacio  $k$ , aprovechando un receptor que esté formado por varias bobinas. El submuestreo del espacio  $k$  produce aliasing. Sin embargo, aprovechando la redundancia de los datos debida a los múltiples receptores, es posible aplicar técnicas de reconstrucción de la imagen. La técnica concreta depende del fabricante, y las más conocidas son SENSE ([Pruessmann et al., 1999](#)) y GRAPPA ([Griswold et al., 2002](#)). Los factores de aceleración más comunes son de 2, 3 y 4, lo que significa que el tiempo de adquisición es (aproximadamente) 2, 3 o 4 veces menor.
- Secuencias rápidas, como es el caso de *Fast Spin Echo* (FSE) ([Listerud et al., 1992](#)). La técnica consiste en aplicar varias secuencias de pulsos en un único intervalo TR. El número de secuencias aplicadas por cada TR se corresponde con el número de líneas adquiridas en ese tiempo, y es denotado por el parámetro *longitud del tren de eco* o *Echo Train Length* (ETL), que determina el factor de aceleración.
- Secuencia *Echo-Planar imaging* (EPI) ([Turner et al., 1990](#)). Esta secuencia permite, en su versión “single-shot”, una adquisición completa del espacio  $k$  en un único TR. Se basa en la aplicación de gradientes fase y frecuencia durante la lectura de la señal, de forma que el espacio  $k$  es completamente muestreado en un único eco.



- Adquisición Half-Fourier. Esta técnica hace uso de la simetría existente en el espacio  $k$ , y consiste en adquirir el 51 % de dicho espacio y aplicar la reconstrucción del mismo aprovechando la simetría mencionada. Con esta técnica, el tiempo de adquisición es reducido casi a la mitad.
- Porcentaje de escaneo. Consiste en adquirir solamente las líneas centrales del espacio  $k$ , dejando sin adquirir las líneas de alta frecuencia. De esta forma, se pierde resolución en la dirección de codificación de fase. El porcentaje que va a ser adquirido es el que determina el factor de aceleración.

### 2.2.3 ARTEFACTOS

Llamamos artefactos a alteraciones en la imagen obtenida o a estructuras que aparecen en la imagen pero que sin embargo no tienen una correspondencia con una estructura real. Son indeseables porque en general dificultan o incluso impiden la utilización de la imagen médica para su propósito diagnóstico por parte de un experto. La MRI, al igual que otras técnicas de imagen médica, puede verse afectada por una gran variedad de artefactos. A lo largo de esta sección hablaremos de algunos de los más comunes y cómo podrían ser solucionados o minimizados.

#### RUIDO

En la práctica, cuando se desea realizar una medida, ésta siempre está contaminada por una componente indeseable a la que llamaremos ruido o señal aleatoria. La relación entre la señal que queremos medir y la señal aleatoria que la contamina es llamada *relación de señal a ruido o signal-to-noise ratio* (SNR).

La SNR en una adquisición MR viene dada por una relación escalar entre la dimensión del voxel de donde proviene la señal, y la cantidad de tiempo empleado para la adquisición de la señal (Edelstein et al., 1986):

$$SNR \propto \Delta x \Delta y \Delta z \sqrt{T_{adq, total}} \quad (2.14)$$

donde  $\Delta x \Delta y \Delta z$  es el volumen del voxel y  $T_{adq, total}$  es el tiempo total empleado en la adquisición. Intuitivamente, cuanto mayor sea el tamaño del voxel, existirán más átomos de hidrógeno excitados en su interior, y por lo tanto la cantidad de señal que estos átomos emitan será mayor. Por otro lado, cuanto mayor sea el tiempo que la ventana de adquisición de datos esté muestreando la información, al tratarse el ruido de una señal aleatoria, su integral tenderá a cero y menor será su efecto. Sin embargo, maximizar la SNR de esta manera empeoraría la resolución de la imagen y produciría tiempos de adquisición muy largos. Esto tiene además otros efectos contraproducentes en la imagen, por lo que es necesario llegar a un equilibrio.

La aplicación de la Ecuación 2.14 en una adquisición 2D sería la siguiente:

$$SNR_{2D} \propto \Delta x \Delta y \Delta z \sqrt{N_{fase} NEX T_{adq}} \quad (2.15)$$

donde  $N_{fase}$  es el número de muestras en la dirección de fase,  $NEX$  es el número de excitaciones o el número de veces que se obtendría cada muestra, y  $T_{adq}$  es el tiempo empleado en cada muestra, que es inversamente proporcional al *ancho de banda o BandWidth* (BW) del receptor.

De la misma forma, en una adquisición 3D existirían dos direcciones de codificación de fase, por lo que:

$$SNR_{3D} \propto \Delta x \Delta y \Delta z \sqrt{N_{fase1} N_{fase2} NEX T_{adq}} \quad (2.16)$$

donde  $N_{fase1}$  y  $N_{fase2}$  son el número de muestras en cada una de las respectivas direcciones de fase.

La relación entre las ecuaciones 2.15 y 2.16 muestra cómo la adquisición en 3D es del orden de  $\sqrt{N_{fase2}}$  veces más eficiente respecto al ruido que la adquisición 2D.

Como puede apreciarse, ninguna de las ecuaciones vistas (Ecuaciones 2.14, 2.15, 2.16), contiene información alguna sobre las propiedades físicas de los receptores, del campo o las características magnéticas del tejido, por lo que no es posible obtener un número absoluto.

## ALIASING, WRAPAROUND O FOLDOVER

Como se ha visto en esta sección, durante la fase de selección del slice, éste es excitado por completo, por lo que, si el FOV que vamos a querer adquirir es menor que la zona excitada, se puede producir este artefacto.

Dado que los gradientes utilizados en la codificación se extienden a lo largo de todo el corte, el FOV determinaría la frecuencia máxima y mínima de cada gradiente y cualquier frecuencia que sobrepase estos límites no será muestreada correctamente. La Figura 2.12 muestra cómo la señal cuya codificación en frecuencia es mayor que la frecuencia máxima establecida por el FOV es registrada erróneamente. De esta forma, la frecuencia percibida va a ser la frecuencia actual menos 2 veces las frecuencia de Nyquist.

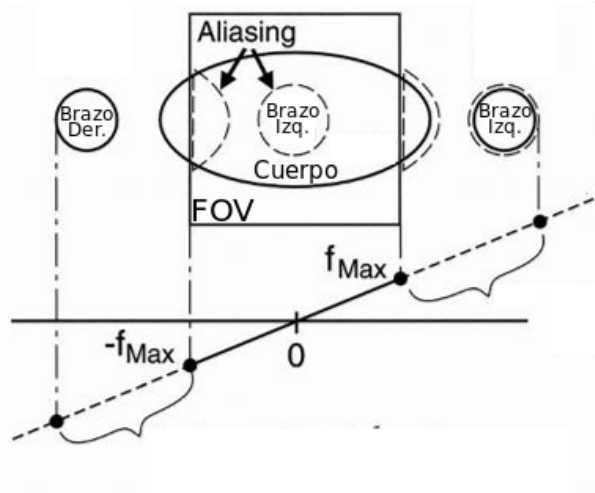


FIGURA 2.12: Aliasing en la codificación en frecuencia. Esta figura muestra cómo se muestrearía erróneamente la imagen en el caso de ocurra este artefacto. Por simplicidad, sólo muestra el aliasing producido por el brazo izquierdo y parte del cuerpo, sin embargo, para este caso, también ocurriría con el brazo derecho y la otra parte del cuerpo que supera el FOV en esta dirección. *Figura obtenida de Hashemi et al. (2012).*

En la dirección de codificación de frecuencia, las máquinas cuentan con un filtro limitador de banda que corrige este efecto. Sin embargo, en la dirección de fase, este filtro no puede ser

aplicado. Entonces, para evitar este artefacto puede en primer lugar aumentarse el FOV en esta dirección, o bien intercambiar la dirección de fase y de frecuencia. En segundo lugar, puede utilizarse la opción de imagen *No Phase Wrap* (NPW), por la cual, la máquina aumenta el FOV que emplea realmente aunque luego recorta la imagen restante, de forma que este proceso resulta transparente para el operador. La tercera opción sería el uso de bandas de saturación para anular la señal en las zonas deseadas.

## DESPLAZAMIENTO QUÍMICO

El desplazamiento químico es un artefacto que se produce debido a que la velocidad de precesión de los protones de hidrógeno de la grasa de los tejidos es ligeramente diferente a la experimentada por los protones del hidrógeno de las moléculas de agua. Dicha diferencia está relacionada con el diferente entorno químico que estos protones encuentran en cada caso.

La relación entre la frecuencia de precesión del agua y de la grasa es de 3.5 *partes por millón* (ppm), por lo que:

$$\omega_{agua} = \gamma B_0 \quad (2.17)$$

$$\omega_{grasa} = (1 - 3,5 \times 10^{-6}) \omega_{agua} \quad (2.18)$$

$$\Delta_{\omega} = 3,5 \times 10^{-6} \gamma B_0 \quad (2.19)$$

Esto significa que, a la hora de muestrear la señal en el espacio k, la grasa va tener un error de  $\Delta_{\omega}$  (Hz). Por otro lado, la frecuencia a la que se muestrea un a línea del espacio k se denomina ancho de banda (BW). Dependiendo del número de muestras tomadas, la frecuencia de separación entre cada píxel es de  $\frac{BW}{n^{\circ} \text{muestras}}$ . Cuando esta separación no es suficiente, la grasa aparecerá desplazada un número determinado de píxeles que viene dado por:

$$\text{Desplazamiento} = \frac{\Delta_{\omega} n^{\circ} \text{muestras}}{BW} \quad (px) \quad (2.20)$$

Por tanto, cuanto mayor sea el BW, menor será este efecto.

## MOVIMIENTO

Este artefacto es debido a movimientos del paciente, ya sean aleatorios, como un estornudo, o periódicos, como el debido a la respiración, el latido cardiaco o incluso el movimiento de la sangre en los vasos. Los movimientos pueden ser voluntarios o involuntarios, pero si suceden durante la adquisición, dado que la máquina no puede reconocer dicho movimiento, considera que todas las muestras del espacio k son coherentes. Sin embargo, cada una de ellas podría pertenecer a imágenes ligeramente diferentes debido al movimiento. Esta inconsistencia en el espacio k es la productora de los artefactos y se traduce en una imagen con un efecto fantasma o “ghosting”.

Cuando el movimiento es aleatorio, la inconsistencia pertenece al corte completo, y el efecto puede apreciarse en toda la imagen. Sin embargo, cuando el movimiento es periódico, bien debido al flujo de sangre, al corazón o al *líquido cefalorraquídeo* o *CerebroSpinal Fluid* (CSF), el artefacto se visualiza en la zona donde se ha producido el movimiento pudiendo ser más difícil de detectar.

Finalmente, cuanto mayor sea el tiempo de adquisición, es decir, cuanto más tiempo pase entre cada muestra, mayor será la inconsistencia, ya que existe un intervalo mayor en el que puede producirse ese movimiento. Por ello, este artefacto está mucho más presente en la dirección de fase que en la dirección de frecuencia.

Una solución para evitar el artefacto producido por el movimiento periódico de la sangre es colocar bandas de saturación a ambos lados de las arterias y venas principales.

## SPIKE

El artefacto de tipo *spike* es causado por un mal funcionamiento en la electrónica de los sistemas o bien en las conexiones, de forma que un único punto del espacio  $k$  recibe una señal anormalmente alta (un pico). Esto produce un efecto de rayado en la imagen. En función de la posición en el espacio  $k$  en la que se produzca dicho pico, las líneas del rayado estarán más o menos separadas, y su ángulo variará. Dado que la codificación en frecuencia es a lo largo del eje horizontal, cuanto más cerca esté de la parte central mayor será la separación entre las líneas. Por otro lado, el desplazamiento a lo largo de la dirección de fase (vertical) producirá que el ángulo de las líneas varíe. Debe tenerse en cuenta que cuanto más cerca esté el pico del centro del espacio  $k$ , dicho artefacto será más visible.

Cuando este artefacto es detectado, deben revisarse que las conexiones de las bobinas sean correctas.

## CONTAMINACIÓN RF

La contaminación RF se debe a que, por alguna razón, una radiofrecuencia se está filtrando en el interior de la sala de la máquina. Dado que los receptores son extremadamente sensibles, una línea del espacio  $k$  será escrita con dicha frecuencia como ruido. Su efecto sobre la imagen es la aparición de unas líneas discontinuas.

Por lo general, aunque pudiesen existir problemas con la jaula de Faraday de la habitación, este artefacto suele originarse porque la puerta está mal cerrada. No existe una forma de solucionar este problema desde la consola, por lo que debe revisarse el aislamiento de la sala.

## Capítulo 3

# SIMULADORES DE RESONANCIA MAGNÉTICA

---

**P**UESTO que el objetivo central de esta tesis es el diseño, desarrollo y evaluación de un simulador de resonancia magnética con propósito educativo, este capítulo hace un repaso por los distintos simuladores de MRI que se han propuesto en la literatura. Existen herramientas de simulación diseñadas con diferentes finalidades, de modo que se ha realizado una división entre ellas teniendo en cuenta este factor. En primer lugar, se enumeran y describen los simuladores encontrados con fines educativos. Posteriormente se nombran y explican aquellos cuyo propósito sirva a un fin investigador. Para finalizar se realiza una breve discusión sobre todos los simuladores analizados.

## 3.1 INTRODUCCIÓN

---

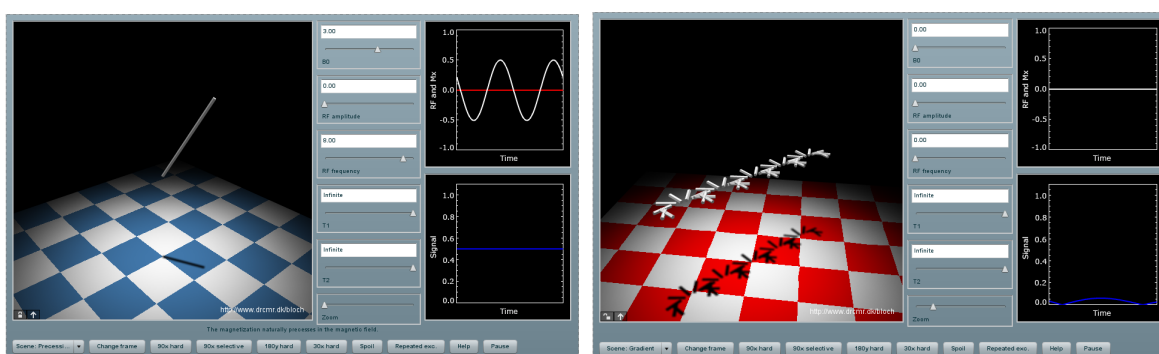
La imagen por MR es una herramienta muy compleja y de gran versatilidad como vimos en el Capítulo 2. Con el fin de mejorar en diferentes facetas de este área, se han creado una gran diversidad simuladores. Un caso interesante, al tratarse de un simulador físico, es el de *MRI Simulator* (Hallowell et al., 2008). Éste artefacto imita la estructura física de una máquina de MR, tratando de emular además los diferentes sonidos y sensaciones que transmite. Su finalidad es proporcionar a los pacientes que van a someterse a una exploración de resonancia un entrenamiento previo a la adquisición real, para así tratar de reducir las exploraciones fallidas por razones de claustrofobia, ansiedad y movimientos excesivos de la cabeza y el cuerpo. Pero más allá de lo excepcional del ejemplo, la mayoría de estos simuladores de MR sólo necesitan un ordenador y el software adecuado para su ejecución.

En este capítulo se va a distinguir entre dos tipos de simuladores según su finalidad: los destinados al mundo educativo, que se encuentran en la Sección 3.2, y los que cuya finalidad es la investigación, que son descritos en la Sección 3.3.

## 3.2 SIMULADORES EDUCATIVOS

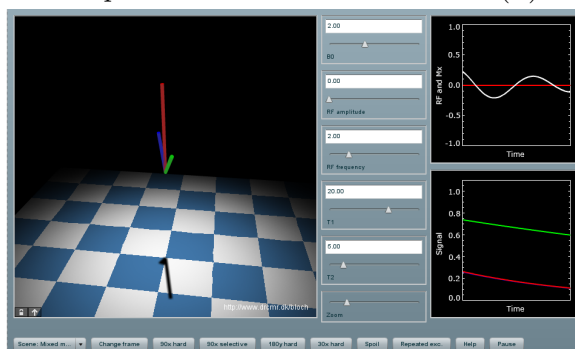
Los simuladores con fines educativos son en general utilizados como un soporte para ayudar a comprender conceptos complejos relacionados con la resonancia magnética. En este apartado veremos varios de estos simuladores, describiendo sus objetivos, detallando sus particularidades y hablando sobre sus características técnicas.

### 3.2.1 BLOCH SIMULATOR



(A) Precesión del espín.

(B) Gradiente.



(C) Espines distintos.

FIGURA 3.1: Simuladores educativos: Bloch Simulator. Las figuras (A), (B) y (C) muestran diferentes simulaciones posibles con el programa. *Las figuras han sido directamente obtenidas del uso del programa.*

Creado por [Hanson \(2007\)](#), es un simulador pensado para el aprendizaje de los conceptos básicos de NMR. A través de este simulador se puede trabajar sobre conceptos como los marcos giratorios empleados típicamente en NMR (Sección 2.2), el desfase de los espines o su reenfoque. Fue diseñado para su uso en el aula o para el trabajo personal de los estudiantes.

El simulador dispone de una interfaz gráfica sencilla, como puede verse en la Figura 3.1. Esta interfaz consta de:

- Panel de visualización principal. Se encuentra situado a la izquierda de la interfaz y proporciona al usuario una vista tridimensional de uno o varios espines, donde cada espín

viene representado por un vector de magnetización.

- Panel central. En este panel pueden modificarse distintas opciones relacionadas con la simulación. Los tres primeros parámetros se corresponden con el campo magnético principal ( $\vec{B}_0$ ) y las características del pulso de RF, es decir, su amplitud y su frecuencia. Los siguientes dos campos tienen que ver con las características magnéticas del espín, definidas por  $T_1$  y  $T_2$ .
- Ventanas bidimensionales. Se sitúan a la derecha de la interfaz. La primera ventana se encarga de representar dos señales: en color blanco se representa la señal de la componente  $x$  del vector de magnetización, mientras que el rojo es empleado para la señal de radiofrecuencia.
- Barra inferior. Esta barra cuenta con un desplegable donde pueden seleccionarse los diferentes tipos de simulación con los que cuenta esta herramienta: equilibrio, precesión (Figura 3.1a), fenómeno de “off resonance” (ver Sección 2.2), excitación, pulso RF aleatorio, inhomogeneidad de campo, gradiente (Figura 3.1b) o espines pertenecientes a distintos materiales (Figura 3.1c). El resto son botones para la simulación de pulsos RF (de 90, 180 y 30 grados), para introducir un gradiente encargado de eliminar la señal o para repetir una excitación previa.

De forma más detallada, la Figura 3.1a representa la precesión libre de un espín con un  $T_1$  infinito, la Figura 3.1b representa cómo los espines van desfasándose por la existencia de un gradiente a lo largo de un eje, mientras que en la Figura 3.1c se observa la relajación de tres espines con distintas propiedades magnéticas. Esta herramienta no tiene un flujo de uso determinado, sino que sencillamente es posible ir variando los diferentes parámetros, y ver cómo estos cambios se reflejan en las gráficas. En cuanto a la tecnología empleada para su implementación, este simulador hace uso de Adobe Flash Player. Eso permite que sea portable y ejecutable, aunque con algunas limitaciones muy importantes —que son comentadas en la Sección 4.5—, en entornos web. El programa puede verse en la siguiente *Uniform Resource Locator* (URL): <http://www.drcmr.dk/BlochSimulator/>.

### 3.2.2 SIMPLIFIED MRI

Simplified MRI fue diseñado como parte de un conjunto de simuladores más amplio para el aprendizaje de mecánica cuántica (McKagan et al., 2008). Su objetivo es mostrar cómo se comportan los vectores que describen el momento magnético nuclear. Como puede verse en la Figura 3.2, su interfaz está formado por dos pestañas similares:

- **Simplified NMR.** La Figura 3.2a muestra cómo estos vectores se alinean con un campo magnético externo. Esta alineación, para el caso del átomo de hidrógeno, puede ser en paralelo o antiparalelo. El simulador permite la opción de seleccionar otro átomo, pero cuyo número cuántico de espín sea el mismo. Permite también la introducción de un pulso de RF y la variación de su amplitud y frecuencia.
- **Simplified MRI.** La Figura 3.2b muestra un comportamiento muy similar a la primera pestaña. En este caso muestra los vectores en el interior de la representación de una cabeza.

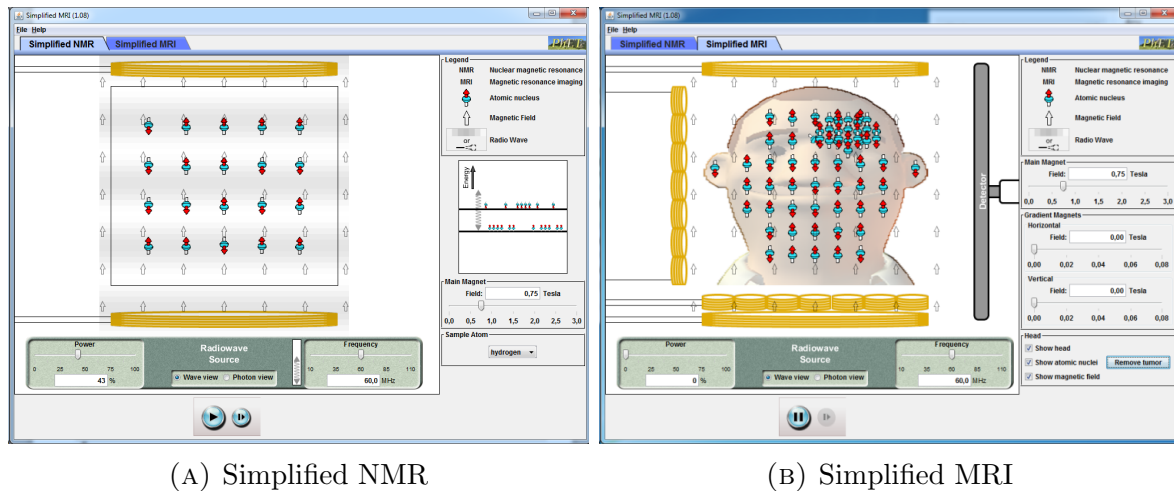


FIGURA 3.2: Simuladores educativos: Simplified MRI. Las figuras (A) y (B) muestran las dos pestañas disponibles: Simplified NMR y Simplified MRI. Las figuras han sido directamente obtenidas del uso del programa.

A esta se le puede añadir un tumor, y ver cómo la concentración de espines es mayor en esa zona. Otra opción que añade es la de variar la potencia de los gradientes de fase y de frecuencia.

Su flujo de funcionamiento es similar al de Bloch Simulator. Una vez seleccionada una de estas dos pestañas, pueden variarse los parámetros y observar el resultado. El orden en el que se varían los parámetros es indiferente. En cuanto a su implementación, el simulador se suministra de forma binaria y funciona sobre una máquina virtual de Java. El simulador puede descargarse en la siguiente URL: <https://phet.colorado.edu/en/simulation/mri>.

### 3.2.3 VIRTUAL MRI

*Virtual Magnetic Resonance Scanner* se describe como un simulador que imita algunas de las funciones y características de un escáner clínico de MR (Hackländer and Mertens, 2005). Su interfaz gráfica está dividida en cuatro paneles y una barra de herramientas, como puede verse en la Figura 3.3:

1. Panel de visualización, que se sitúa a la izquierda y ocupa gran parte de la interfaz. Este panel está destinado a la visualización de las imágenes simuladas.
2. Los botones situados en la parte superior derecha sirven para: cargar la referencia de la imagen sintética, cargar un nuevo modelo, guardar la imagen seleccionada e imprimir la imagen seleccionada.
3. Panel de pestañas situado justo debajo de los botones. Contiene cuatro pestañas que explicaremos brevemente. “Viewer” permite variar las opciones de visualización, como mostrar el panel dividido, mostrar la información de la imagen, ver el histograma o visualizar



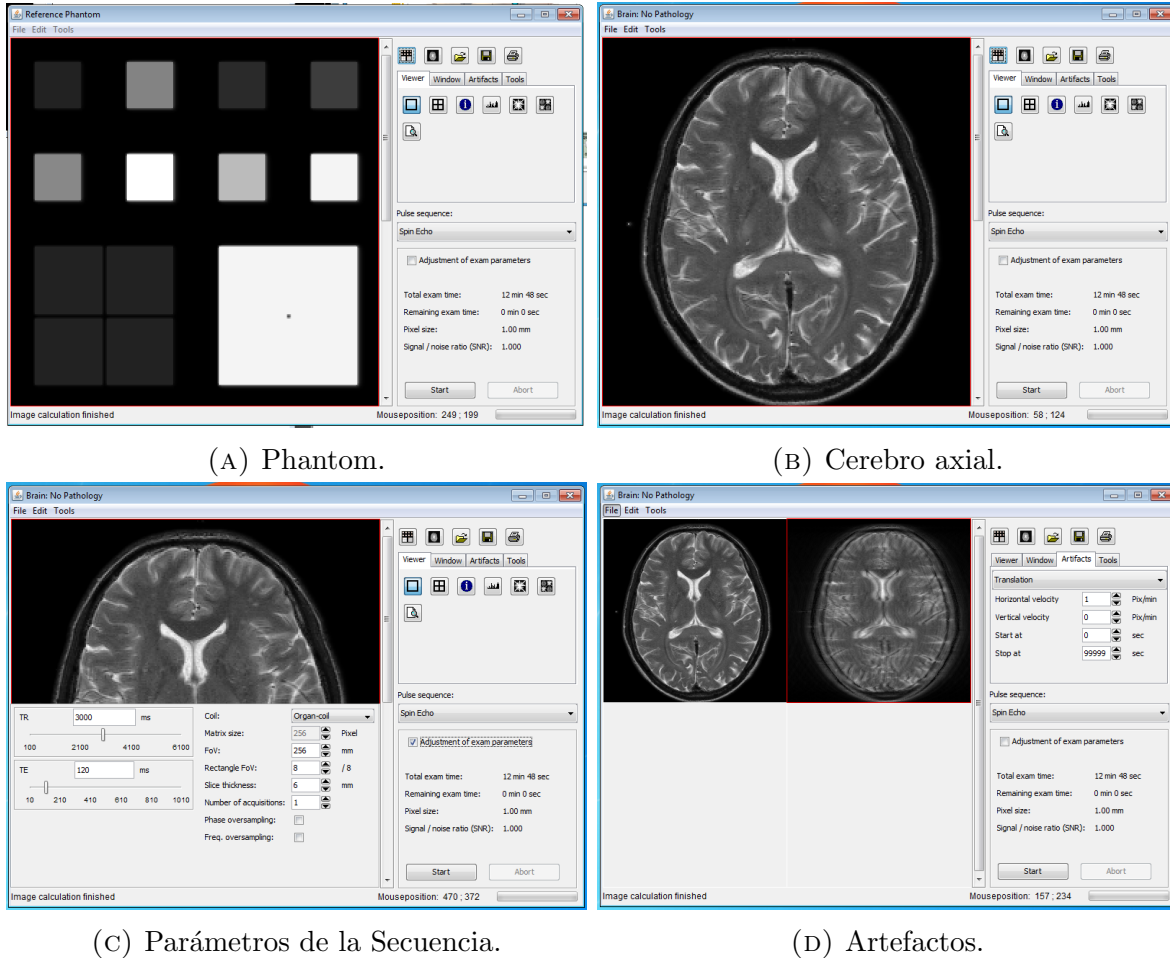


FIGURA 3.3: Simuladores educativos: Virtual MRI. Las figuras (A), (B), (C) y (D) muestran diferentes casos de uso del programa Virtual MRI. *Las figuras han sido directamente obtenidas del uso del programa.*

el espacio  $k$ . “Window” permite variar el brillo y el contraste de las imágenes. “Artifacts” permite la simulación de algunos artefactos de movimiento (Figura 3.3d). Finalmente, “Tools” da opciones sobre el comportamiento de la interfaz.

4. En la parte inferior derecha se pueden seleccionar las secuencias de pulsos y se pueden determinar sus parámetros pulsando al botón “Adjustment of exam parameters” (Figura 3.3c). Además, se muestran las informaciones sobre el examen y contiene los botones para iniciar o abortar la secuencia de pulsos.

El flujo previsto de uso del programa consiste en, primero, cargar un modelo, bien un phantom sintético (Figura 3.3a) o bien un corte axial de una imagen cerebral (Figura 3.3b). Una vez hecho esto, ya es posible pulsar directamente el botón “scan” para obtener la imagen simulada, puesto que Virtual MRI tiene fijadas una secuencia y unas opciones por defecto. Estas opciones pueden cambiarse una vez finalizada una adquisición, y se obtiene una nueva imagen pulsando el botón “scan” de nuevo. Las imágenes se van almacenando en orden en el visor (Figura 3.3d).

Su simulación permite la aparición de dos artefactos de movimiento, el transitorio y el pulsante. Por otro lado cuenta con varias secuencias de pulsos, a saber: spin echo, inversion recovery, saturation recovery, turbo spin echo, gradient echo y spoiled gradient echo. Sin embargo, en las pruebas realizadas, la secuencia de turbo spin echo no funcionó, la barra de progreso de la simulación se quedaba congelada y no avanzaba sin importar el tiempo esperado. Se realizaron varias pruebas, en distintos ordenadores y con largos tiempos de espera (mayores de 1 hora), obteniendo sistemáticamente el mismo resultado.

Virtual MRI está programado puramente en Java. Por lo tanto sólo requiere tener instalado una versión superior a la 1.2 del *Java Runtime Environment* (JRE). Para su instalación debe descargarse el archivo que contiene la versión 3.2.14 de Virtual MRI. El simulador dispone además de un manual donde se detalla su manejo.

Este simulador fue usado por estudiantes internos de medicina durante un año, donde ha ganado una amplia aceptación (Hackländer and Mertens, 2005). El software puede descargarse en la siguiente URL: <http://www.iftm.de/index.php/e-learning/virtual-mri/downloads>.

### 3.2.4 SIMULADOR DE TORHEIM

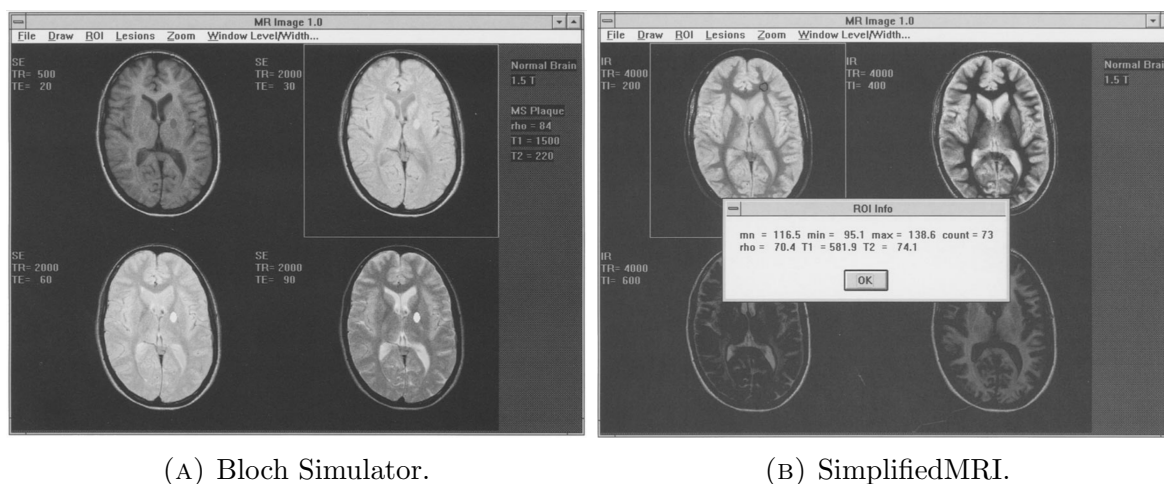


FIGURA 3.4: Simuladores educativos: Simulador de Torheim. Las figuras han sido obtenidas de Torheim et al. (1994)

Este simulador de MRI no cuenta con un nombre específico, por lo que en este documento nos referiremos a él como “Simulador de Torhteim” o “Torheim”.

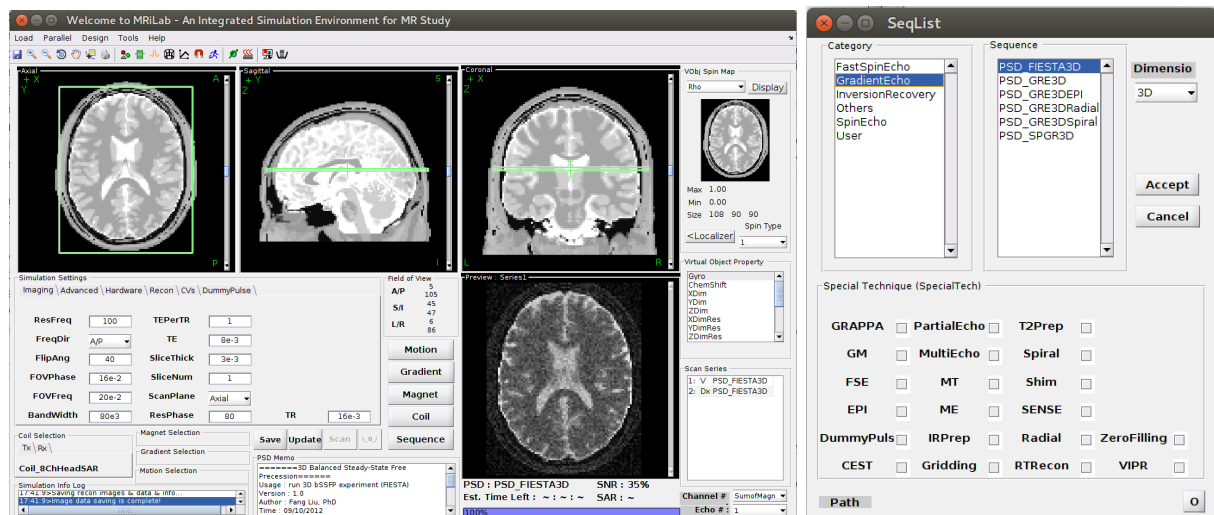
Esta herramienta fue desarrollada con fines didácticos y formativos por Torheim et al. (1994), y muestra un comportamiento muy similar Virtual MRI, si bien fue desarrollado bastantes años antes. Su interfaz (Figura 3.4) se compone por un visor principal dividido en 4 partes y una barra superior con diferentes menús. El menú “File” permite la carga de una secuencia, bien inversion recovery o bien spin echo, así como la selección de sus parámetros más relevantes (Figura 3.4a). Una vez cargada la imagen es posible dibujar una zona de interés y obtener los parámetros del modelo empleado en la simulación (Figura 3.4b). También se permite hacer zoom, añadir algunas lesiones simuladas al modelo, y la variación del brillo de la imagen. Este prototipo fue

programado en C++, y fue probado en clases y en sesiones individuales, obteniendo comentarios positivos sobre su uso (Torheim et al., 1994).

### 3.3 SIMULADORES EN LA INVESTIGACIÓN

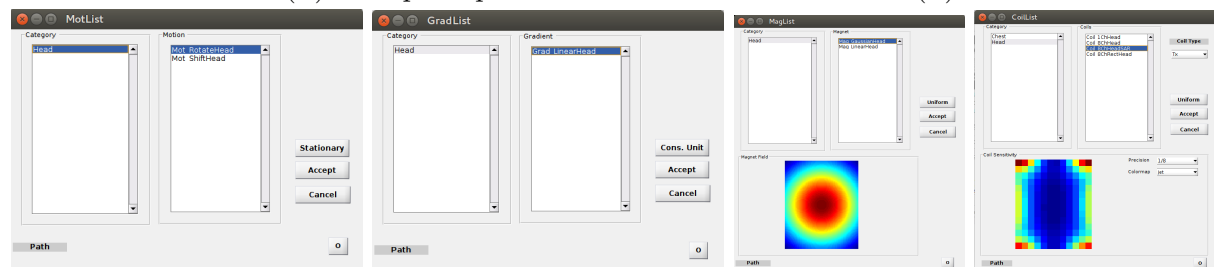
Los simuladores más orientados a la investigación, en contraposición con los educativos, centran sus desarrollos en herramientas que permitan la simulación de los fenómenos que suceden en MR de la forma más precisa posible. Para ello hacen uso de las ecuaciones de Bloch, tratando de enfrentarse al alto coste computacional que eso conlleva utilizando estrategias más o menos complejas. Varios de estos sistemas se recogen a continuación, y al igual que en el apartado anterior, se tratará de describir sus objetivos, particularidades y características técnicas.

#### 3.3.1 MRILAB



(A) GUI principal.

(B) Secuencias.



(C) Movimiento.

(D) Gradiente.

(E) Imán.

(F) Bobina.

FIGURA 3.5: Simuladores en la investigación: MRiLab. La figura (A) muestra la interfaz principal del programa, mientras que en las figuras (B), (C), (D), (E) y (F) se observan paneles secundarios accesibles desde el panel principal. Las figuras han sido directamente obtenidas del uso del programa.

MRiLab fue creado por Liu et al. (2013), y el proyecto se ha mantenido activo posteriormente (Liu et al., 2017). Actualmente es un simulador de MRI para ordenador con el propósito de mejorar el entendimiento y el desarrollo de las técnicas que se aplican en resonancia magnética.

Su simulación sigue la evolución discreta del giro del vector de magnetización de un tejido en un intervalo de tiempo determinado empleando las ecuaciones de Bloch, y puede ser acelerada mediante el uso de una *unidad de procesamiento gráfico o Graphic Processor Unit* (GPU). Además cuenta con una emulación numérica de equipos de resonancia magnética como las bobinas de recepción y emisión o los campos magnéticos generados por los imanes. Para su personalización, MRiLab proporciona herramientas dedicadas a analizar el pulso de RF, diseñar las secuencias de MR, configurar las bobinas, investigar las propiedades relacionadas con el campo magnético, visualizar la SAR y evaluar la técnica de creación de imágenes en tiempo real.

El manejo de MRiLab se realiza mediante una *interfaz gráfica de usuario o Graphical User Interfacel* (GUI) (Figura 3.5a). Esta interfaz se divide en varias partes:

- El localizador geométrico, formado por una vista axial, una sagital y una coronal del volumen. Este se sitúa en la parte superior de la pantalla y permite, de forma gráfica, seleccionar la zona que desea excitarse. Sin embargo, la región a seleccionar debe ser únicamente ortogonal, y sólo se permite la planificación sobre el modelo.
- Selector del modelo. Una vez cargado un modelo, este cuenta con distintos mapas. En el panel situado en la parte superior derecha, puede seleccionarse uno de estos mapas para ser cargado en el localizador geométrico.
- Opciones de simulación. Se compone por un panel con distintas pestañas. Sin embargo, el número de pestañas de este panel puede aumentar en función de las opciones seleccionadas en otros paneles como sucede con los “Dummy pulses”. A continuación se enumeran algunas de ellas:
  - **Imaging:** Son parámetros relevantes para la formación de la imagen, como su resolución, el campo que desea adquirirse o la configuración de los tiempos de la secuencia.
  - **Advanced:** Contiene opciones avanzadas relevantes para la adquisición de la imagen, como el *numero de excitaciones o Number of EXcitations* (NEX), el aliasing en las distintas direcciones o el Shimming con el que se trata de homogeneizar el campo magnético principal.
  - **Hardware:** Las especificaciones referentes a la máquina son modificadas aquí, como por ejemplo la pendiente de los gradientes, su grado de realismo (cómo de rápido puede activarse el gradiente) o el campo principal.
  - **Recon:** Las opciones de reconstrucción de la imagen son modificadas en esta pestaña. Puede elegirse por ejemplo la activación de la reconstrucción automática, el tipo de formato de la imagen de salida, la herramienta que se usa para la reconstrucción, etc.
  - **Cvs:** Estas siglas hacen referencia a variables controlables. Dichas variables se encuentran en cada una de las secuencias que han sido implementadas. Su nombre está siempre formado por “CV” mas el número de la variable.

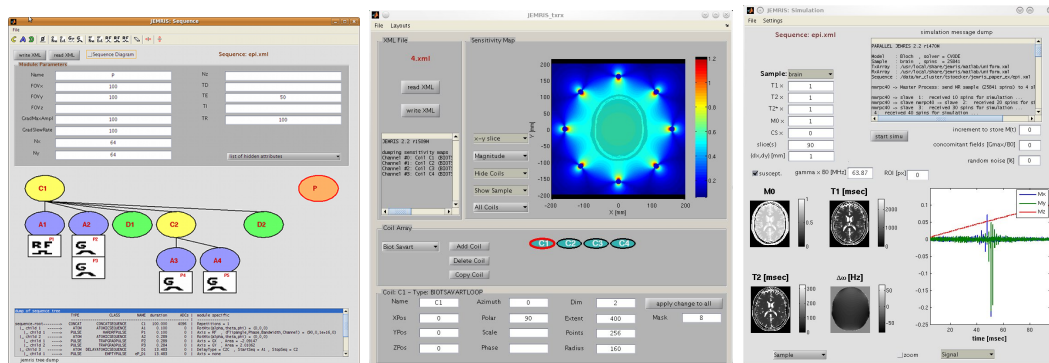
- **DummyPulse:** Controla las variables de los pulsos de RF generados al principio de la secuencia para llegar al estado estable del vector de magnetización.
- Grupo de botones situados a la derecha de las opciones de simulación:
  - **Motion:** Abre el panel de movimiento (Figura 3.5c) que selecciona le movimiento que se va a simular durante la adquisición.
  - **Gradient:** Abre el panel de gradiente (Figura 3.5d) que selecciona el tipo de gradiente.
  - **Magnet:** Abre el panel del imán (Figura 3.5e). Este panel activa una inhomogeneidad del campo magnético.
  - **Coil:** Abre el panel de la bobina (Figura 3.5f). Desde este panel puede seleccionarse la bobina que va a ser usada tanto en la recepción como en la trasmisión.
  - **Sequence:** Abre el panel de la secuencia (Figura 3.5b). Éste permite la selección del tipo de secuencia que va a aplicarse, como Fast Spin Echo o Inversion Recovery. Además, cuenta con la opción de seleccionar distintas modalidades en la adquisición, como EPI, o activar la adquisición en paralelo (GRAPPA o SENSE).
- Paneles de información. En la parte inferior cuenta con paneles que informan de las opciones seleccionadas.
- Visor del resultado. La visualización del resultado de la simulación se muestra en el cuarto visor. Además, en su parte derecha, aparecen las series que han sido adquiridas, y al pulsar en una de ellas es posible volver a cargar su resultado.

El flujo de funcionamiento del programa comienza por la carga de un modelo. Una vez cargado es posible elegir uno de los diferentes mapas del modelo en el localizador geométrico. A partir de este momento, en cada cambio realizado es necesario hacer clic en el botón “Update”. Tras seleccionar las secuencias, sus parámetros, las bobinas y el resto de opciones que se han mostrado, es posible hacer clic en el botón “Scan” para llevar a cabo la simulación y visualizar el resultado.

El sistema completo funciona sobre MATLAB. El peso de la simulación se ejecuta sobre programas escritos en C/C++ o en CUDA si se hace uso de las funciones de aceleración en GPU. Para la activación del procesamiento en paralelo es necesaria la configuración y compilación de parte del código, así como contar con una GPU Nvidia. Dispone, además, de un detallado manual de funcionamiento. El programa puede encontrarse en la siguiente URL: <http://mrilab.sourceforge.net/>.

### 3.3.2 JEMRIS

*Jülich Extensible MRI Simulator* (JEMRIS) es un programa abierto que proporciona un entorno de simulación y desarrollo de secuencias (Stöcker et al., 2010). Su desarrollo ha contado con una extensión externa para la simulación de fluidos en movimiento, además de los ya soportados tejidos estáticos (Fortin et al., 2014). Utiliza programación en paralelo para permitir



(A) Editor de secuencias de JEMRIS. (B) Editor de bobinas de JEMRIS. (C) Simulador de JEMRIS.

FIGURA 3.6: Simuladores en la investigación: JEMRIS. Las figuras (A), (B) y (C) muestran las interfaces de las 3 herramientas que componen JEMRIS. Las figuras han sido directamente obtenidas del uso del programa.

su escalado desde un ordenador con varios núcleos a un cluster de computación. En cuanto a la simulación, permite la introducción de diversos gradientes, distintas geometrías de bobinas así como el cálculo del desplazamiento químico, el desfase entre espines debido al efecto de la inhomogeneidad de campo y otra serie de procesos que surgen en la resonancia magnética. Esta simulación hace uso de las ecuaciones de Bloch para el seguimiento del vector de magnetización a lo largo del tiempo.

Su interfaz gráfica está formada por un conjunto de tres herramientas independientes (Figura 3.6):

- Herramienta para desarrollo de secuencias mediante el concepto de árbol (Figura 3.6a). En esta interfaz pueden, por un lado, introducirse los parámetros por defecto de la secuencia, tanto las dimensiones de la imagen como los tiempos. Por otro lado, pueden seleccionarse los pulsos y el orden en el que se aplican. Por ejemplo, en la Figura 3.6a el elemento C1 representa la secuencia; A1 se corresponde con el pulso de RF y A2 con los gradientes fase y frecuencia. Finalmente D1 indica un tiempo de espera. A partir de la secuencia C1 se elabora otra secuencia, esta vez destinada a la escritura y llamada C2, de la cual cuelgan a su vez los gradientes de escritura. El tiempo D2 es el tiempo que transcurre hasta la siguiente iteración del pulso de secuencias. El resultado de todo este proceso se codifica en forma de un *lenguaje de marcas extensible* o *Extensible Markup Language* (XML).
- Herramienta para el diseño de bobinas (Figura 3.6b). En este caso se puede definir el número de bobinas, y para cada una de ellas fijar su posición, su inclinación y su sensibilidad. Cuenta además con un esquema superior en el que puede verse cómo queda la sensibilidad de la bobina. Al igual que en la primera herramienta vista, el resultado genera un fichero XML.
- Herramienta para la ejecución de la simulación (Figura 3.6c). En esta utilidad deben primero cargarse las secuencias y las bobinas creadas previamente, aunque cuenta con una secuencia EPI ya generada. Es necesario además elegir el modelo sobre el que se va a



realizar la simulación, junto con sus parámetros. Una vez elegidos la bobina, la secuencia y el modelo, podría comenzarse la simulación. El resultado de esta simulación se muestra en la parte inferior derecha de la interfaz.

Este conjunto de herramientas hace uso de la MATLAB para su funcionamiento gráfico, aunque el núcleo de simulación está programado en C++. Para la ejecución de ese núcleo se requiere la compilación del código. El programa puede encontrarse en la siguiente URL: [http://www.jemris.org/ug\\_downl.html](http://www.jemris.org/ug_downl.html).

### 3.3.3 SIMRI

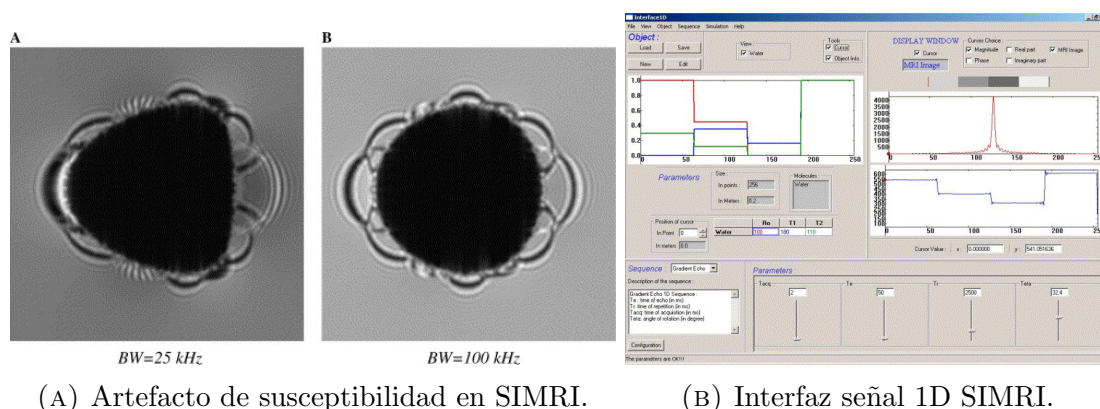


FIGURA 3.7: Simuladores en la investigación: SIMRI. *Las figuras han sido obtenidas de Benoit-Cattin et al. (2005)*

SIMRI es un conjunto de funciones desarrolladas en C por Benoit-Cattin et al. (2005). Hace uso de las ecuaciones de Bloch para la simulación secuencias como SE o GRE. También permite la simulación de campos estáticos paralelos, el desplazamiento químico o la susceptibilidad magnética (Figura 3.7a). La nomenclatura de las funciones es descriptiva, así como el nombre usado para las constantes. Para operaciones de gran carga computacional hace uso de una programación en paralelo que aproveche la computación en malla. Para su ejecución es necesaria, en primer lugar la instalación de la biblioteca SIMRI, la creación del script y su compilación.

Además, cuenta con una interfaz para el estudio de señales 1D que se ejecuta como un programa portable (Figura 3.7b). Este programa permite la simulación del comportamiento de un espín a lo largo de una secuencia de pulsos y está preparado para su funcionamiento en Windows. La evolución del espín en el tiempo puede observarse en una gráfica 2D. El código puede encontrarse en la siguiente URL: <https://sourceforge.net/projects/simri/>

### 3.3.4 MRISIMUL

MRISIMUL es una biblioteca diseñada y creada por Xanthis et al. (2013). El objetivo de su desarrollo es la realización de una simulación de MR que no haga asunciones con respecto a la

secuencia de pulsos. Para su cumplimiento, ejecuta una simulación paso a paso de las ecuaciones de Bloch y ha sido probada sobre tres modelos 3D, a saber: un phantom, un cerebro y un corazón humano.

Por otro lado, la implementación hace uso de la programación en GPU mediante CUDA-C consiguiendo reducir el tiempo de simulación considerablemente. Por ejemplo, en un volumen 3D con 1.350.000 isocromas, el tiempo de simulación es de aproximadamente 5 minutos usando una GPU Tesla C2070, mientras que en *unidad de procesamiento central o Central Processing Unit* (CPU) con 8 núcleos, este tiempo se eleva por encima de las 2 horas. En contraposición, el uso de esta tecnología hace que el hardware deba tener una GPU de la marca Nvidia.

Su utilización ha demostrado ser interesante para la simulación de artefactos debidos al movimiento o al flujo en adquisiciones cardiovasculares o embrionarias (Kantasis et al., 2015). En este caso utiliza un cluster de varios nodos, donde en cada nodo se ejecuta parte de la simulación. Por desgracia, este simulador no ha podido ser probado dado que su código no está disponible en ningún repositorio.

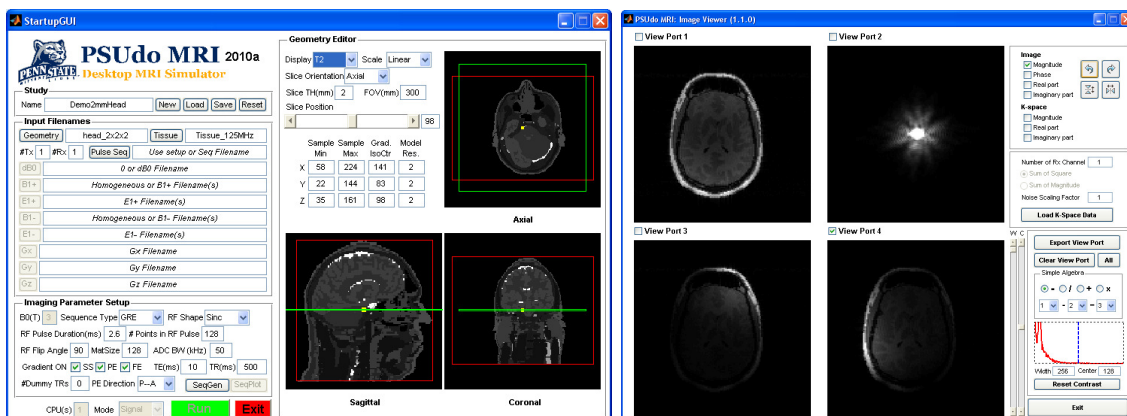
### 3.3.5 PSUDoMRI

PSUDoMRI es otro simulador de MRI que hace uso de las ecuaciones de Bloch utilizando el marco rotatorio de referencia. Fue creado por Cao et al. (2010), y ha ido evolucionando como se muestra en los artículos Oh et al. (2011) y Cao et al. (2014). PSUDoMRI se centra en una simulación realista de los campos magnéticos para el cálculo de la señal, el ruido y la SAR. En sus resultados, compara la simulación realizada por ellos con respecto a la derivada de primer orden que describe la ecuación de Bloch, concluyendo que la simulación se corresponde con el resultado teórico.

Cuenta con una GUI desarrollada en MATLAB aunque el núcleo de la simulación está implementado en C/C++. Esta interfaz se divide en 3 herramientas independientes, como muestra la Figura 3.8:

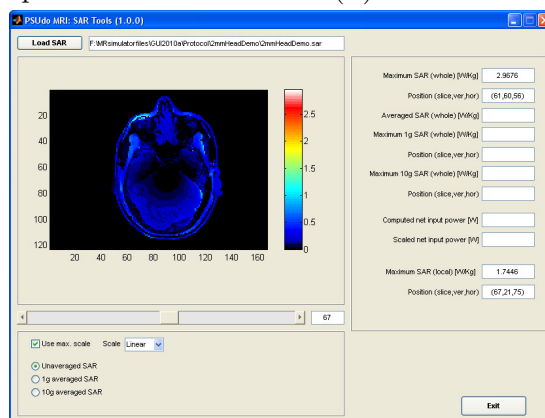
- Panel principal dividido en tres partes (Figura 3.8a):
  - Panel de introducción de los parámetros y los scripts. Abarca casi por completo la parte lateral izquierda y permite la selección de los modelos, las secuencias y sus parámetros, así como otras opciones relacionadas con las propiedades físicas del escáner, como el campo magnético, los gradientes, etc.
  - En la parte derecha se encuentra la planificación del corte que va a ser excitado. Su selección se realiza mediante la introducción numérica de los parámetros “Sample Min” y “Sample Max”. La orientación del slice debe sin embargo permanecer siempre ortogonal a los ejes x, y, z.
  - En su barra inferior cuenta con dos botones para la ejecución de la simulación y el cierre del programa. Además cuenta con un indicador del número de CPUs que se están utilizando. Sin embargo, el elemento más interesante es el del modo. Cuenta con tres modalidades: señal, ruido y SAR. El modo de señal genera una imagen de contraste con el resultado de la simulación. En el modo de ruido, el resultado es un mapa con el ruido en cada una de las bobinas. Finalmente, para el modo SAR son





(A) Panel principal PSUdoMRI.

(B) Reconstrucción paralela PSUdoMRI.



(c) Visualización del SAR PSUdoMRI.

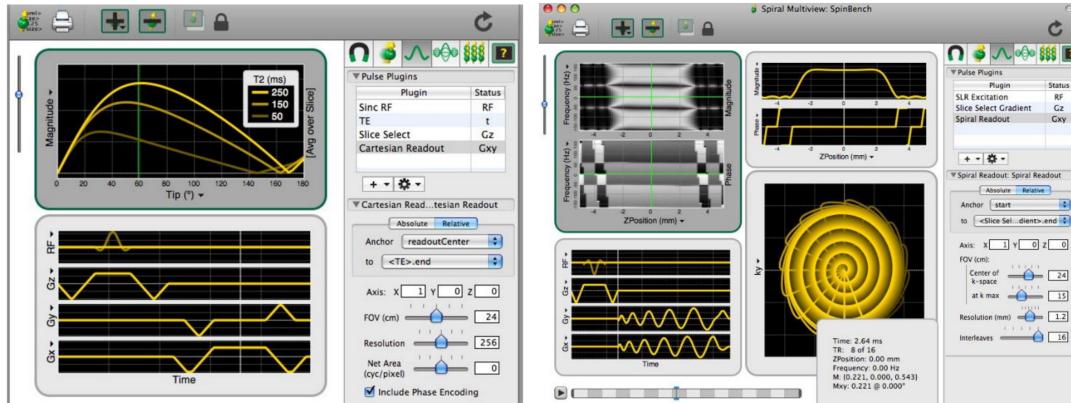
FIGURA 3.8: Simuladores en la investigación: PSUdoMRI. La figura (A) muestra la interfaz de la herramienta principal, mientras que las figuras (B) y (C) muestran herramientas para la reconstrucción de la imagen. La figura (A) ha sido directamente obtenida del uso del programa PSUdoMRI. Las figuras (B) y (C) han sido tomadas de la documentación de PSUdoMRI, ver URL: <https://cai2r.net/resources/software/psudomri-simulator>

necesarios el campo magnético  $B_1$  y el campo eléctrico asociado  $E_1$ . El resultado genera nuevas imágenes de ruido para cada una de las bobinas.

- Panel de reconstrucción (Figura 3.8b). Sirve principalmente para la visualización de las imágenes generadas en la simulación. Cuenta con 4 visores, y al seleccionar uno se puede elegir lo que se desea ver en él. Permite ver las imágenes de magnitud o de fase de cada una de las bobinas o el resultado de su reconstrucción. Lo mismo sucede con el espacio k.
- Panel SAR (Figura 3.8c). Permite la carga de una imagen con la SAR de un volumen para su visualización.

El programa proporciona secuencias de tipo Spin Echo, Gradient Echo, y un modelo de cerebro para su prueba. Además, incluye un breve manual de uso. El archivo ejecutable puede encontrarse en la URL: <https://cai2r.net/resources/software/psudomri-simulator>.

### 3.3.6 SPINBENCH



(A) Generación de la señal SpinBench. (B) Escritura de la señal SpinBench.

FIGURA 3.9: Simuladores en la investigación: SpinBench. La figura (A) muestra la amplitud de la señal para tres tejidos a lo largo del tiempo para una secuencia dada. La figura (B) muestra la trayectoria seguida en el espacio  $k$  para una secuencia de adquisición dada. Las figuras han sido directamente obtenidas del uso del programa.

SpinBench es un programa creado por Overall and Pauly (2007) en la Universidad de Stanford. Está pensado para la simulación de secuencias de pulsos y ha sido desarrollado usando el software XCode. Este programa no requiere instalación, pero sólo funciona bajo el sistema operativo Mac OS.

Su interfaz gráfica (Figura 3.9) permite el diseño y la creación de secuencias de pulsos sin la necesidad de programación. A la hora de diseñar los pulsos, su GUI permite la selección de los distintos gradientes mientras se ve cómo la magnitud de la señal evoluciona a lo largo del tiempo (Figura 3.9a). Además cuenta con opciones centradas en la escritura de la señal, donde pueden visualizarse varios aspectos interesantes como la trayectoria del espacio  $k$ , la variación de los gradientes o la magnitud de la señal (Figura 3.9b). Por otro lado, cuenta con herramientas para la visualización de la red de magnetización de forma similar a Bloch Simulator Hanson (2007) y la selección de las propiedades del imán principal.

En contraposición a otros simuladores, el resultado de la simulación describe la evolución de la componente transversal del vector de magnetización en gráficas unidimensionales. En algunos casos, sin embargo, también hace uso de gráficas bidimensionales, para la trayectoria seguida en la lectura del espacio  $k$ . Sin embargo, no permite realizar una simulaciones que generen imágenes de contraste. Puede descargarse en la URL: <https://en.freedownloadmanager.org/Mac-OS/SpinBench-FREE.html>.

### 3.3.7 BLOCHSOLVER

BlochSolver es un simulador de imágenes de MRI desarrollado por Kose and Kose (2017). Como su nombre indica, se basa en la ecuación de Bloch resolviéndola de forma numérica.

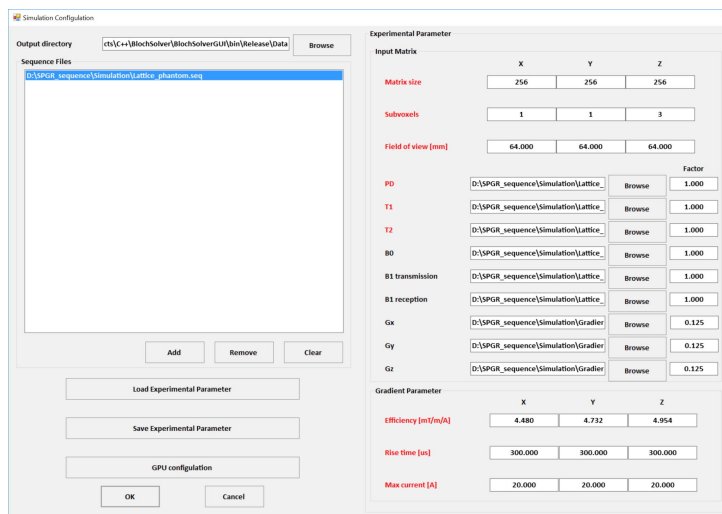


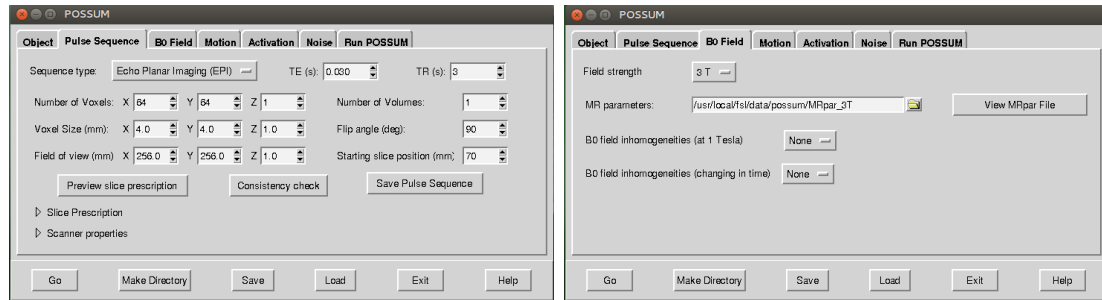
FIGURA 3.10: Simuladores en la investigación: BlochSolver. *Figura obtenida obtenida de Overall and Pauly (2007)*

Este simulador ha sido programado usando la tecnología CUDA y en su desarrollo se implementaron 3 secuencias de pulsos: GRE tridimensional, spoiled-GRE tridimensional y GRE multislice. Su rendimiento fue evaluado mediante la ejecución del simulador en dos dispositivos. El primer dispositivo se trataba de un montaje en paralelo de dos GPUs modelo GTX 1080. En el segundo, el sistema se componía CPU Xeon de 18 núcleos. El resultado mostró que la ejecución en GPU fue 14 veces más rápida que en CPU. De ello, [Kose and Kose \(2017\)](#) concluyen que este tipo de simuladores se convertirán en herramientas indispensables tanto en la investigación como en el desarrollo de MRI.

Para facilitar el lanzamiento de la ejecución, BlochSolver incluye una interfaz (Figura 3.10). Esta GUI permite la selección de los diferentes scripts que contienen las secuencias, el modelo sobre el que se va a ejecutar, el tamaño de la matriz y la selección de los gradientes y de los campos magnéticos.

### 3.3.8 POSSUM

POSSUM (*Physics-Oriented Simulated Scanner for Understanding MRI*) es un simulador creado por [Drobnjak et al. \(2006\)](#). Su objetivo es la investigación de algunos artefactos de MR, como el de movimiento rígido, las inhomogeneidades del campo principal, el desplazamiento químico y las corrientes de Foucault en *resonancia magnética funcional o functional magnetic resonance* (fMRI). Utiliza las ecuaciones de Bloch junto a un modelo geométrico de cerebro que incluye una imagen *dependiente del nivel de oxigenación sanguínea o Blood Oxygenation Level Dependent* (BLOD). Posteriormente, se mejoró la simulación de los campos magnéticos transitorios que producen cambios en las imágenes de MR ([Drobnjak et al., 2010](#)). Varias publicaciones hacen uso de POSSUM para la comparación directa y cuantitativa de métodos de postprocesado para imágenes de difusión ([Graham et al., 2016](#)) y evaluación de la corrección de estas imágenes ([Graham et al., 2017](#)).



(A) Secuencia de pulsos POSSUM.

(B) Campo B0 POSSUM.

FIGURA 3.11: Simuladores en la investigación: POSSUM. La figura (A) muestra las opciones disponibles para la secuencia de pulsos, mientras que la figura (B) muestra las opciones disponibles para establecer el campo  $B_0$ . Las figuras han sido directamente obtenidas del uso del programa.

Esta herramienta forma parte de la *biblioteca de software FMRIB's o FMRIB's Software Library (FSL)*. POSSUM cuenta con una interfaz (Figura 3.11) compuesta por diversas pestañas. En cada una de estas pestañas puede seleccionarse una serie de parámetros relacionados con la simulación:

- **Object.** Permite la selección del modelo sobre el que va a realizarse la simulación. Por defecto, *escáner simulado orientado a la física para comprender la resonancia magnética o Physics-Oriented Simulated Scanner for Understanding MRI (POSSUM)* proporciona el volumen de un cerebro.
- **Pulse Sequence** (Figura 3.11a). Permite la selección de la secuencia de pulsos y sus propiedades. Cuenta con dos secuencias implementadas: EPI y SE, aunque es posible añadir más. En esta pestaña también es posible elegir el corte y otras propiedades del escáner, como el BW.
- **B0 Field** (Figura 3.11a). Permite la selección del modelo del campo magnético principal.
- **Motion.** Permite la activación del artefacto de movimiento mediante la selección de un fichero.
- **Activation.** Permite la activación de distintas zonas del modelo de cerebro. Para su simulación utiliza una función temporal que hace variar el valor  $T_2^*$  a lo largo del tiempo en una región determinada del modelo.
- **Noise.** Permite que la adquisición tenga o no ruido térmico.
- **Run POSSUM.** Permite elegir el directorio de salida, el número de procesadores utilizados y cada cuánto tiempo guardar la evolución de la simulación.

El flujo de control de la aplicación no tiene por qué tener un orden determinado y contiene un valor por defecto para todos los parámetros, por lo que es posible iniciar una simulación sin la necesidad de variar ningún elemento. Puede instalarse siguiendo los pasos descritos en la siguiente URL: <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL>.

### 3.3.9 ODIN

*Interfaz de desarrollo orientada a objetos para NMR o Object-Oriented Development Interface for NMR* (ODIN) es una herramienta de simulación creada por Jochimsen and Von Mengershausen (2004). Su objetivo es la producción y desarrollo de nuevas secuencias de pulsos. Al igual que el resto de simuladores, realiza un seguimiento de la magnetización usando las ecuaciones de Bloch. Como resultado, este programa ha sido utilizado en diferentes artículos para el desarrollo de una herramienta para la simulación de un sistema SPECT/MRI (Chen et al., 2007), para el estudio de las variaciones en las imágenes de contraste y su optimización (Van Reeth et al., 2016) y para el uso de teorías de control óptimo para diseñar pulsos de radiofrecuencia (Lefebvre et al., 2017).

Incorpora una interfaz gráfica escrita en Qt (Figura 3.12) y que en su última versión está formada por cuatro herramientas:

- **ODIN** (Figura 3.12a). Es el panel principal y da acceso al resto de utilidades. En él es posible la variación de los parámetros de la secuencia para la generación de la señal y para su escritura.
- **Geoedit** (Figura 3.12b). Es un editor para seleccionar la geometría sobre la que se desea realizar la simulación. La interacción se realiza mediante el deslizamiento de las barras que se encuentran en la parte inferior, y es posible conseguir cualquier orientación.
- **Pulsar** (Figura 3.12c). Es una interfaz para la generación y simulación de pulsos de radiofrecuencia.
- **MiView** (Figura 3.12e). Sirve al propósito que su nombre indica: visualizar las distintas imágenes generadas permitiendo la variación de su brillo y su contraste.

El primer paso para su funcionamiento es la selección del archivo de la secuencia que va a emplearse. En su carga, ODIN compila este archivo y actualiza la interfaz con los distintos parámetros disponibles para esta secuencia en particular. Finalizada la carga, ya es posible realizar la planificación geométrica o ver la secuencia de pulsos y, cuando los parámetros sean los adecuados, se puede lanzar la simulación. Durante la ejecución, ODIN genera una ventana gráfica que muestra la parte de la secuencia que se está calculando en ese instante (Figura 3.12d). Al finalizar, se ejecuta la ventana “MiView” para la visualización del resultado. Los modelos que proporciona son tres cerebros iguales con distinta resolución y otros tres phantoms iguales con distinta resolución. Ambos modelos deben ser descargados de forma independiente. En cuanto a las secuencias de pulsos, ODIN cuenta con varias ya implementadas, como EPI, *Rapid Acquisition with Relaxation Enhancement* (RARE), *Spectral Presaturation with Inversion Recovery* (SPIR) o *Fast Imaging with Steady-state Precession* (FISP).

Finalmente, ODIN puede funcionar bajo diferentes sistemas operativos: Windows, Mac OS y GNU/Linux. Se suministra de forma binaria y el código es abierto. Para su uso en Linux es necesaria la compilación del código con bibliotecas actuales. Sin embargo, la ejecución del binario en Windows permitió su uso inmediato. En Mac OS no ha podido confirmarse su funcionamiento. No se ha encontrado ningún manual para su uso. El programa puede descargarse en la siguiente URL: <https://sourceforge.net/projects/odin/>.





## 3.4 DISCUSIÓN Y CONCLUSIÓN

Como se ha podido comprobar a lo largo de este capítulo, durante los últimos años se han propuesto esencialmente dos tipos de simuladores de MR. En primer lugar se encuentran aquellos que están enfocados a un propósito educativo para personas que no tienen un conocimiento avanzado sobre los fundamentos físicos de la resonancia. Por otro lado, hay un conjunto de simuladores pensados para realizar un análisis profundo del fenómeno de la resonancia magnética, orientados a tareas técnicas como el diseño de secuencias o el análisis de la sensibilidad de bobinas.

Puesto que los simuladores de perfil educativo están destinados a un público sin conocimientos técnicos, hacen uso de una GUI simple, propiciando, en la medida de lo posible, su sencillez en el uso. Además, en su conjunto, procuran hacer el acceso a su sistema lo más fácil posible, bien usando sistemas web o programas portables.

Otro aspecto relevante observado es que las simulaciones que realizan suelen ser rápidas. Esta característica atiende a razones cognitivas, donde reducir los tiempos implica minimizar la latencia que existe entre la petición de la simulación y el resultado. Este parece ser un requisito esencial en este tipo de simuladores, puesto que cualquier proceso de aprendizaje requiere de una cierta agilidad en el manejo de dichas herramientas. No resultaría viable una espera de muchos minutos o incluso horas para comprobar los resultados de una simulación.

En el caso de Bloch Simulator, la señal simulada es unidimensional, por tanto, el cálculo de la evolución del espín a lo largo del tiempo puede realizarse a la par que su visualización. Por otro lado, en el caso de Simplified MRI, el concepto en el que se centra es muy concreto y su simulación es meramente ilustrativa.

En general, la consecución de tiempos de simulación cortos conlleva algún tipo de contrapartida. Éste es el caso de Virtual MRI y de el simulador de Torheim, donde la simulación se simplifica asumiendo secuencias de pulsos ideales. Este tipo de simulación, como alternativa a una simulación paso a paso de las ecuaciones de Bloch, reduce enormemente el coste computacional de la simulación y, por tanto, el tiempo. En contraposición, las imágenes producidas tienen un error mayor con respecto a las imágenes generadas mediante una simulación más detallada (Kenta et al., 2009).

Referente a los simuladores centrados en investigación (Sección 3.3), el diseño de secuencias de pulsos es la aplicación más frecuente. Una excepción es el simulador POSSUM, que está orientado a la simulación de fMRI, una modalidad específica y no convencional de la MR y que representa características y complejidades particulares. No obstante, POSSUM habilita la escritura de secuencias de pulsos propias para comprobar su funcionamiento en la simulación.

Analizando la manera en la que las distintas herramientas observadas afrontan la simulación, es posible generalizar el proceso como se muestra en el esquema de la Figura 3.13. Según este esquema, la simulación se estructura alrededor de cinco bloques:

- **Opciones de la simulación:** Este bloque abarca todo lo referente a los parámetros que se seleccionan previamente a la simulación, tales como: el pulso de secuencias elegido y sus propiedades, la bobina, el modelo (ver siguiente punto) sobre el que se realiza, la zona anatómica, la existencia de artefactos y su configuración, el tamaño de la matriz de reconstrucción, etc. Habitualmente, estos parámetros son proporcionados a través de una GUI por el usuario. Esta interfaz puede ser sencilla y estar compuesta únicamente por

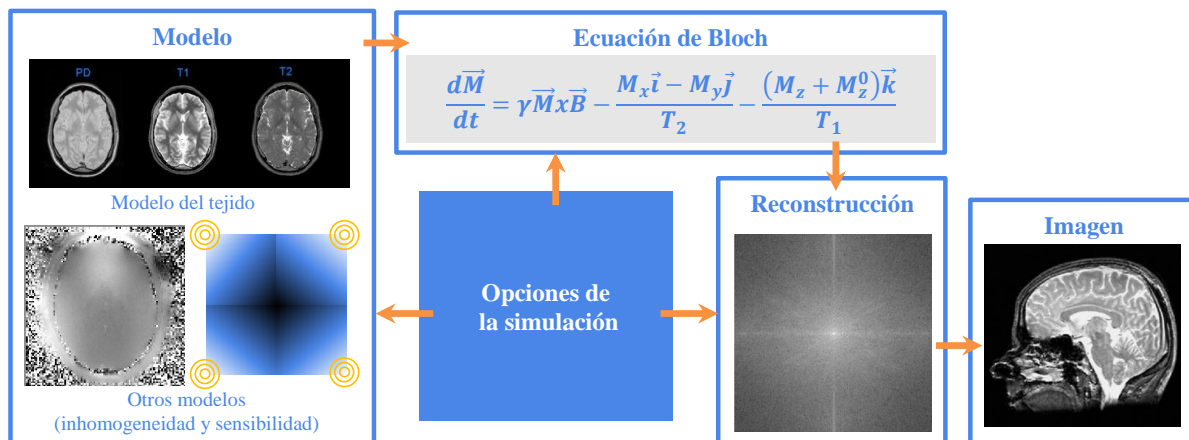


FIGURA 3.13: Bloques en la simulación de resonancia, a saber, el modelo, las ecuaciones de Bloch, la reconstrucción y la imagen. El modelo está compuesto por las características del tejido así como por la descripción de los campos magnéticos. El bloque de las ecuaciones de Bloch hace referencia a las formas de aplicar éstas en una simulación. El tercer bloque consiste en la reconstrucción de la imagen después de la aplicación de las ecuaciones. El bloque de imagen hace referencia al resultado de la simulación.

campos en los que se introducen los datos manualmente –como en el caso de *BlochSolver* o *POSSUM*–, o bien ser más detallada y contar con soporte gráfico que ayude a entender mejor los parámetros introducidos, e incluir además herramientas de edición –como *SpinBench*, *JEMRIS*, *PSUDO-MRI*, *ODIN* o *MRiLab*–. Finalmente, en otros simuladores es necesario introducir esta configuración mediante línea de comandos, como en *SIMRI*, o mediante la programación de un script, como en *MRISIMUL*.

- Modelo:** Nos referimos con este término a la información sobre los parámetros físicos del tejido cuya respuesta a la MR se pretende simular. Los parámetros básicos son los tiempos de relajación  $T_1$  y  $T_2$ , así como la densidad de protones (PD) de una determinada región anatómica. Estos parámetros se almacenan habitualmente en sendas matrices de las mismas dimensiones. Las dimensiones de cada matriz habituales son: 2D en el caso de imágenes o slices; 3D en el caso de volúmenes que contienen una zona anatómica determinada (cerebro, una rodilla, etc); o 4D en el caso de volúmenes que varían en el tiempo, como en el caso de resonancia cardiaca. En el caso de pretender la realización de simulaciones de otros tipos no convencionales de resonancia, estos modelos podrían tener dimensiones distintas, como en el caso de la resonancia de difusión o la espectroscopía. Finalmente, el modelo puede contener una gran cantidad adicional de información referente a otras propiedades físicas de interés, como por ejemplo el campo magnético principal y sus inhomogeneidades, o la sensibilidad de cada una de las bobinas.
- Ecuaciones de Bloch:** Las ecuaciones de Bloch, como vimos en la Sección 2.2, modelan la variación de los vectores de magnetización intra voxel. En una simulación pueden diferenciarse habitualmente dos partes: por un lado, la generación de la señal, y por otro, la lectura de la misma. Existen secuencias especializadas en ambas partes y en función de lo que se quiera demostrar, puede ser interesante simular únicamente una de estas partes o



ambas. Por lo tanto, el resultado de una simulación podría ser una imagen compleja que representaría la componente transversal del vector de magnetización, o bien el espacio  $k$  generado en la adquisición de una de las bobinas. Las ecuaciones de Bloch se aplican de tres formas distintas en los simuladores investigados:

1. Una formulación sencilla, que es la empleada habitualmente en los simuladores educativos como *Virtual MRI*. Se usa la expresión algebraica obtenida de una secuencia de pulsos ideal mediante la resolución de las ecuaciones de Bloch. Este tipo de simulación, sin embargo, no permite incluir efectos como los artefactos de *off-resonance* o el desplazamiento químico, al menos directamente.
2. Una simplificación donde la ecuación de Bloch se divide en dos operaciones [Treceño Fernández et al. \(2016\)](#). La primera, una *rotación* tridimensional del vector de magnetización, que dependerá tanto del campo magnético como del tiempo durante el cual el vector de magnetización esté sometido a dicho campo. Y la segunda, un *escalado* de la señal gobernado por las constantes  $T_1$  y  $T_2$  y debido al campo magnético principal. Este tipo de simulación proporciona un error pequeño con respecto a la teoría, como se prueba en [Cao et al. \(2014\)](#), y se utiliza en la mayoría de los simuladores vistos: *POSSUM*, *ODIN*, *JEMRIS* o *SIMRI*. Sigue teniendo sin embargo algunas limitaciones, como que no es capaz de simular los ecos estimulados [Zhi-Pei Liang \(2000\)](#).
3. La resolución de la ecuación diferencial de Bloch paso a paso, mediante cualquiera de los métodos numéricos disponibles. Es el caso de *MRISIMUL* y *BlochSolver*.

Finalmente, en este bloque, otros aspectos físicos del hardware pueden ser tenidos en cuenta, como la sensibilidad de las bobinas o el ruido procedente del sistema.

- **Reconstrucción:** Dependiendo de la simulación, pueden darse distintos casos que requieran la necesidad de un proceso de reconstrucción. Dos ejemplos de esta situación se producen cuando el tamaño de la matriz adquirida difiera con el tamaño de la matriz de salida, por la aplicación de una técnica de aceleración, o bien que se apliquen algoritmos para la eliminación de artefactos, como los originados por las corrientes de Foucault. En este bloque, los simuladores aplican herramientas para reconstruir la adquisición, rellenando el hueco existente entre la generación de la señal y la imagen resultante. Las herramientas para ello pueden ser propias, como sucede con *PSUdoMRI*, o hacer uso de otras aplicaciones externas como como Gadgetron ([MS and TS, 2012](#)) en el caso *MRiLab*.
- **Imagen:** Se refiere a la imagen generada por la simulación, y en ocasiones, abarca una herramienta que el simulador proporciona para su inspección. Esta utilidad puede encontrarse integrada en la interfaz –como en *JEMRIS* o *MRiLab*– o bien ser una ventana distinta –como en *ODIN* o *PSUdoMRI*–.

La aplicación de las ecuaciones de Bloch en sus versiones 2) y 3), así como el empleo de algoritmos de reconstrucción, tienen un coste computacional muy elevado. Este coste, además, se ve incrementado en función del modelo, ya que cuanto mayor sea el tamaño de sus dimensiones, mayor será el coste. Como consecuencia, los tiempos de simulación son muy elevados. Con el objetivo de reducir estos tiempos se ha hecho uso de técnicas de clustering para la computación

en paralelo –como en *JEMRIS* o *SIMRI*– o se han implementado soluciones que hacen uso de GPUs –como son el caso de *BlochSolver*, *MRISIMUL*, *POSSUM* o *MRiLab*–.

Por otro lado, los simuladores dirigidos al mundo investigador asumen que su público va a contar con elevados conocimientos técnicos. De hecho, algunos de los simuladores vistos son sólo un conjunto de funciones que deben ser invocadas mediante código como *MRISIMUL*. Otros, además, necesitan la instalación de bibliotecas y la compilación de su código fuente para ser utilizados en algunos sistemas operativos como *ODIN* o *JEMRIS* en el caso de GNU/Linux. Y varios dependen para su ejecución de MATLAB, como sucede en simuladores *MRiLab*, *PSUdoMRI* y *JEMRIS*.

Finalmente, en ninguno de los simuladores enumerados a lo largo de este capítulo se ha evaluado su facilidad de uso o su utilidad de manera cuantitativa. Esto es comprensible en el caso de los simuladores enfocados a su uso en investigación, puesto que se trata de sistemas muy especializados. Sin embargo, los simuladores educativos están dirigidos a estudiantes u otro personal en formación, y un análisis cuantitativo de su uso resultaría de gran utilidad tanto para su desarrollo continuado como para comprobar su utilidad. Solamente en dos casos se ha reportado algún tipo de evaluación cualitativa en este tipo de simuladores: la primera pertenece al simulador de Torheim, e indica que los comentarios de los estudiantes después de su uso fueron buenos; la segunda se refiere al simulador Virtual MRI, y en ella se señala que la herramienta tuvo una buena aceptación por parte de los alumnos. Como puede comprobarse, se trata de evaluaciones cualitativas muy genéricas como para permitir extraer conclusiones de interés.

## Capítulo 4

# SISTEMAS DE TUTORÍA INTELIGENTE (ITS)

---

EN este capítulo se comienza haciendo una breve introducción a los ITS, recordando brevemente su historia, la motivación que los originó y algunas de sus aplicaciones. A continuación se describe su arquitectura y se realizan algunas observaciones sobre los tutores cognitivos. Para finalizar, se introducen las *herramientas de creación o Authoring Tools* (AT) y se hace un repaso de algunas de las herramientas más populares para la creación de ITS.

## 4.1 INTRODUCCIÓN

---

A lo largo de la historia se ha empleado de manera exitosa un modelo educativo donde una persona es la responsable de la formación de sus alumnos; esta persona debe orientar y dirigir a los alumnos durante su aprendizaje, bien sea en una materia concreta o a lo largo de un curso académico. Sin embargo, este modelo presenta una clara dependencia del factor humano, que se utiliza de forma intensiva y que limita el número de alumnos a los que puede atenderse. Como consecuencia, se han realizado diversos intentos para automatizar las tareas de tutoría en este modelo educativo. Los ITS se presentan como una respuesta imparcial y de gran disponibilidad, los cuales han evolucionado a lo largo de los años y que hoy en día siguen expandiéndose.

En los tiempos precedentes a la creación del primer ITS, finales de los años 60, la *inteligencia artificial o Artificial-Intelligence* (AI) estaba ya presente en varios sistemas informáticos y también avanzaban sus aplicaciones a la ciencia cognitiva. Por otro lado, con una finalidad comercial, existía la *instrucción asistida por ordenador o Computer-Assisted Instruction* (CAI) que era ya una tecnología madura. La unión de ambas originó el que se considera el primer ITS, llamado SCHOLAR Carbonell (1971). Este sistema fue utilizado en Sudáfrica para evaluar el conocimiento de sus alumnos en geografía. Varios años después, Sleeman and Brown (1982) fueron los primeros en acuñar el término ITS, definiéndolo como todo aquello concerniente con el uso de técnicas de AI para la ejecución adaptativa de tutorías y aprendizaje acorde a lo que se conoce sobre un determinado dominio. Este concepto se popularizó y, como consecuencia, en 1988 se celebró la primera conferencia sobre estos sistemas de tutoría. Su finalidad era la de compartir ideas y consolidar conceptos y teorías referentes a los ITS. Esta conferencia hoy en

día sigue celebrándose y, al igual que estos sistemas, se denomina ITS. Posteriormente, en 1990, comenzó a publicarse la *revista internacional de inteligencia artificial en educación o International Journal of Artificial Intelligence in Education* (IJAIED). En esta revista, se difundió el artículo con el título “Fundamentos teóricos para sistemas de tutoría inteligentes” (Self, 1990). Este manuscrito hizo especial énfasis en la necesidad de establecer unas bases científicas para los ITS, en contraposición a lo que llamaba “teorías informales”. En las últimas décadas y respaldados por la IJAIED, por la Conferencia bianual de *inteligencia artificial en educación o Artificial Intelligence in Education* (AIED) y por la también Conferencia bianual de ITS, se ha establecido una arquitectura bastante aceptada que veremos en la subsección 4.2.

Durante todo este proceso, los campos de aplicación en los que se han utilizado los ITS han sido muy diversos: aplicaciones militares como *BILAT* (Hill Jr et al., 2006), que tenía cuarenta mil líneas de diálogo dedicadas a personajes virtuales, *Helicopter PilotUS* (Mulgund et al., 1998) usado en simulaciones de vuelo con helicópteros o el *Tactical Action Officer* (Stottler and Vinkavich, 2000) utilizado en el entrenamiento de tácticas navales; también en lectura, como el *Project Listen* (Mostow et al., 2001), donde los estudiantes mejoraron la comprensión lectora con un tamaño del efecto de 0.63; en microbiología, donde *Crystal Island* (Sabourin et al., 2012) hacía uso de entornos 3D para el aprendizaje de problemas; en matemáticas, como el *Cognitive Tutor Algebra I (CTAI)* (Pane et al., 2014) que adapta el nivel en problemas de álgebra con tamaños de efecto de 0.7 en diseños experimentales y 70 % de probabilidad de recomendar estos cursos o el *Wayang Outpost* (Arroyo et al., 2014) con más de 300 problemas de matemáticas y que ha demostrado mejorar la confianza y reducir la frustración de los estudiantes al realizar estos problemas. Mencionamos también el campo de la física, donde uno de los ejemplos más emblemáticos es el llamado *Andes* (Schulze et al., 2000); este ITS ha tenido un largo proceso de desarrollo y está basado en varios estudios preliminares. En el primero se concluye que las auto-explicaciones que el alumno genera durante el estudio de conceptos físicos provienen de dos vertientes, la generada durante el estudio teórico y la generada durante el análisis de los ejemplos (Chi and VanLehn, 1991). En el segundo, mediante el uso de un modelo en cascada para el aprendizaje de esta materia, consigue correlaciones entre los ejemplos propuestos a los estudiantes y la cantidad de conocimiento adquirido (Vanlehn and Jones, 1991). Y, finalmente, el tercero (Vanlehn et al., 1994) y cuarto (Ur and Vanlehn, 1995) hacen uso de “estudiantes simulados” y demuestran que éstos pueden ser útiles para el aprendizaje del dominio. Además, estos estudiantes simulados ofrecían la posibilidad de realizar un análisis por ordenador de las tareas cognitivas y de la habilidad que el tutor demostraba aprendiéndolas, lo cual resultó especialmente útil para el diseño del ITS. Así se construyó el primer prototipo de *Andes* (Gertner et al., 1998). En su versión más avanzada (Schulze et al., 2000), *Andes* ha demostrado ser un sistema que proporciona ayuda personalizada para la resolución de 500 ejercicios y ha conseguido resultados con un tamaño de efecto de 0.52 en estudiantes sin antecedentes científicos y del 0.223 en ingenieros, cantidad que subió hasta el 1.21 una vez que se incluyeron diagramas en las explicaciones.

Mousavinasab et al. (2018) realizaron una revisión de 53 artículos en los que se usaban diversos ITS y que fueron publicados entre 2007 y 2017. De estos artículos, el mayor número de ellos, un 37.73 %, se centraban en el aprendizaje de sistemas informáticos, mientras que en segundo lugar empataban los centrados en temas médicos o de salud y en matemáticas, ambos con un 15.09 %. Algunos ejemplos más cercanos a la finalidad de esta tesis, aplicados a campos clínicos y al aprendizaje de programas, son:

- [Jeon et al. \(2007\)](#) implementan un sistema de aprendizaje electrónico para exponer los objetos de aprendizaje a los alumnos de manera interactiva, adaptable y flexible. Se basa en un sistema de simulación de una máquina de anestesia virtual y está orientado al aprendizaje de personal médico. En un artículo posterior, [Jeon and Su \(2010\)](#) evalúan las decisiones que toma el sistema, cómo maneja la información y cómo se adapta a las acciones de los usuarios.
- [Almohammadi et al. \(2016\)](#) realizan un experimento masivo con aproximadamente 1500 alumnos. Utilizan una plataforma online para la enseñanza de Microsoft Excel y PowerPoint. Los contenidos se dividen a lo largo de 20 lecciones, de las cuales cada una está formada por un apartado teórico, una demostración, un ejercicio práctico, un vídeo explicativo y un ejercicio donde los estudiantes reciben comentarios en función de las respuestas. Los resultados observados fueron un incremento en la nota de un 26 % sobre los alumnos que no utilizaron un sistema adaptativo.
- [Kazi et al. \(2012\)](#) abordan el aprendizaje basado en problemas para mejorar las habilidades de razonamiento clínico. En este caso, se utiliza un ITS que aprovecha la ontología para proporcionar de forma efectiva comentarios a los alumnos. Aquí se evaluaron la sugerencias generadas por el sistema y las provistas por una persona experta. Ambas obtuvieron notas muy semejantes, 4.4 y 4.2 respectivamente, en una escala de 5.

En conclusión, la popularidad de los ITS ha crecido significativamente debido a sus resultados. Por ejemplo, [VanLehn \(2011\)](#) encontró que los tamaños del efecto asociados con la tutoría humana —de 0.79— y la tutoría inteligente —de 0.76— eran altamente comparables. Revisiones más recientes ([Ma et al., 2014](#)) han aportado que los ITS brindan mejoras —de moderadas a grandes— en comparación con la instrucción dirigida por maestros, la instrucción no basada en computadora y que no usa un ITS y los libros de texto o cuadernos de ejercicios. Sin embargo, los ITS presentan tamaños de efecto pequeños o medianos si se comparan con una tutoría humana individual ([Kulik and Fletcher, 2016](#)).

## 4.2 ARQUITECTURA DE UN ITS

---

La arquitectura clásica de los ITS está formada por la trinidad ([Rosenberg, 1987](#)): dominio, estudiante y tutoría. Sin embargo, en 1990, Self ([Self, 1990](#)) manifestó que la adopción incondicional de este modelo causaría problemas en la implementación de los ITS y restringiría el alcance en la investigación sobre ellos. Por otro lado, [Nwana \(1990\)](#), durante la revisión de distintas arquitecturas propuestas para los ITS, explica que las diferentes filosofías que adoptan los ITS hacen especial énfasis en uno o varios componentes: dominio, estudiante y tutor. Estas aseveraciones, hicieron que posteriores investigaciones usasen ese modelo pero de una forma laxa. Hoy en día, existe la interfaz como cuarto componente y el conjunto de estos cuatro componentes principales ha sido aceptado por varios autores ([Vanlehn, 2006](#); [Nkambou et al., 2010](#); [Woolf, 2010](#)). La Figura 4.1 muestra estos cuatro componentes, los cuales describimos a continuación. Los nombres por los que se los conoce puede variar dependiendo del estudio:

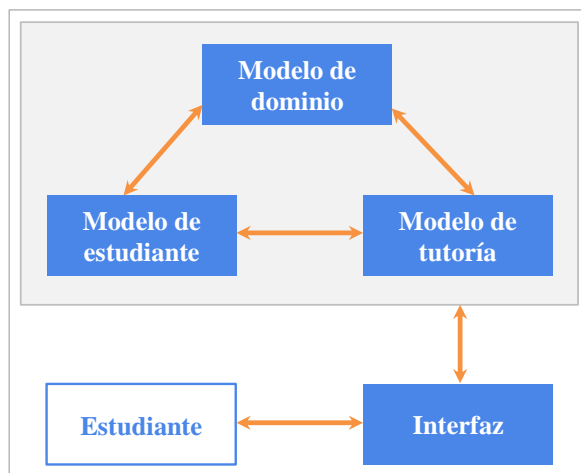


FIGURA 4.1: Arquitectura de cuatro componentes, a saber, el modelo de dominio, el modelo de estudiante, el modelo de tutoría y la interfaz. El componente de la interfaz sería el encargado de establecer la comunicación entre el estudiante y los distintos modelos.

- **Modelo de dominio** (o modelo de conocimiento experto o, simplemente, modelo experto): Contiene los conceptos, reglas y estrategias que deben ser aprendidas en la resolución de los problemas. Puede interpretar uno o varios papeles: ofrecer el conocimiento experto, ocuparse de evaluar al estudiante, responsabilizarse de detectar los errores del estudiante, etc. Puede usar como base un plan de estudios estructurado, donde se incluyan todos los elementos de conocimiento vinculados al dominio. La estructura de organización del plan podría ser jerárquica, usar reglas de producción, tener una base ontológica, etc. Cada plan estaría compuesto por unidades de conocimiento que podrían ser más o menos detalladas. Su función principal sería el seguimiento del rendimiento del alumno.
- **Modelo de estudiante** (o modelo de diagnóstico de estudiante): Idealmente, es el encargado de almacenar estados cognitivos y afectivos del estudiante tales como: el nivel (o su falta) de conocimiento, las actividades realizadas, sus respuestas, el comportamiento mostrado, el estilo de aprendizaje, etc. Se han identificado seis roles fundamentales (Elsom-Cook, 1993): (1) Correctivo: su función es ayudar a corregir problemas en el aprendizaje; (2) Elaborativo: su función es ayudar a enmendar posibles conocimientos parciales; (3) Estratégico: su función es ayudar a iniciar cambios significativos en la estrategia de tutoría; (4) Diagnóstico: su función es ayudar a diagnosticar errores en el conocimiento del estudiante; (5) Predictivo: su función es ayudar a determinar la respuesta probable del estudiante; (6) Evaluativo: su función es ayudar a evaluar al estudiante o al ITS. Estos roles se han ido expandiendo y diversificando pero, en resumen, el modelo del estudiante debe considerar estados cognitivos, afectivos y motivacionales del estudiante y su evolución durante el transcurso del aprendizaje.
- **Modelo de tutoría** (o modelo de instrucción o modelo pedagógico): tomando información del modelo de dominio y del modelo de los estudiantes, su función es detectar la deficiencia de conocimiento y tomar decisiones sobre las estrategias y acciones de la tutoría. Las

sugerencias, comentarios, las recomendaciones, la elección del siguiente tema y el contenido adaptativo también forman parte de las funciones de este modelo. Brevemente, debe ser el ente encargado de saber qué hacer en cada momento y realizar la toma de decisiones.

- **Interfaz o componente de comunicación:** es la encargada de interpretar las interacciones que suceden entre estudiante y el tutor. Las entradas pueden ser diversas, como la voz, la escritura, o una pulsación de ratón, y produce resultados en diferentes medios como texto, diagramas o animaciones. Estos resultados son los encargados de proporcionar acceso a los elementos de conocimiento del dominio.

Después de varias décadas, existe un amplio espectro de arquitecturas y ninguna parece cumplir por completo todas las estrategias que un ITS espera de estos cuatro componentes. Además, más allá de la visión clásica de componentes, los ITS ofrecen una serie de funciones o tareas. La implementación de estas tareas hace que la extensión de uno de estos componentes sea mayor o tome competencias asociadas a otro componente. Por ejemplo, en el componente tradicional, el encargado de llevar a cabo el diagnóstico es el componente de estudiante ([Polson and Richardson, 2013](#)); sin embargo, en otras implementaciones este servicio se considera una función del componente tutor ([Rus and Arthur, 2009](#)).

## 4.3 DISEÑO DE UN ITS. MODELOS COGNITIVOS *vs.* SISTEMAS EXPERTOS

En el diseño de un tutor se deben tomar decisiones sobre cada uno de los diversos componentes. La finalidad es crear una experiencia de aprendizaje donde el tutor tenga un plan definido de acción, es decir, que haga uso de estrategias pedagógicas bien fundamentadas. Para estructurar un ITS existen una gran variedad de paradigmas. Tres de ellos son particularmente populares debido a su forma de representar el conocimiento. El modelo *basado en restricciones* o *Constraint-Based* (CB) y el modelo *basado en reglas* o *Rule-Based* (RB) son dos tipos de modelos cognitivos que se usan frecuentemente en el diseño de ITS. El tercero es más complejo y consiste en la integración de sistema experto en un ITS.

### MODELO BASADO EN RESTRICCIONES

El modelo CB se centra en los requisitos que la solución debe cumplir ([Mitrovic et al., 2007](#)). En otras palabras, el enfoque de este modelo se basa en una perspectiva cognitiva que distingue el conocimiento procedimental y el declarativo ([Ohlsson, 1992](#)). Por tanto, este modelo estriba en la definición de un conjunto de restricciones o soluciones correctas. En el momento en el que el alumno infringe una restricción, el tutor diagnostica un error y le proporciona ayuda ([Graesser and D’Mello, 2012](#)). En el modelo CB, en consecuencia, no es necesario representar el espacio correcto e incorrecto, basta con definir las restricciones que determinan todas las posibles soluciones válidas. Cualquier solución que no cumpla las restricciones se considera incorrecta.

Introduce el concepto de *clase equivalente*, que se refiere a cualquier solución que genera la misma reacción por parte del tutor. Cada clase equivalente puede ser considerada un fragmento



del dominio, generando así un enfoque modular. Este conjunto de módulos hace que la respuesta del alumno sea correcta o incorrecta (Nkambou et al., 2010). Sin embargo, el camino que toma el alumno para que la solución sea adecuada no es determinante para el diagnóstico de errores.

## MODELO BASADO EN REGLAS

El modelo RB, en contraposición del modelo CB, se cimienta sobre las reglas que el estudiante debe seguir paso a paso durante la ejecución de un problema (Koedinger and Alevén, 2007). Esto hace que se utilice en problemas que puedan estructurarse en determinados pasos; en otras palabras, son útiles para problemas bien definidos. Cada uno de los pasos puede recibir un comentario en función de si ha sido correcto o no.

Para la construcción de este modelo suele realizarse un análisis de las tareas cognitivas (Conati and Merten, 2007). Estas tareas se definen mediante la observación de usuarios, tanto expertos como novatos, que utilizan diferentes estrategias de resolución de problemas. De esta forma se crean modelos de tarea que son denominados reglas de producción. Existen diversas formas de representar estas reglas; las más comunes son mediante el uso de árboles con los objetivos o espacios donde los estados quedan unidos por una transición o acción. En ocasiones estas acciones puede ser etiquetadas como errores (Nkambou et al., 2010).

Los sistemas que utilizan un *seguimiento de modelo o Model-Tracing* (MT) son una variante del modelo RB. Estos modelos codifican las operaciones para resolver un problema como un conjunto de reglas de producción (Alevén et al., 2009a). A medida que el alumno va realizando las operaciones, el sistema rastrea el modelo que ha sido previamente creado por el conjunto de reglas de producción. Al quedar el problema totalmente definido, el sistema es capaz de sugerir al alumno el próximo paso, hacer una demostración o evaluar el conocimiento adquirido por éste. Los autores que utilizan los tutores de MT establecen como objetivo el evaluar el proceso de razonamiento que ha seguido el alumno para resolver el problema. Al ser un problema bien definido, puede asumirse que tanto el punto de comienzo del problema como el punto final están definidos.

## SISTEMA EXPERTO

El sistema experto se refiere al modelo que es capaz de emular la capacidad de decisión que posee una persona experta (Kabanza et al., 2005). Estos modelos resuelven los problemas simulando una persona humana, como un tutor *basado en el agente o Agent-Based* (AB) (Rus et al., 2014) o una conversación, como un tutor *basado en el diálogo o Dialogue-Based* (DB) (Graesser et al., 2014). Dicho de otra forma, tratan de imitar las habilidades operativas de las que dispondría una persona experta y su capacidad para modelar y afrontar un problema. También abarcaría la elección de las variables y cómo éstas se conectan con los diferentes elementos que se presentan en una situación real. A diferencia de los modelos cognitivos vistos, donde la solución se muestra más claramente, el sistema experto afronta el problema de una forma más aristotélica, devolviendo respuestas más abstractas y dando pie al raciocinio. Por ejemplo, donde un modelo RB señalaría habitualmente el elemento erróneo, el sistema experto podría limitarse a proporcionar un comentario abstracto sobre el error. Sin embargo, la principal desventaja de este sistema es que su desarrollo es mucho más costoso que el de los modelos cognitivos, especialmente si el dominio no está definido con precisión (Nkambou et al., 2010).



---

## 4.4 HERRAMIENTAS DE CREACIÓN DE UN ITS

---

### 4.4.1 JUSTIFICACIÓN DE SU NECESIDAD

---

El diseño e implementación de un ITS es un proceso muy costoso y que requiere de un conocimiento profundo tanto a nivel pedagógico como de desarrollo. Incluso, en posesión de dicho conocimiento, en la literatura se estiman tiempos de desarrollo para los ITS que varían entre 200 (Woolf, 2010) y 300 horas (Murray, 2003) por cada hora de instrucción. Las *herramientas de creación o Authoring Tools* (AT) surgen como respuesta a la necesidad de reducir complejidad en la creación de un ITS. Sus objetivos, en consecuencia, son:

- Automatizar los procesos de creación de un ITS, tanto en su metodología como en su desarrollo.
- Proporcionar herramientas para que tanto diseñadores, investigadores y profesorado puedan organizar los conocimientos que desea transmitir.
- Fomentar el uso de buenas prácticas de diseño en pedagogía mediante el uso de interfaces o herramientas.
- Habilitar la creación rápida de prototipos de ITS.
- Promover la reutilización de código a través de módulos y estructuras de datos independientes del dominio.

Para la consecución de estos objetivos, las AT cuentan con varios elementos, los principales de los cuales se enumeran a continuación:

- Una *framework* modular, con diferentes herramientas y con relaciones definidas entre ellas.
- Un conjunto de especificaciones que establecen las fases que deben sucederse para la creación de un sistema. Estas fases podrían ir desde la especificación de las características del sistema hasta su despliegue.
- Un conjunto de ejemplos o casos de ITS para la creación o la reutilización de la herramienta, así como para facilitar su uso.
- Una plataforma donde almacenar las pruebas realizadas, su evolución y los sistemas finales para su despliegue.
- Una plataforma donde almacenar todos los datos generados durante una prueba o experimento.

### 4.4.2 DESCRIPCIÓN DE LAS PRINCIPALES HERRAMIENTAS

---

A lo largo de estos años han sido muchas las implementaciones de AT, cada una de ellas con su propia tecnología para la creación de un ITS. Los sistemas que van a ser descritos seguidamente son conjuntos de herramientas pensados para ser usados por profesores. Este

personal generalmente no cuenta con formación técnica y, por ello, uno de los objetivos de estas herramientas es que requieran poco o ningún conocimiento de programación. Por otro lado, la mayoría de ellas hace uso de los paradigmas vistos en la sección 4.3.

1. **Authoring Software Platform for Intelligent Resources in Education (ASPIRE)**. Es un tutor CB que ha sido creado por el Grupo de Tutoría de Computación Inteligente en la Universidad de Canterbury, Nueva Zelanda (Mitrovic et al., 2009). Implementa un ITS que ha sido testado con éxito en pruebas de laboratorio y aulas. Cuenta con dos herramientas, ASPIRE-Author y ASPIRE-Tutor. En primer lugar, ASPIRE-Author es una plataforma para la creación de nuevos tutores y está a disposición de los expertos para su uso. Entre sus propiedades, ASPIRE-Author permite la generación automática del dominio y para ello necesita una descripción del conocimiento de alto nivel, ejemplos de problemas y sus soluciones. En segundo lugar, ASPIRE-Tutor es un servidor que almacena los diversos tutores creados, y los deja disponibles para su uso. El funcionamiento de ASPIRE es estructurado, estableciendo estrictos pasos para la creación de un tutor, a saber: 1) Modelado de la estructura del dominio; 2) Composición de la ontología del dominio; 3) Modelo de las estructuras de problemas y soluciones; 4) Diseñar la interfaz del estudiante; 5) Agregar problemas y soluciones; 6) Generación de restricciones de sintaxis; 7) Generación de restricciones semánticas; 8) Despliegue del dominio.
2. **CTAT** Es uno de los conjuntos de herramientas con más éxito y se utiliza para el desarrollo de tutores cognitivos mediante reglas de producción (Aleven et al., 2016). Además, este sistema fue el primero en implementar un tipo de tutor de MT y está desarrollado por la Universidad Carnegie Mellon (CMU). CTAT permite a los autores vincular los conocimientos de tutoría a una interfaz y mostrar las soluciones al problema de forma rápida (Aleven et al., 2009b). Se compone de dos herramientas: CTAT Behavior Recorder y CTAT *HyperText Markup Language* (HTML) Editor. CTAT Behavior Recorder es la herramienta esencial de este sistema. Su principal característica es la de generar un gráfico que almacena el comportamiento del usuario. Una vez generado, puede establecer diversos tipos de concordancia entre los datos que introduce el usuario y los requisitos que éstos deben cumplir para ser correctos. Así mismo, esta herramienta habilita la introducción de los comentarios e indicaciones que se desean dar al alumno. Por otro lado, CTAT HTML Editor es una herramienta web que permite el diseño rápido de interfaces sencillas utilizando los componentes propios de CTAT. Una vez creada la interfaz, ésta se comunica con CTAT Behavior Recorder para habilitar el seguimiento de las acciones del usuario. Para el despliegue del tutor CTAT propone diferentes formas: 1) un despliegue local; 2) un despliegue en su TutorShop, el cual es un *sistema de gestión de alumnos o Learner Management System* (LMS) propio. En este espacio pueden subirse tanto la interfaz como los ejercicios implementados y permitir así su acceso a los alumnos; 3) Un despliegue en otro LMS como SCORM, LTI o OLI; 4) Un despliegue en un servidor web.
3. **ASSISTments**. Su desarrollo surgió de la cooperación entre la CMU y el Worcester Polytechnic Institute (WPI). La base de este sistema toma como cimientos el trabajo que ya se había realizado en CTAT. Por ello, este programa usa el mismo tipo de tutor específico de MT de CTAT, el de seguimiento de ejemplo. Actualmente es una plataforma web que permite compartir contenido entre profesores. Esta materia tiene una gran

diversidad, abarcando temas como matemáticas, estadística, idiomas o lectura. A su vez, cada una de estas materias divide su contenido en porciones específicas de conocimiento o ASSISTments. Cada ASSISTment puede considerarse un problema que se compone por las preguntas, las sugerencias y los mensajes de error que el alumno va a recibir (Heffernan and Heffernan, 2014). Por otro lado, cuenta con la herramienta ASSISTment Builder (Razzaq et al., 2009) que está pensada para crear, editar, probar e implementar contenido para un tutor de manera efectiva. Este constructor, es una simplificación de los tutores de seguimiento de ejemplo de CTAT en el sentido de que ya no son compatibles con la redacción de reglas de producción y sólo permiten una línea de razonamiento de una sola dirección.

4. **SimStudent**. Usando como base la creación de tutores cognitivos que permite CTAT, SimStudent ayuda a los autores en la creación de un sistema experto utilizando aprendizaje automático (machine-learning) (Matsuda et al., 2015). Otorga a los usuarios dos formas diferentes de crear este sistema. La primera se denomina *creación por autoría*, donde el autor es el encargado de instruir de forma interactiva a SimStudent. La manera de hacerlo es planteando primero un problema a SimStudent; en este instante, SimStudent irá realizando pasos para la resolución del problema y el autor debe proporcionarle comentarios a medida que avanza. Además, SimStudent puede solicitar que le demuestre qué pasos debe dar o que señale cuándo SimStudent no puede dar un paso concreto. La segunda forma de creación es mediante la demostración; en este caso, el autor realiza los pasos y el tutor los registra. En el caso de existir varias soluciones, SimStudent trata de generalizar el ejemplo para adaptarse a todas las soluciones. También cuenta con un entorno de aprendizaje en línea llamado *Artificial Peer Learning environment Using SimStudent* (APLUS) que consta de tres elementos: (1) una interfaz en la que el estudiante podría enseñar a SimStudent de manera similar a como el autor instruye a SimStudent; (2) un tutor cognitivo que proporciona al estudiante unas pautas sobre cómo enseñar a SimStudent y cómo resolver problemas; (3) un agente con un avatar 3D que también tiene la capacidad de aprender a través del estudiante.
5. **Extensible Problem-Solving Tutor (xPST)**. Está diseñada para la creación de tutores MT y fue desarrollada por el centro de aplicaciones de realidad virtual de la Iowa State University en colaboración con la University of Tampa. A diferencia de otros sistemas, la filosofía de esta herramienta se orienta principalmente a la creación de ITS en interfaces o programas ya existentes (Gilbert et al., 2015). La arquitectura de xPST, de la misma forma que sucede con CTAT, hace una clara separación entre la interfaz del alumno y el motor de tutoría. El elemento que media entre la comunicación de ambas partes es llamado *TutorLink module*. De esta forma, la interfaz puede ser cualquier programa existente, siempre y cuando TutorLink sea capaz de traducir las acciones al lenguaje que xPST entiende. Además TutorLink debe ser capaz de comunicar los comentarios del tutor a la interfaz. TutorLink ha sido escrito para los programas Microsoft .NET, el motor de juego Torque 3-D (Kodavali et al., 2010) y el navegador Firefox (Gilbert et al., 2009). La herramienta para la creación del tutor es web y se le debe indicar cuál es la interfaz, normalmente mediante un enlace a la misma. En este momento, el plugin proporciona los nombres de los elementos (la id en el código HTML) que compondrán el tutor. La forma de introducir los comentarios para cada elemento es mediante la introducción de texto en

el archivo xPST. Una vez creado el tutor, es directamente posible guardarlo y ejecutarlo.

6. **ASTUS**. Es una framework para la creación de tutores MT y fue desarrollado por el departamento de Ciencias de la Computación de la Université de Sherbrooke (Paquette et al., 2015). En este departamento consideran que los tutores MT usa conductas pedagógicas integrales, flexibles y reutilizables. Utiliza un sistema de representación de conocimiento propio, donde el conocimiento declarativo se divide en componentes semánticos y episódicos, mientras que el conocimiento procedimental se subdivide en tres categorías. La primera hace referencia a los procedimientos más complejos; éstos se corresponden con un algoritmo, es decir, con la secuencia o posibles secuencias de pasos necesarios para resolver un problema. La segunda categoría abarcaría los conocimientos primitivos, a saber, aquellos conocimientos de los que el usuario ya dispone. En tercer lugar, se encontraría el conjunto de reglas que representan las habilidades dominadas, como la combinación de patrones o la aritmética elemental. Como contrapartida, no cuenta con herramientas propias para la implementación.
7. **AutoTutor Authoring Tools**. Son herramientas de creación de tutores desarrolladas por la University of Memphis. Se utiliza para el desarrollo de tutores que utilizan un lenguaje natural. Las pruebas realizadas con este sistema han demostrado que puede ofrecer ayudar al estudiante mediante el uso de conversaciones (Graesser, 2016). Cuenta con una herramienta de creación de scripts llamada *AutoTutor Script Authoring Tool* (ASAT) (Cai et al., 2015), la cual está disponible en dos variantes: ASAT-X, que es una herramienta basada en el lenguaje XML y ASAT-V que se utiliza para la generación de scripts relacionados con Microsoft Visio. Sin embargo, la generación de reglas de conversación en estas herramientas puede ser compleja para los instructores. En consecuencia, se creó la herramienta AutoTutor Lite que es más intuitiva (Wolfe et al., 2013). Todas estas herramientas incorporan una teoría instruccional, haciéndolas capaces de guiar al autor del ITS al definir los elementos del diálogo, las acciones o las transiciones.
8. **Situated Pedagogical (SitPed)** creada por la Universidad del Sur de California, se centra en gran medida en la autoría basada en vista previa (Lane et al., 2015). Está dirigida a los autores que no tengan experiencia en programación y su objetivo es la creación de tutores para el entrenamiento de habilidades “blandas”, como el asesoramiento. Trata de utilizar la perspectiva del alumno en la mayor medida posible. Con este fin, proporciona una serie de herramientas para ejecutar su flujo de funcionamiento, el cual dividimos en cuatro pasos.

El primer paso es la definición de tareas, donde SitPed cuenta con una herramienta destinada a la creación de modelos de tareas simples que definen el comportamiento correcto o incorrecto de un escenario. El segundo paso es conectar dichas tareas a los datos o escenarios para habilitar su evaluación. Por ejemplo, en una conversación ramificada, los estudiantes seleccionan las respuestas de un conjunto de elementos. Aquí el autor debe explorar todas las posibles ramas para vincular cada posible elección con las habilidades y conceptos erróneos del dominio. El tercer paso consiste en la creación de consejos y pistas que el alumno recibirá por parte del tutor. Finalmente, un cuarto paso trata de añadir contenido para reflexionar sobre la práctica. Este proceso debe ser probado de forma iterativa hasta lograr la consecución de un tutor.

### 9. *Generalized Intelligent Framework for Tutoring (GIFT) Authoring Tools (GAT).*

Las herramientas de creación de GIFT son un conjunto de programas abiertos para la creación de tutores y su evaluación (Sottolare et al., 2012). Ha sido creada por el *laboratorio de investigación del ejército de EE.UU. o U.S. Army Research Laboratory (ARL)* para el desarrollo del *modelo de aprendizaje del ejército de EE.UU. o U.S. Army Learning Model (ALM)*. Integra herramientas como ASAT para la creación de diálogos para el apoyo. Cuenta además como muchas otras herramientas que no han sido mencionadas en esta sección, como *Domain Knowledge File Authoring Tool (DAT)*, *CourseAuthoring Tool (CAT)*, *Pedagogy Configuration Authoring Tool (PCAT)*, *Survey Authoring System (SAS)*. De esta forma GIFT soporta gran variedad de tutores, tales como los basados en diálogo, los tutores basados en GIFT, y los *modelos de información de estudiantes para entornos de aprendizaje inteligentes o Student Information Models for Intelligent Learning Environments (SIMILAR)*. Por esta razón GIFT ha servido para crear tutores en varios dominios como, por ejemplo, la resolución de acertijos, atención a las víctimas o criptografía.

## 4.5 DISCUSIÓN Y CONCLUSIONES

Como puede verse, el número y tipologías de herramientas existentes para la creación de un ITS no es menor. La mayoría de ellas funcionan bien en sistemas donde el problema está bien definido, es decir, donde el punto de inicio del problema y su estado final están bien determinados o, en caso de existir varios estados iniciales o finales, éstos están bien delimitados. En este contexto, parece claro el éxito de los tutores cognitivos, en especial los CB o los RB; el ejemplo más notable es el CTAT, el cual ha sido utilizado en diversos sistemas como ASSISTments o SimStudent. Por otro lado, los tutores basados en conversación también han demostrado su eficacia en la resolución de los problemas bien definidos y ASAT es un modelo a seguir.

Sin embargo, en el caso en que el problema esté mal definido, la programación de los ITS se vuelve muy costosa. Para los tutores CB o RB sería necesario crear un modelo de conocimiento y de tutoría que fuese capaz de abarcar todas las posibles soluciones del problema, incluyendo las óptimas y las subóptimas pero correctas –tal es el caso de CTAT, ASSISTments, ASPIRE, xPST, ASTUS–. Por otro lado, en los tutores conversacionales, el árbol de decisiones crecería notablemente, ya que la conversación tendría que abarcar puntos de vista muy dispares –como sucede en ASAT–. En respuesta a estos obstáculos, se crearon sistemas que pudiesen aprender, es decir, sistemas que hacen uso de técnicas de aprendizaje automático, donde son los estudiantes o el autor los que enseñan al sistema mediante la ejecución del problema desde distintas perspectivas –como ocurre en SitPed o SimStudent–. En otras palabras, son los propios sistemas los encargados de dar pautas a los estudiantes y autores para que sean éstos los que instruyan al propio sistema. De esta forma, cuantas más veces sea repetido un ejercicio, más probable es que abarque todos los posibles estados.

Un aspecto relevante de las AT es cómo afrontan la construcción de nuevos sistemas, dada su gran disparidad; tales sistemas pueden ser muy estructurados y que el entorno para hacerlo sea único –esto sucede con ASPIRE y ASSISTments, donde todo el proceso se lleva a cabo en una web– o, por contra, ser más descentralizados y permitir, en primer lugar, la creación de una interfaz, y posteriormente, habilitar la incorporación del sistema de tutoría a ésta GUI

—como por ejemplo CTAT, ASTUS, ASAT y GIFT—. Abundando en ello, quizás el sistema más descentralizado es xPST, donde su objetivo es facilitar la incorporación de un ITS a cualquier interfaz. Sin embargo, este sistema ha tenido muy poco soporte y hoy en día no es funcional.

La forma en la que se construyen influye también en el comportamiento del tutor. Por ejemplo, el acoplamiento existente entre la interfaz y el sistema de tutoría marca en gran medida su funcionamiento. En el caso de xPST y CTAT este acoplamiento es bajo, ya que ambos utilizan un sistema intermedio de comunicación entre la interfaz y el sistema. Esto permite crear interfaces complejas sin tener demasiado en cuenta el sistema, pero también limita algunas facetas en su comportamiento. Un caso claro puede verse al comparar CTAT con ASTUS. En ambos se programó un problema matemático donde se realizaba una resta. Al realizar los comentarios en un error, ambos marcaban la zona errónea, sin embargo, CTAT indicaba con precisión el elemento erróneo. En su lugar, ASTUS hacía uso del conocimiento semántico del problema, permitiendo diversos grados, podía no marcar nada en primer lugar, posteriormente marcar la columna donde se encontraba el error o finalmente señalar el error concreto. Emular el comportamiento de ASTUS no es sencillo con un bajo acople entre la interfaz y el sistema. Por contra, el desarrollo del problema en ASTUS fue más costoso.

La tecnología empleada, esto es, el sistema que se utiliza para su implementación así como los lenguajes con los que son compatibles, es otro limitante. Estos sistemas están en constante evolución, y la actualización por alguna de las partes puede repercutir en incompatibilidades. Sin embargo, la tendencia parece clara, ya que la mayoría de los sistemas dan soporte a interfaces que se ejecutan en entornos web. En el momento de comienzo de la tesis, algunas usaban tecnologías como Adobe Flash, como es el caso de ASPIRE. No obstante, Adobe Flash ha experimentado diversos problemas de seguridad como recogió (Jobs, 2010), donde además expuso una serie de razones por las que no se incorporaría Flash en los productos con iOS. Flash también fue posteriormente desactivado en los navegadores Mozilla Firefox y Google Chrome. La tendencia, hoy en día, parece ser el lenguaje JavaScript, que es utilizado por CTAT, ASSISTments, SimStudent o ASAT.

Para finalizar, la enumeración de AT expuesta en la Sección 4.4.2 incluye las herramientas más populares y mejor documentadas para la creación de un ITS durante el desarrollo de la tesis. Existen, no obstante, un conjunto de herramientas que no han sido contempladas en esta lista, tales como: *Cognitive Model Software Development Kit* (SDK) (Blessing et al., 2009), *Tutorial Development Kit* (TDK) (Blessing et al., 2009), PHP ITS (Weragama and Reye, 2013), SQL-Tutor (Mitrovic et al., 2002), SlideTutor (Crowley and Medvedeva, 2003), *AutoTutor Script Authoring Tool for Assessment* (ASATA) (Graesser, 2016) o SAS, la cual ha sido nombrada en GIFT. Todas ellas fueron descartadas inicialmente por diversas razones, entre ellas, signos de obsolescencia, poca documentación, un estadio muy inicial de desarrollo, etc.

Los sistemas de ITS recientemente descritos tampoco han sido contemplados en esta lista. Actualmente se está apostando por la integración de sistemas ya establecidos y la estandarización de componentes; ejemplos de ello son ElectronixTutor (Graesser et al., 2018) o SPOKE-IT (Nye et al., 2018). El segundo se basa en objetos de conocimiento que se puedan compartir entre distintos tutores, haciendo uso a su vez de AutoTutor. La reutilización de tecnologías se realiza a mayor escala en ElectronixTutor, el cual se ha desarrollado combinando recursos de aprendizaje inteligentes de Dragoon, AutoTutor, LearnForm, ASSISTments y BEETLE-II. Como decimos, no obstante, en la tesis hemos tomado como punto de partida las herramientas descritas en la sección 4.4.2.



## Capítulo 5

# DESARROLLO DEL SIMULADOR

---

ESTE capítulo describe los pasos dados para el desarrollo de un simulador de resonancia magnética. En la primera sección se indicarán las características que el sistema debe cumplir. En la siguiente sección se establece un diseño del sistema y del programa. Posteriormente se detalla cómo se implementa el programa y cómo se afrontan cada una de las partes que lo componen. Luego, describiremos los resultados mostrando características de la herramienta educativa creada. Finalmente haremos unas consideraciones sobre el grado de cumplimiento de los objetivos y algunas limitaciones del sistema.

## 5.1 INTRODUCCIÓN

---

El objetivo principal de la tesis (Sección 1.2) establece que debe llevarse a cabo *el diseño, desarrollo y evaluación de un simulador de resonancia magnética como herramienta educativa*. Esta sección se centra en el diseño y desarrollo de la aplicación, explicando las razones que motivan las decisiones tomadas en su diseño y el procedimiento que se ha empleado para su implementación. Para simplificar en la medida de lo posible la explicación, el conjunto de las versiones del simulador (Sección 7.2), será tratado como una única versión final. En consecuencia, en esta sección, describiremos el análisis (Sección 5.2), el diseño (Sección 5.3) y la implementación (Sección 5.4), teniendo en mente la última versión. Finalmente se expondrán los resultados (Sección 5.5) y a continuación se harán algunas consideraciones sobre ellos (Sección 5.6).

## 5.2 ANÁLISIS

---

En el análisis recopilamos todos los requisitos que debe cumplir nuestro programa. El objetivo es poder concretar cada una de las características que le permita a nuestro simulador ser una herramienta educativa de relevancia en la formación de radiólogos y principalmente de técnicos radiólogos. Para ello y a lo largo del tiempo, se han tenido en cuenta tres fuentes de orientación. La primera consiste en la información recabada del estudio del estado del arte. La segunda es el compendio de ideas obtenido en las distintas reuniones mantenidas con expertos

radiólogos y profesores de la SERAM, donde se han discutido aspectos relevantes del simulador. Finalmente, la tercera considera los comentarios proporcionados por los estudiantes en las diversas pruebas realizadas (Sección 7.2). En consecuencia, se han identificado un conjunto de *requisitos funcionales* (FR) y *requisitos no funcionales* (NFR), los cuales pasamos a continuación a enumerar:

1. El sistema debe ser capaz de simular adquisiciones a partir de unos parámetros dados. De forma abreviada, debe ser capaz de:
  - a) Simular la imagen de contraste de diferentes secuencias de pulsos, tales como: SE, FSE, *Dual Spin Echo* (DSE), GRE, etc.
  - b) Simular efectos que puedan surgir en la imagen como artefactos debido al ruido, aliasing, movimiento, etc.
  - c) Generar un imagen en formato médico que contenga la adquisición simulada.
2. La interfaz debe permitir la variación de estos parámetros de simulación, incluyendo los que se indican a continuación:
  - a) Selección de los parámetros de adquisición básicos, como: TE, TR, *tiempo de inversión* (TI), FA, ETL y BW. En ocasiones, éstos deberán variar su nomenclatura en función de la secuencia, como es el caso del *tiempo de eco efectivo* ( $TE_{eff}$ ).
  - b) Debe además contar con un panel que contenga las opciones de imagen. Éstas son necesarias para solventar artefactos o aplicar determinadas técnicas. Debe contar con opciones como habilitar NPW, activar las bandas de saturación, etc.
  - c) Debe permitir la selección de otros parámetros que afectan a la imagen, como *Partial Fourier* (PF), NEX, *Scan Percentage* (SP), etc.
  - d) Debe habilitar paneles que permitan la variación de los parámetros de ciertas técnicas como *diffusion weighted imaging* (DWI) o la adquisición en paralelo.
3. La planificación geométrica debe permitirse para una zona dada. Esta planificación debe contar con parámetros como el FOV en fase y en frecuencia, el tamaño de la matriz, el grosor del corte, la separación entre ellos, etc. Además, debe ser funcional para poder adquirir nuevos cortes sobre los ya adquiridos anteriormente.
4. El técnico radiólogo suele guiarse por algunos indicadores como la SNR relativa, el tiempo del escáner o el tamaño del voxel. Es importante que el simulador cuente con estos indicadores.
5. En otro orden de ideas, como se especifica en la Sección 1.2, el simulador debe permitir la realización del flujo de trabajo completo que un técnico radiólogo realiza. Para ello, es necesario que el simulador cuente con:
  - a) Un panel donde introducir la información del paciente.
  - b) Un panel para la selección de la posición del paciente.
  - c) Un panel para la selección de la bobina.





4. Sólo los usuarios profesores o administradores podrán crear protocolos y secuencias visibles para el resto.
5. El sistema debe proporcionar indicaciones sobre el estado para poder identificar fallos; en particular, debe contar con indicadores como que el registro ha fallado o cuál es el estado de la simulación.
6. El sistema evitará, en la medida de lo posible, las especificidades asociadas a los fabricantes así como el empleo de nombres comerciales para las secuencias.

## 5.3 DISEÑO

---

En esta sección describimos en primer lugar el diseño de alto nivel del sistema, donde se detalla su arquitectura, el propósito que ésta cumple y los lenguajes, las bibliotecas y/o las herramientas que van a ser utilizados. Posteriormente, se detallan decisiones sobre el diseño del programa a más bajo nivel y, por tanto, más cercano al de la implementación.

### 5.3.1 DISEÑO DEL SISTEMA

---

En el estado del arte visto en el Capítulo 3 se observaron una gran variedad de simuladores de MR. De su estudio, se advirtió que la forma que éstos tenían de afrontar el proceso de la simulación de MR se podía esquematizar en cinco grandes bloques, como se representó en la Figura 3.13. Estos bloques se denominaron: opciones de simulación, modelo, ecuaciones de Bloch, reconstrucción e imagen. De ellos, el modelo estaba formado por una gran cantidad de datos, mientras que los bloques de ecuaciones de Bloch y de reconstrucción implicaban una gran carga computacional. Teniendo en cuenta además el NFR 1, se optó por una arquitectura cliente-servidor web, donde la labor del programa se reparte entre el proveedor de servicios, es decir, el servidor web, y el encargado de demandar dichos servicios, a saber, el cliente, en este caso, una aplicación web. Por lo tanto, las tareas como la visualización de las imágenes o la selección de las opciones de simulación se plantean en el lado del cliente, mientras que las tareas que implican cierta carga computacional o que requieren cierta capacidad de almacenamiento —como el modelo—, permanecen en la parte del servidor. De esta forma, la aplicación será accesible desde cualquier lugar y/o momento, requiriendo únicamente el uso de un navegador web.

Específicamente, el diseño del servidor se plantea con una *arquitectura orientada a servicios* o *Service-Oriented Architecture* (SOA) que se rige por los principios descritos en su manifiesto (Arsanjani et al., 2009). Los servicios pensados para esta arquitectura se esquematizan en la Figura 5.1 y son los siguientes:

- **Servicio archivos estáticos.** Éste, como su nombre indica, está destinado a servir archivos que, una vez generados, no varían. Ejemplos de ello son los archivos que conforman la interfaz gráfica, los correspondientes a las imágenes resultantes de una simulación o los archivos que muestren patologías para su estudio.

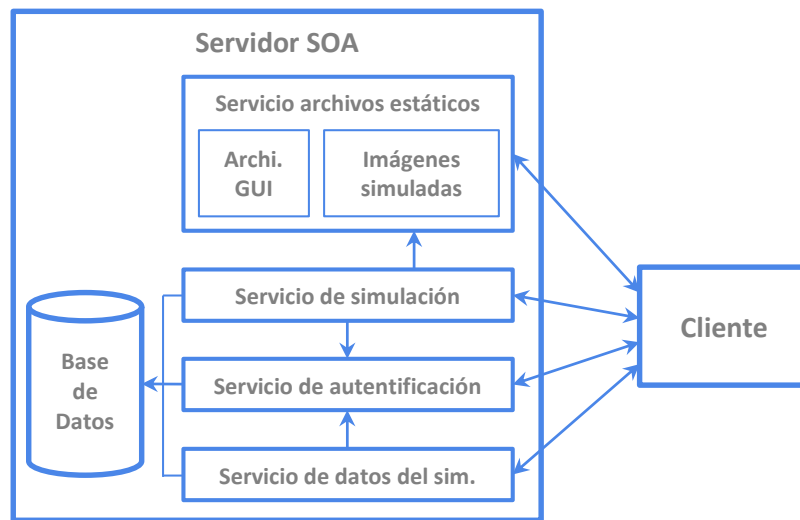


FIGURA 5.1: Servidor con arquitectura SOA. Esta figura muestra cómo el cliente accede a los cuatro servicios principales que ofrece nuestra herramienta: servicio de archivos estáticos, servicio de simulación, servicio de autenticación y servicio de datos del simulador. Además se muestran las relaciones existentes entre sí y entre éstos y la base de datos.

- **Servicio de autenticación.** En esta tesis también nos referimos a él como registro, y es el servicio encargado de permitir o denegar el acceso a la herramienta, además de otorgar a los usuarios diferentes roles, cumpliendo así el NFR 3. Tiene acceso a la base de datos, donde los usuarios registrados son almacenados.
- **Servicio de datos del simulador.** Sirve para proporcionar datos necesarios en la GUI, como los protocolos, partes anatómicas, bobinas, etc., disponibles en la base de datos. Como la Figura 5.1 representa, para acceder a esta información el cliente debe estar registrado.
- **Servicio de simulación.** Es el servicio a cargo de realizar la simulación y, por tanto, el principal responsable de satisfacer el FR 1. Debe tener acceso a la base de datos para conocer los modelos y características disponibles, así como para poder comprobar si el usuario está registrado. Una vez finalizada la simulación, debe situar el archivo en un lugar accesible para el cliente.

Para su desarrollo se ha seleccionado Django (Burch, 2010) que es una de las *framework* web de alto nivel más populares escritas en Python. El uso de este lenguaje da gran flexibilidad a la *framework* y permite un desarrollo más rápido y ordenado. Además, Django sigue la filosofía *no te repitas* o *Don't Repeat Yourself* (DRY) que fomenta la reutilización de código. Por otro lado incorpora por defecto un sistema de administración que permite la gestión de la base de datos desde la web y para su activación sólo es necesario cierto nivel de configuración. También incorpora la herramienta *mapeo objeto-relacional* o *Object-Relational Mapping* (ORM) que permite la realización de consultas de tipo *Structured Query Language* (SQL) pero sin utilizar expresamente SQL. Finalmente, permite un diseño modular, donde cada aplicación implementa

una arquitectura *modelo vista controlador* o *Model-View-Controller* (MVC). Además, cada una de estas aplicaciones podría proporcionar uno o varios servicios. En nuestro caso, todo lo relacionado con el servicio de autenticación, es decir, la creación de la cuenta, el login, el logout o la actualización de los datos, será una única aplicación. Por otro lado, el servicio de simulación, así como el encargado de almacenar y suministrar los datos del simulador, serán proporcionados por una segunda aplicación.

Dentro del servicio de simulación, se encontraría el núcleo encargado de realizar dicha simulación, el cual debe cumplir el FR 1 y el NFR 2. Para ello, se opta por un lenguaje compilado, C++, que, además, cuenta con las bibliotecas para el manejo, creación y visualización de imágenes médicas *Insight segmentation and registration ToolKit* (ITK) y *Visualization ToolKit* (VTK). La conexión entre Python y el núcleo de simulación se hará a través de un *Wrapper* que se detallará en el apartado de implementación (Sección 5.4.2).

En el lado del cliente, se alojará la interfaz de la consola de MRI que es responsable total o parcialmente de los FR 2, 3, 4, 5, 6 y 7, y de los NFR 3 y 6. Además, debe incorporar otra serie de herramientas para cumplimentar los FR 8 y 9. Por ello, el diseño debe ser modular de forma que la interfaz sea fácilmente escalable y permita la incorporación de nuevas herramientas y funcionalidades.

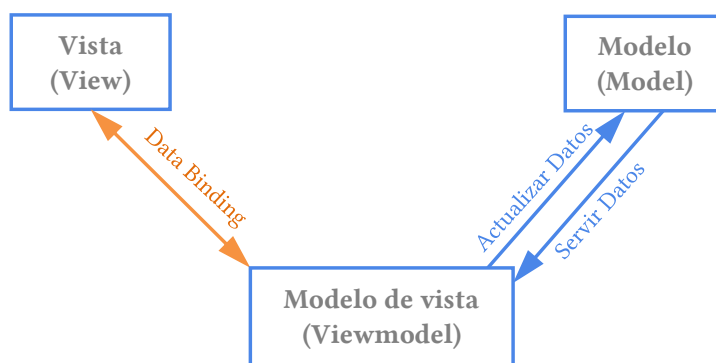


FIGURA 5.2: Arquitectura *Modelo-Vista-Modelo de vista*. Esta figura muestra la comunicación existente entre los componentes de la arquitectura MVVM. Como puede verse, la comunicación entre la *Vista* y el *Modelo* se realiza exclusivamente a través de la *Modelo de vista*.

AngularJS es una *framework* para la generación de aplicaciones web que posee una orientación modular. Además, hace uso de una arquitectura MVC, sin embargo, también es posible hacer uso de ella de forma que tenga una mayor orientación a componentes, siguiendo la arquitectura *modelo vista modelo de vista* o *Model-View-ViewModel* (MVVM), la cual fue creada por John Gossman en 2005 y popularizada posteriormente (Anderson, 2012). Esta arquitectura se representa en la Figura 5.2 y está formada por 3 bloques, donde el *modelo* (Model) se refiere a los datos y a la capa que da acceso a éstos, la *vista* (View) es la estructura, el diseño y la apariencia que el usuario ve en la pantalla, y define cómo van a ser representados los datos del modelo, y el *modelo de vista* (Viewmodel) contiene el manejo de los eventos que el usuario va a generar a través de la *vista* y sirve como canal entre la *vista* y el *modelo*. Para que esto sea realizado de forma eficiente, es necesaria la existencia de un enlace entre *vista* y el *modelo de vista* que sincronice ambos; este enlace es referido como *Data-Binding*. Habida cuenta de su

importancia en el simulador, hemos incluido el Apéndice B, con detalles específicos sobre esta *framework*. Además, en el cliente, también será necesaria la visualización de imágenes médicas; con este fin, se utilizará la biblioteca *X Toolkit* (XTK) que es de código abierto y proporciona un interesante soporte para este tipo de imágenes.

Por otra parte, en el diseño gráfico de la aplicación se va a utilizar *Syntactically Awesome Style Sheets* (SASS) que tiene una sintaxis más amplia del lenguaje *Cascading Style Sheets* (CSS) aunque no es directamente interpretable por los navegadores, por lo que requiere de una compilación. Además, con el propósito de que la aplicación web sea más rápida tanto en la carga como en la ejecución, es necesaria la optimización del tamaño de las imágenes así como la minimización del código *JavaScript* (JS), HTML y CSS. Por otra parte, al tratarse de una arquitectura cliente-servidor, es necesario que los archivos sean copiados al servidor antes de ser probados.

Este conjunto de tareas, parciales o totales, para la adecuación del código y de los archivos, deben ser realizados cada vez que quiera comprobarse el correcto funcionamiento de la aplicación durante su desarrollo. Para optimizar este proceso se crearon los *task runner*, que permiten su programación y su ejecución, bien de forma iterativa (cuando se detecta un cambio en el código) o bien bajo demanda con la ejecución de un comando. En nuestro caso particular, el task runner elegido será *Gulp*, que permite la programación de las tareas mencionadas mediante JS.

### 5.3.2 DISEÑO DEL PROGRAMA

Con la elección de una aplicación cliente-servidor, el desarrollo queda dividido de forma natural en dos grandes bloques; por un lado la *Interfaz* del simulador que se encuentra en el cliente y, por otro lado, el *Núcleo de Simulación* alojado en el servidor. De los FR y los NFR surgen bloques más pequeños como el sistema de registro, el diseño de los ejemplos, o el visor de casos.

#### INTERFAZ

A lo largo de esta tesis, cuando empleemos el término “la interfaz”, nos referiremos a su parte principal que es la consola de MR, mientras que el resto de partes, que podrían considerarse dentro de la GUI, las tomaremos como desarrollos independientes. Por ello, se ha diseñado una barra lateral (Figura 5.3-a), a través de la cual se podrá acceder a estos otros desarrollos.

La interfaz ha sido diseñada para imitar la interfaz de una consola de MRI real, así como para brindar soporte a todos los pasos necesarios para ejecutar la simulación. Como se muestra en la Figura 5.3, está formada por los *Paneles* etiquetados desde la b hasta la l, los cuales se han diseñado para cumplir varios de los FR vistos. En primer lugar, la interfaz se diseña para que el usuario deba introducir la información del paciente en el *Panel c* (FR 5-a). A continuación, deben introducirse en el *Panel d* la posición de entrada del paciente y la bobina que está siendo utilizada (FR 5-b y 5-c). Luego debe darse la posibilidad de seleccionar el protocolo; esto se realiza mediante un panel desplegable que será accedido desde el botón situado a la derecha de “Protocol” (Figura 5.4a) en el *Panel e* (FR 5-d).

El protocolo debe contener información sobre la parte anatómica para la que está diseñado. Además debe indicar si es público o privado para poder ser visto por uno o varios usuarios.



FIGURA 5.3: Panel principal del simulador. El panel principal a su vez se divide en diferentes partes enumeradas de la *a* a la *l* y que servirán de guía en el texto.

Finalmente, debe incluir todas las secuencias que lo conforman. Por otro lado, las secuencias, además de ser públicas o privadas, deben contener todos los parámetros necesarios para fijar el tipo de secuencia de pulsos y sus tiempos, la información de la planificación espacial inicial, las opciones de la imagen así como otros parámetros iniciales.

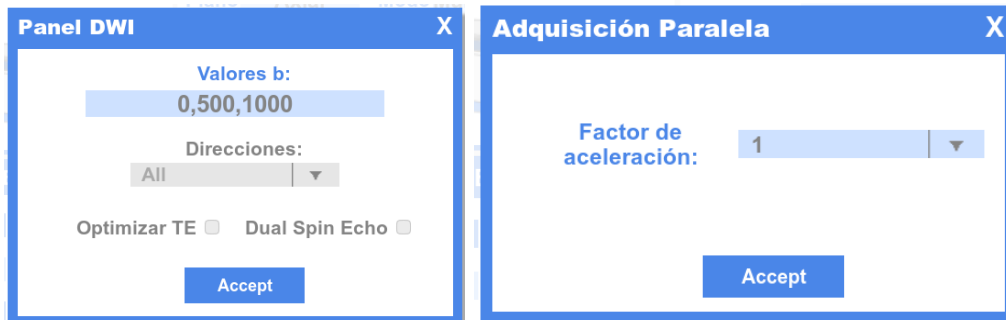
Una vez seleccionado el protocolo, esta información poblará la interfaz gráfica, aunque en primera instancia, sólo se verá reflejado en el *Panel g*, el cual debe permitir la selección de las secuencias que conformen dicho protocolo. Al seleccionar una secuencia, su información debe ser accesible para el usuario, tanto para su visualización como para su modificación. De esta manera, se deben rellenar automáticamente los *Paneles e, h, j, k y l* y los parámetros deben ser modificables, bien a través de esos paneles o de otros; ello permite el cumplimiento de los FR 2-a, 2-c, 4. Los FR 2-b y 2-d serán cubiertos mediante el uso de *Paneles desplegables*, donde el panel de opciones de imagen (Figura 5.4b) es accedido desde el *Panel e*, mientras que las opciones de DWI (Figura 5.4c) y de adquisición en paralelo (Figura 5.4d) serán accesibles mediante el *Panel i*. Finalmente, la planificación geométrica, de la que hablaremos más adelante, también será ejecutada en un panel desplegable.

Una vez rellenados/modificados los datos, debe ser posible iniciar la simulación desde la interfaz. Con este fin, se emplea el botón “Scan”, situado en la parte inferior derecha del *Panel l*. Al ser pulsado, los datos de la interfaz se envían al servidor, el cual, mediante el uso del *núcleo de simulación* explicado más adelante, obtendrá la imagen resultante, momento en el cual devolverá a la interfaz su localización para su descarga. La imagen, una vez descargada, se visualizará automáticamente en el *Panel f*. El visualizador donde se muestra la imagen está diseñado para atender a una serie de comandos básicos, de los cuales algunos son habituales en



(A) Selector de protocolos.

(B) Opciones de imagen.



(C) DWI.

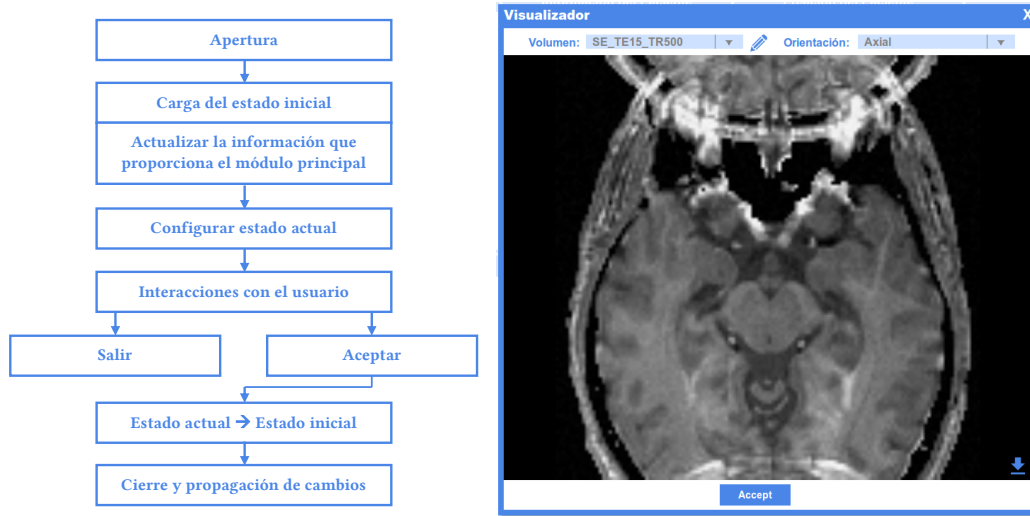
(D) Aceleración paralela.

FIGURA 5.4: Paneles desplegados I.

la inspección de imágenes. Éstos son: el uso de la ruleta para el desplazamiento entre cortes, zoom manteniendo el botón derecho pulsado y desplazando el ratón lateralmente, pulsado del botón izquierdo y desplazamiento vertical u horizontal para la variación del brillo y el contraste y, finalmente, doble clic para reiniciar las propiedades la imagen. Existe también una opción de visualización más avanzada mediante un panel desplegable (Figura 5.5b).

Los *paneles desplegados* están diseñados para ofrecer posibilidades que no aparecen en la consola principal, permitiendo que un panel emerja con nuevas opciones específicas, como se observa en la Figura 5.4. Además, las opciones elegidas en estos paneles, en general, tendrán repercusión en el panel principal, por lo que la información debe ser propagada por las partes afectadas. El flujo de funcionamiento que los paneles desplegados deben seguir se corresponde con el esquema de la Figura 5.5a; En concreto, al abrir cada panel se carga la información inicial, que bien puede estar dada por la información guardada del estado anterior o por la información que proviene del panel principal. A continuación, se configura el estado actual del panel, es decir, la información que el usuario verá y manipulará, haciendo una copia del estado inicial. Ahora, para el usuario debe ser posible la interacción con el panel desplegable. Si el usuario pulsara “salir”, se presupone que no quiere que se guarden/apliquen los cambios, por lo que





(A) Esquema de un panel desplegable.

(B) Panel Visualizador.

FIGURA 5.5: Paneles desplegables II.

se desecharía el estado actual. Sin embargo, si el usuario pulsara “aceptar”, el estado actual se copiaría al estado inicial (que el panel vería en su próxima apertura); además, se cerraría el panel y, en caso de ser necesario, se propagarían los cambios por el panel principal. Existirían excepciones, como puede ser el panel de visualización, donde los cambios que se realicen no tienen ninguna influencia sobre el panel principal, o el visualizador, donde se incluye la función de descargar la imagen simulada (FR 5-e)

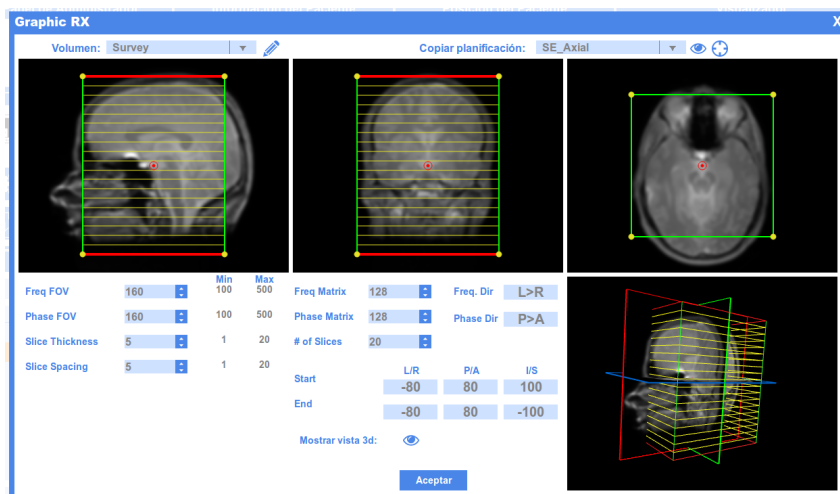


FIGURA 5.6: Panel para la planificación geométrica. Puede verse la planificación de los cortes sobre las imágenes sagital, coronal y axial del *survey*. Además, en la parte inferior derecha de la imagen, aparece una visión 3D de las imágenes junto a la planificación, que sirve como referencia para entender mejor la colocación de los cortes.



La planificación geométrica es quizás el panel desplegable con el diseño más complejo (Figura 5.6), ya que la inicialización debe tener en cuenta todas las planificaciones e imágenes disponibles en ese momento en la interfaz. Debe permitir que cualquiera de las imágenes adquiridas pueda ser usada para realizar la planificación, además de permitir la manipulación de todos los parámetros que influyan en la localización geométrica de los cortes (FR 5). Además, la posición de los cortes con respecto a la zona anatómica puede en ocasiones ser confusa; por ello, facilitar su visualización mediante una imagen tridimensional de ambas puede resultar interesante para los estudiantes. Esta opción debe estar desactivada por defecto y sólo ser habilitada a petición del usuario. La posición de los cortes, a la que nos referiremos como *región de interés* o *Region Of Interest* (ROI) de aquí en adelante, debe permitir la interacción con el ratón. Para su desplazamiento debe pulsarse en la parte central y arrastrar a la zona indicada. Para su rotación se utilizarán los vértices de la ROI y finalmente para su extensión, se utilizarán las líneas rojas que delimitan la ROI. Este comportamiento imita el manejo de una máquina real. Dada su complejidad, los detalles de este panel se ven con mayor profundidad en el Apéndice C.

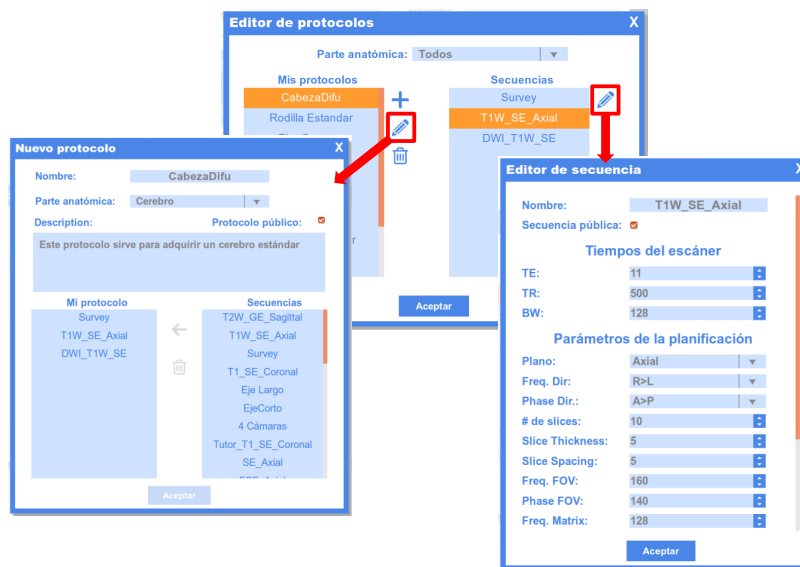
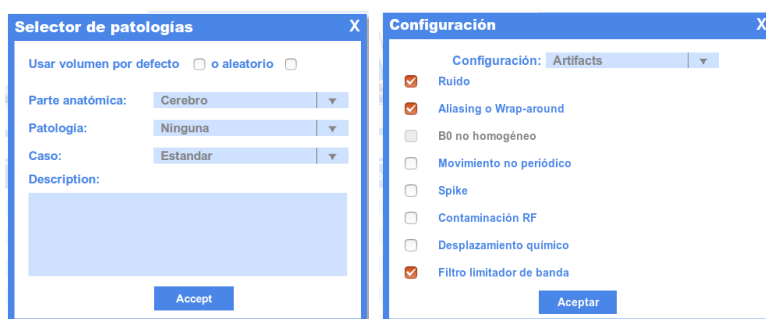


FIGURA 5.7: Herramienta para la creación de protocolos y secuencias. Se compone por 3 paneles, el principal (parte central), donde se pueden visualizar los distintos protocolos creados, y los secundarios, destinados a la creación y edición de protocolos (parte izquierda) y a la edición de secuencias (parte derecha).

Otra utilidad educativa, se encuentra en la creación y edición de los protocolos y secuencias propias del usuario (FR 7). Para ello se ha ideado un sistema de paneles accesibles desde el *panel b* de la Figura 5.3 y llamado “Mis protocolos”. La Figura 5.7 muestra su diseño, que sigue un patrón similar a los paneles desplegable, si bien hace uso de tres: el “Editor de protocolos” es el principal y permite la visualización de los protocolos propios y el acceso a los otros dos; el llamado “Nuevo protocolo” sirve para la creación y la edición de protocolos y el tercero, el “Editor de secuencias” sirve para la modificación de cada una de las secuencias que conforman un protocolo. Con ese diseño, el usuario inicialmente vería los protocolos propios y directamente

tendría la opción de añadir, modificar o borrar uno de ellos. Además, cuando el protocolo estuviese seleccionado, las secuencias que lo conforman se verían en el área de Secuencias del “Editor de protocolos” y, de igual forma, permitir el acceso a su modificación.

Una vez iniciada la creación o modificación de un protocolo, el usuario debe poder elegir una serie de secuencias; éstas pueden ser propias, es decir, que el usuario las crease previamente, o genéricas, es decir, que estén disponibles por defecto para que el usuario pueda usarlas en sus protocolos. Si es una secuencia genérica, ésta debe ser duplicada en la base de datos actualizando el propietario de la secuencia. Al modificar los parámetros de una secuencia, las opciones disponibles variarán en función de dicha secuencia (por ejemplo, en una secuencia SE, los tiempos que aparecerían sería el TE y el TR, mientras que en una secuencia GRE, podría aparecer también el FA).



(A) Panel de modelos.

(B) Panel de configuración.

FIGURA 5.8: Paneles despleables III

Dado el carácter del programa, debe ser posible acceder a características con las que no cuenta la consola de MR. Un ejemplo sería la selección del modelo anatómico sobre el que se va a llevar a cabo la simulación (FR 6-a), donde su diseño puede verse en la Figura 5.8a. Otro ejemplo se da en la configuración de parámetros de la máquina o la activación de artefactos de la imagen, cuyo diseño puede verse en la Figura 5.8b y atiende al cumplimiento de los FR 6-b y 6-c.

Finalmente, hay momentos en los que el usuario inicia una acción, como puede ser pulsar “Scan” y, si no ocurre nada, tiende a pensar que algo no ha funcionado. Por este motivo, el simulador debe mostrar al usuario su estado, bien mediante mensajes de espera, de error o de finalización (NFR 5).

Como consideración adicional, en el diseño web y de aplicaciones móviles se opta por el uso de diseños adaptativos (responsive) donde, en función del tamaño de la pantalla y del dispositivo desde el cual se acceda, se reestructuran la posición, el tamaño y el orden de los elementos que conforman la interfaz. En nuestro caso, dado que imitamos a una consola real, este diseño no es adecuado. Tampoco lo es un diseño en el que los elementos tengan un tamaño relativo dado que se deformaría al modificar el tamaño de la ventana o la pantalla. La opción más conveniente es la utilización de tamaños fijos, usando como medida el *pixel* (px). Este tamaño debe ser fijado tanto para la letra como para los elementos del código HTML, permitiendo de esta manera un reescalado “exacto”, el cual va a realizarse siempre que el zoom sea aplicado.

## NÚCLEO DE SIMULACIÓN

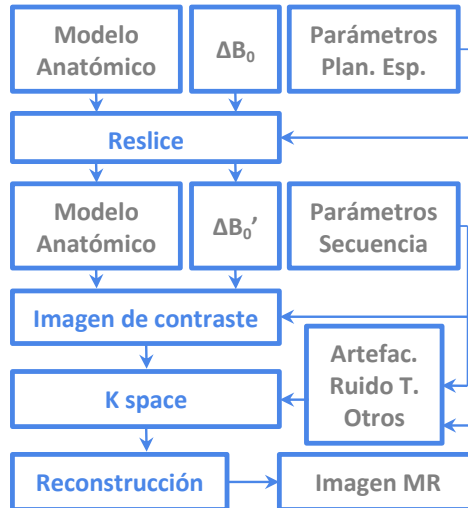


FIGURA 5.9: Esquema del núcleo de simulación. Los bloques con las letras en color gris representan los datos, tanto introducidos como generados. Los bloques con las letras en color azul representan las operaciones realizadas.

La Figura 5.9 muestra los pasos que el núcleo de simulación debe dar hasta obtener la imagen simulada. En primero lugar, las entradas serían, por un lado, el *Modelo Anatómico* y el campo magnético de inhomogeneidad ( $\Delta B_0$ ); por otro, los datos geométricos contenidos en los *Parámetros de la Planificación Espacial*.

La selección del *Modelo Anatómico* puede ser automática o manual por parte del usuario, y cada una de ellos está compuesta por un conjunto de volúmenes 3D que contienen las propiedades del tejido necesarias para la simulación de MR, a saber, mapas  $T_1$ ,  $T_2$  y *densidad de protones o Proton Density* (PD), los cuales sólo podrán ser accedidos por el propio *núcleo de simulación* (NFR 3). Éstos han sido obtenidos en un equipo real de MR utilizando secuencias de relaxometría (Ramos-Llordén et al., 2017). El *Modelo Anatómico*, además, puede estar formado por otras características del tejido, como el *coeficiente de difusión aparente o Apparent Diffusion Coefficient* (ADC). En cuanto a  $\Delta B_0$  es un campo sintético que persigue el objetivo de resaltar la necesidad de un procedimiento de *shimming*, es decir, un procedimiento que, antes de la adquisición real, realice una homogeneización del campo magnético.

Los *parámetros de planificación* contienen información geométrica que especifica el número de cortes, su orientación y otras propiedades adicionales. Con las entradas detalladas, pasa a realizarse la operación de *Reslice*, que crea volúmenes del tamaño y con las propiedades especificadas por el usuario por medio de los parámetros de planificación geométrica. Creando en este punto estos subvolúmenes, es posible minimizar el uso de memoria RAM y la carga computacional en cada petición de simulación.

A continuación, estos subvolúmenes pasan a ser una entrada al núcleo de simulación, conjun-

tamente con los *Parámetros de la Secuencia*; ambas entradas son utilizadas para la simulación de la imagen de contraste mediante la evaluación de la expresión algebraica de la secuencia específica (Zhi-Pei Liang, 2000; Bernstein et al., 2004).

Posteriormente, se aplica una transformada de Fourier a la imagen de contraste, produciendo el llamado espacio k. En este espacio, se pueden simular fácilmente varios artefactos, como efectos de movimiento, picos procedentes de alteraciones de radiofrecuencia o ruido térmico. Estos efectos serán simulados de la forma vista en la Sección 2.2.3. El espacio k permite la incorporación de efectos adicionales, como la atenuación de los trenes de ecos de las secuencias rápidas o la reconstrucción con relleno parcial de este espacio.

Finalmente, se realiza la reconstrucción de la imagen. Ésta, dependiendo de cómo haya resultado la manipulación del espacio k, puede consistir directamente en la aplicación de la transformada de Fourier inversa o, si fuese necesario, de un relleno previo del espacio k en adquisiciones parciales. En este bloque también se incluye la estimación de mapas adicionales de imagen, como es el caso del cálculo del ADC. El resultado es la imagen de contraste, que contendrá información espacial adicional (orientación, origen y tamaño de vóxel) para su correcta visualización, así como otro tipo de información relacionada con la secuencia o la información del paciente (NFR 2.c).

En este esquema, dada la forma y el lenguaje en el que ha sido diseñado, el *Núcleo de simulación* tiene acceso en todo momento a los parámetros de la simulación (NFR 2.b), y los tiempos de simulación son compatibles con la práctica docente (NFR 2.a).

## EJEMPLOS

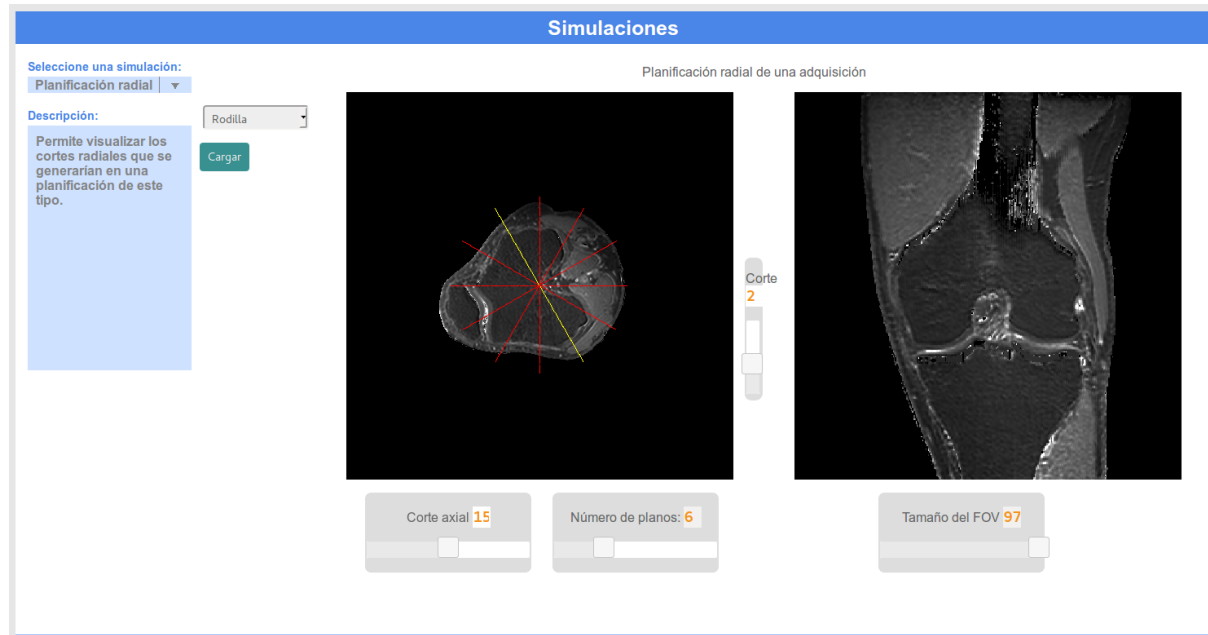


FIGURA 5.10: Ejemplo de cómo sería una planificación radial.

Para hacer una análisis más detenido de un fenómeno, en ocasiones es necesario aislarlo del resto, y el apartado de *Ejemplos* atiende a este propósito. Este apartado está diseñado para contener todas aquellas simulaciones o programas que tengan como fin específico mostrar en detalle un proceso o un efecto relacionado con la MR (FR 8). Su acceso se debe permitir desde la barra lateral vista en la Figura 5.3, y su boceto se plantea con tres componentes principales. En primer lugar, un menú desplegable que permita elegir el ejemplo a ejecutar. En segundo lugar, un panel que albergue una descripción del ejemplo, para que el alumno pueda entender en qué consiste el ejemplo en cuestión. En tercer lugar, el componente más importante es el área donde se visualizará e interactuará con el ejemplo; ésta, además, tendrá siempre un tamaño fijo. Con este diseño se pretende que el desarrollo de estos ejemplos sea totalmente independiente de la plataforma. El diseño final puede verse en la Figura 5.10

## VISOR DE CASOS

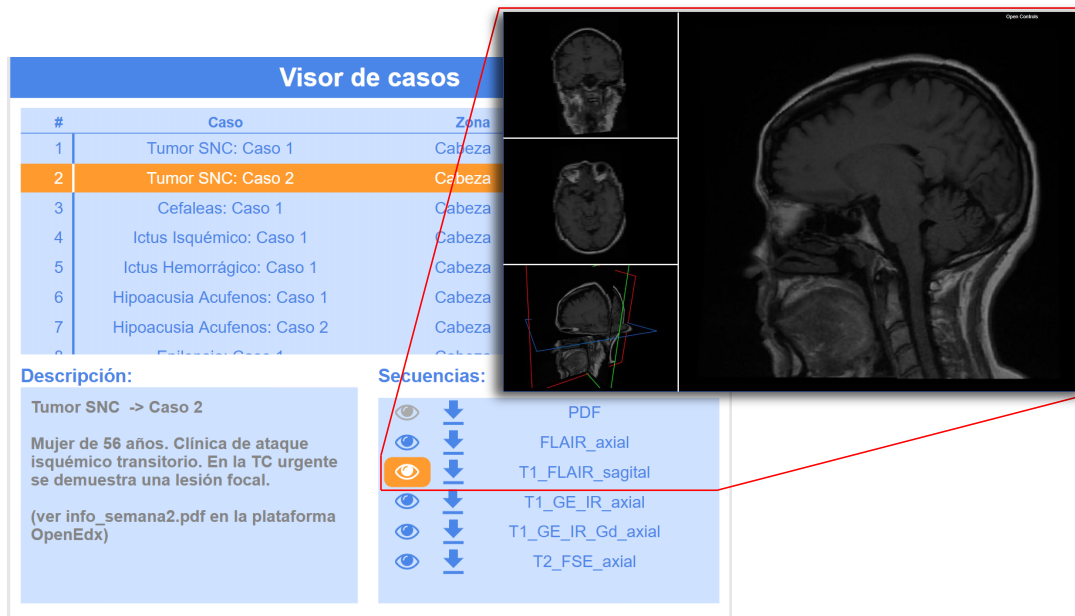


FIGURA 5.11: Visor de casos. Se compone por dos herramientas, el panel donde se seleccionan los distintos casos (a la izquierda) y la herramienta para su visualización (a la derecha).

El estudio de casos reales permite la adquisición de experiencia en la visualización y el entendimiento de imágenes de MR por parte del estudiante, fundamentalmente para casos con patología. En este tipo de casos es muy difícil disponer de mapas de relaxometría, requisito de entrada en nuestro sistema de simulación (recuérdese la figura 5.9). Ello se debe a que desde una perspectiva ética es difícil de justificar que una persona con patología permanezca en la máquina más tiempo del estrictamente imprescindible según el protocolo pautado.

La incorporación de un útil que preste estos servicios otorga un valor añadido a nuestra herramienta educativa (FR 9). El diseño se centra en dos partes: el menú que da acceso al caso y el visor que permite la visualización del mismo. El menú será la carta de presentación que vea

el usuario al entrar en esta utilidad y su diseño debe permitir la selección del caso entre todos los disponibles, así como su acceso a la información. Con este fin, el panel se estructura en tres partes, el menú seleccionable, un apartado para la descripción del caso y un tercer recuadro que da acceso a toda la información del caso. Cada caso está compuesto por un documento que describe los detalles del mismo y que puede ser descargado, así como las imágenes adquiridas para su diagnóstico, las cuales puede ser o descargadas o examinadas mediante el uso del visualizador. Para evitar un comportamiento molesto para el usuario, cada imagen que desee visualizarse debe ser abierta en el visor en una pestaña nueva.

El visor debe estar compuesto por las tres vistas bidimensionales (cortes axial, sagital y coronal) y, además, debe contar con una vista tridimensional que permita ver la situación del corte en el espacio. El diseño final puede verse en la Figura 5.11. Además, debe ser posible maximizar cualquiera de las vistas. Para ello debería valer con pulsar en la vista minimizada. En la Figura 5.11, donde se muestra el diseño, las vistas minimizadas se encontrarían a la izquierda, siendo éstas coronal, axial y 3d, mientras que la vista sagital estaría maximizada a la derecha.

## OTROS

El servicio de simulación (Figura 5.1) utiliza el *protocolo simple de acceso a objetos* o *Simple Object Access Protocol* (SOAP) habida cuenta de su necesidad de intercambiar solicitudes y respuestas con el cliente. Sin embargo, el resto de servicios se diseñan con el sistema de *transferencia de estado representacional* o *Representational State Transfer* (REST), permitiendo la creación de una *interfaz de programación de aplicaciones* o *Application Programming Interface* (API) para la solicitud, creación, modificación y borrado de la información de la base de datos. Esto se aplica tanto al sistema de registro como a la información relevante para la interfaz, es decir, a los modelos anatómicos disponibles, las bobinas, los protocolos y las secuencias.

Mediante el sistema de registro, cada usuario podrá contar con el rol de profesor y alumno (FR 10). Además, se selecciona la posibilidad de etiquetar al usuario como activo (o no), permitiendo restringir el acceso a los usuarios sin la necesidad de eliminar su cuenta. Adicionalmente, se contempla la posibilidad de usuario administrador.

## 5.4 IMPLEMENTACIÓN

Esta sección se centra en los detalles de implementación que resultan de interés para el entendimiento del funcionamiento del simulador, tanto en su núcleo como en su parte gráfica. En primer lugar, a través de los paneles desplegable, veremos cómo se implementa la arquitectura MVVM, de forma que sea fácil extrapolar este método al resto de los componentes de la interfaz (Sección 5.4.1). Las siguientes secciones hacen referencia al núcleo de simulación. En primer lugar se muestra cómo éste se comunica con el servidor (Sección 5.4.2), posteriormente se describe cómo son generadas las imágenes resultantes de las secuencias de pulsos (Sección 5.4.3). Finalmente se explica cómo se generan los volúmenes para su descarga y visualización en el cliente (Sección 5.4.4).

### 5.4.1 PANELES DESPLEGABLES

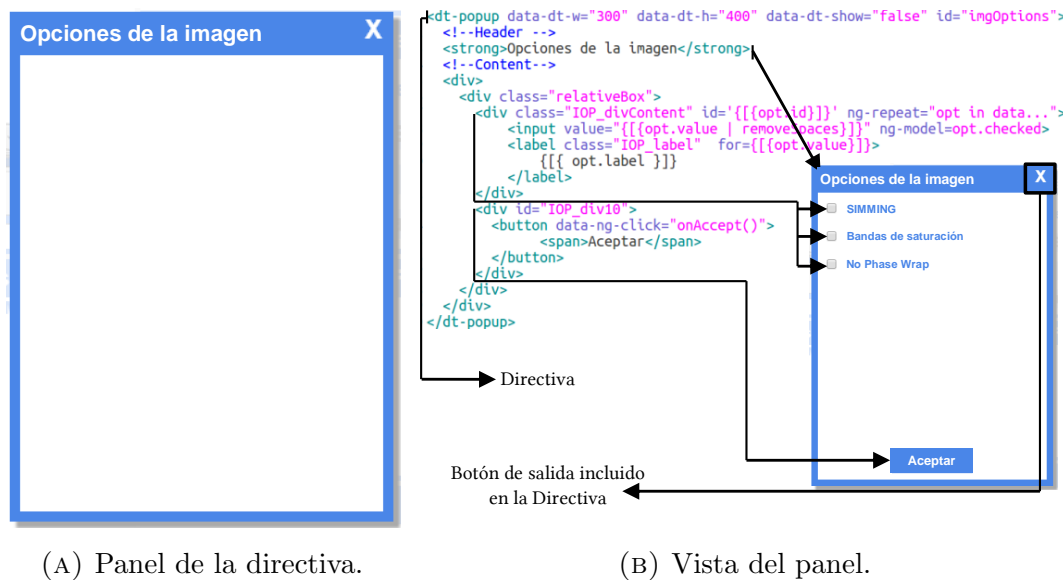


FIGURA 5.12: Opciones de la imagen. La figura (A) muestra cómo sería el panel de opciones de la imagen sólo introduciendo el título en la directiva. En la figura (B) se muestra el código de la *Vista* y cómo repercute en la apariencia del panel.

La interfaz de una consola real de MRI es actualizada cada cierto periodo de tiempo, y por tanto, va variando su aspecto gráfico así como la forma de actuar en ciertos procedimientos. En consecuencia, nuestra herramienta debe ser capaz de evolucionar también, de ahí el diseño modular con el que cuenta, donde cada componente puede ser desarrollado de forma independiente al resto de la interfaz y, aun así, seguir manteniendo un diseño gráfico común. Quizá uno de los casos más evidentes son los paneles desplegados que van a ser explicados a continuación.

En primer lugar, todos los paneles desplegados hacen uso de la misma directiva. Una directiva es una de las herramientas que incorpora AngularJS y puede ser considerada como un fragmento de código pensado para ser reutilizado a lo largo del desarrollo de la aplicación (ver Apéndice B.5). En este caso particular, esta directiva se denomina `dtPopup` y se declara como un elemento que cuenta con varios atributos para definir su ancho, su alto, su nivel de superposición (`z-index`) o si debe estar desplegado u oculto por defecto. Un ejemplo puede verse en el siguiente fragmento de código, cuyo efecto se muestra en la Figura 5.12a.

```

1 <dt-popup data-dt-h="400" data-dt-w="300" data-dt-show="
  true" data-dt-z-index="1" id="imgOptions">
2   <!-- Cabecera -->
3   <strong>Opciones de la imagen</strong>
4   <!-- Contenido -->
5   <div>
6     <div class="relativeBox">
7     </div>

```

```

8 |     </div>
9 | </dt-popup>

```

En este código puede verse la definición inicial de un nuevo panel que tendrá como título “Opciones de la imagen” y su tamaño total será de 400px de altura y 300px de anchura. Además, el `dt-z-index` permite que sea visible por encima del resto de elementos de la interfaz. El parámetro que permite iniciarlo en su forma minimizada (o maximizada) por defecto sería el `dt-show`, que en este caso se corresponde con “true”, por lo que sería visible desde el comienzo.

Una versión abreviada del código utilizado en nuestra herramienta se muestra en la Figura 5.12b, donde vemos que el `dt-z-index` no existe, ya que la propia directiva fija un parámetro por defecto, y este sólo sería necesario cambiarlo si deseamos que el panel sea visible por encima de otro panel desplegable. Además, vemos cómo `dt-show`, en realidad, está fijado como “false”, ya que sólo debe ser visible cuando el usuario lo indique. El mecanismo que hace posible la visualización del panel se implementa nuevamente a través de una directiva, pero esta vez, la directiva es del tipo “atributo”. El código que se muestra a continuación se correspondería con el botón que contiene 3 puntos (suspensivos), que está situado a la derecha de la etiqueta “Opciones imagen”, en el *Panel e* de la Figura 5.3.

```

1 | <button class="ip_button" id="ip_optButton"
   |     data-dt-popupid="imgOptions">
2 |     <span>...</span>
3 | </button>

```

La función principal de esta directiva es detectar cuándo se pulsa el ratón sobre el elemento que la posea como atributo. En el momento que se clicca en el elemento, se inicia una función que busca entre todas las directivas del tipo “dtPopup”, de forma que se localice la directiva cuya “id” se corresponda con la variable que previamente ha sido pasada como parámetro al atributo —“imgOptions” para este caso—. A continuación, indicaría a la directiva “dtPopup” que se cambia el estado de `dt-show`, inicializando así el algoritmo visto en la Figura 5.5a.

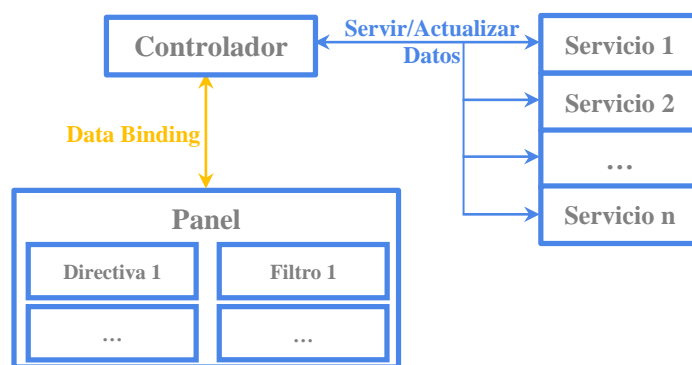


FIGURA 5.13: MVVM en AngularJS. Muestra la arquitectura MVVM aplicada a AngularJS, donde la *Vista* sería el Panel, el *Modelo* serían los distintos servicios y la *Modelo de vista* sería el controlador.

Llegados a este punto, el foco se centra en cómo se implementa la arquitectura MVVM en un panel desplegable. Antes de comenzar, es conveniente señalar que AngularJS cuenta con



*controladores* y *servicios* además de las ya comentadas directivas (consultar Apéndice B). La Figura 5.13 indica cómo sería esta implementación en AngularJS, la cual usa la misma estructura que vimos en la Figura 5.2. Comparando ambas, la *Vista* se correspondería con la Figura 5.12b, la cual describe el aspecto gráfico del panel, quien, a su vez, puede contener directivas y filtros propios o nativos de AngularJS (consultar Apéndice B). Observando el código de la figura 5.12b más detenidamente, puede verse cómo los elementos son generados mediante la directiva “ng-repeat”, la cual recibe como parámetro un vector que contiene distintos objetos. Cada uno de estos objetos contiene las variables “value”, “label” y “checked” que son utilizadas para generar los elementos del código HTML de forma iterativa. Este vector, a su vez, estaría contenido en la variable “data” que es proporcionada por el *Modelo de Vista*, como veremos más adelante.

Por otro lado, el encargado de controlar el flujo de información entre la *Vista* y el *Modelo*, así como los eventos ocurridos en la interfaz, sería el controlador y se correspondería con el *Modelo de Vista*. Finalmente, el *Modelo* estaría formado por uno o varios servicios, los cuales contendrían la información necesaria para el *Modelo de Vista*.

Siguiendo con el ejemplo de las opciones de la imagen, el código del controlador puede observarse a continuación, al cual se le inyectan los servicios `imgOptionsData` y `RMManagerData`. De esta forma, el *Modelo de Vista* tiene acceso al *Modelo* que estaría formado, en este caso, por los dos servicios nombrados anteriormente, donde el primero contiene los datos del panel de las opciones de imagen y el segundo contiene información sobre el panel principal. Con estos datos, y gracias a la variable `$scope`, el enlace entre la *Vista* y el *Modelo de Vista* quedaría realizado. Un ejemplo puede verse en la función `onAccept()` la cual es llamada desde el código visto en la Figura 5.12b y que, como vemos, se define en el controlador. Volviendo al flujo del panel (Figura 5.5a), la directiva ejecutaría, siempre que el panel sea abierto, la función `onOpen()`, la cual realizaría la carga del estado inicial (línea 18), y en caso de ser necesario, cargaría la información actual del panel principal (línea 20) y, finalmente, configuraría el estado actual (línea 22). Una vez realizadas las interacciones por parte del usuario, en este caso, se plantearían dos opciones, o salir pulsando el botón “X” y no guardar los cambios (líneas 13–15), o guardar los cambios (línea 9) y salir propagando los cambios (línea 11) al pulsar el botón “Aceptar”.

```

1  .controller('imgOptionsCtrl', ['$scope', 'imgOptionsData', '
    RMManagerData', function($scope, imgOptionsData,
    RMManagerData){
2      //Datos
3      $scope.data = imgOptionsData.getCurrentData();
4      $scope.initData = imgOptionsData.getInitData();
5
6      //Eventos
7      $scope.onAccept = function(){
8          //Copia los datos actuales (currentData) a que va a ser
          cargada al inici (initData));
9          imgOptionsData.setInitData(imgOptionsData.
            getCurrentData());
10         //Llama a la función encargada de propagar la informaci
            ón;
11         $scope.$parent.data.imgOptions.applyImgOptions();

```

```

12   };
13   $scope.onExit = function(){
14     //Hacer algo al pulsar X
15   };
16   $scope.onOpen = function(){
17     //Inicializa los datos actuales (currentData)
18     imgOptionsData.setCurrentData(imgOptionsData.
19       getInitData());
20     //Carga información del estado actual del panel
21     principal
22     var _opt = RManagerData.getImgOpt();
23     //configura el estado actual
24     imgOptionsData.setValues(_opt);
25   };
26 }]);

```

Finalmente, se observará el código de uno de los servicios inyectados, llamado `imgOptionsData`.

```

1   .service('imgOptionsData', [function(){
2     //Datos
3     var currentData = {
4       options:{
5         checkBox:[
6           {id:'1', value: 'SHI', label:'SHIMMING', checked:
7             false},
8           {id:'2', value: 'SAT', label:'Bandas de saturación',
9             checked:false},
10          {id:'3', value: 'NPW', label:'No Phase Wrap',
11            checked:false},
12        ],
13      }
14    };
15    var initData = angular.copy(currentData);
16    //Servicio
17    var service = {};
18    service.getCurrentData = function(){
19      return currentData;
20    };
21    ...
22    service.setValueCheckedOptions = function(_opt){
23      ...
24    }
25    return service;
26 }]);

```

En primer lugar (línea 3), define la información que va a contener, llamada `currentData` y a continuación (línea 12) hace una copia de la misma, y la denominada `initData`. Este código sólo

va a ser ejecutado la primera vez que el servicio sea construido, quedando su estado inicial con dos variables, `currentData` e `initData` exactamente iguales. Estas variables, más adelante, son las que serán modificadas por el usuario a través de la *Vista*. A continuación se define el objeto que va a ser instanciado, llamado “service” (línea 14) y que contendrá los métodos necesarios para el acceso (líneas 15–17) y el correcto manejo de la información (líneas 19–21) por parte de el *Modelo de Vista*. Recordando la Figura 5.12b, un objeto como el `currentData` sería el que el *Modelo de Vista* proporcionaría, mediante la variable `$scope`, a la directiva `ng-repeat` de la *Vista*.

Todas las características vistas en este punto son aplicadas de manera similar a otros componentes de la interfaz, pudiendo ser éstos directivas o no, como pueden ser las entradas de datos numéricos (directiva), los paneles seleccionables (directiva) o los paneles que conforman la consola. Además, esta forma de aplicar la arquitectura MVVM es también extrapolable a los paneles externos a la interfaz principal, como sería el visor de casos o los ejemplos.

## 5.4.2 WRAPPER

El núcleo de simulación se ha escrito en C/C++, mientras que el servidor lo está en Python, por lo que es necesario establecer una vía de comunicación entre ambos. Una solución inmediata sería ejecutar el programa compilado mediante una llamada al sistema operativo desde Python. En este caso, la comunicación podría venir dada por un archivo estructurado, donde los parámetros fueran escritos previamente con Python y posteriormente leídos por el programa en C/C++. Para que fuese inequívoco, el nombre del archivo debería ser un parámetro variable y único para cada caso, que debería ser pasado al núcleo de simulación como argumento. En consecuencia, para cada simulación por parte de cualquiera de los usuarios, sería necesaria la creación de dicho archivo, provocando un número creciente de ficheros, los cuales, posteriormente, habría que eliminar. Además, habría que asegurar la sincronización entre la creación y cierre del fichero escrito y el inicio de la simulación, provocando posibles retardos. Finalmente, también limitaría las latencias de acceso al disco en comparación con comunicación entre el servidor y el núcleo.

Como alternativa, se utiliza una solución más elaborada, que consiste en la creación de un *wrapper*, el cual no es otra cosa que una interfaz que permitiría la comunicación entre el servidor y el núcleo de simulación. Para ello se utiliza la biblioteca de Python llamada “ctypes” que permite el acceso tanto a bibliotecas de *objetos compartidos* o *Shared Objects* (so) (línea 3 del fragmento de código que sigue) (en GNU/Linux) como a las *bibliotecas de enlaces dinámicos* o *Dynamic Link Library* (dll) (en Windows), posibilitando su manipulación como si se tratasen de objetos en Python (línea 5). A continuación puede verse un ejemplo de cómo cargaría la biblioteca en Python.

```
1 | from ctypes import cdll
2 | #sim nos permite acceder a cualquiera de las funciones que
   |     se encuentren en la librería simuladorLib.so
3 | sim = cdll.LoadLibrary("simuladorLib.so")
4 | #simularC sería la función a la que llamamos núcleo de
   |     simulación
5 | sim.simularC(...)
```

Dado que “ctypes” permite realizar una comunicación directa con el programa en C, deben ser utilizados también los tipos de C. Al tratarse ambos de lenguajes tipados, deben especificarse el tipo de cada argumento que vaya a ser pasado a la función de C. En el caso de nuestro simulador, dada la gran cantidad de parámetros con los que va a contar, las variables estarán ordenadas en estructuras, donde por ejemplo, una estructura podría contener las rutas del modelo anatómico o los parámetros de la planificación geométrica.

Para ilustrar esto, se utilizará una estructura que contendrá un array 1D. La estructura debe estar definida tanto en el código Python como en el código C/C++. En el código en Python que sigue, heredando de la clase `Structure` que “ctypes” proporciona (línea 2), comenzaremos definiendo los campos que posee la estructura, a saber, `data` (línea 3) que será un puntero a los datos del vector que estarán almacenados en `float`, y `len` (línea 5), que contendrá la longitud del vector y será un `int`. Además, se escribe la función `_init_` (línea 7), la cual se ejecuta en el momento en que la clase sea instanciada y asigna los valores correspondientes a los respectivos campos de la clase `vector` (línea 8–11).

```

1 | from ctypes import c_int, c_float, Structure, POINTER
2 | class vector(Structure):
3 |     _fields_ = [("data", POINTER(c_float)),
4 |                #Datos del vector
5 |                ("len", c_int)]
6 |                #Longitud del vector
7 |     def __init__(self, v=np.array([], np.float32)):
8 |         npArr = np.array(v)
9 |         myV = npArr.astype(np.float32)
10 |         self.data = np.ctypeslib.as_ctypes(myV)
11 |         self.len = len(myV)

```

En C/C++ se declara la estructura de forma normal, pero utilizando la misma nomenclatura para los campos y el mismo tipo.

```

1 | //Estructura
2 | typedef struct _vector{
3 |     float *data;
4 |     int len;
5 | } _vector;

```

Antes de continuar, convendría ver cómo se declararía en C/C++ la función que lleve a cabo el núcleo de simulación. La función utilizaría un argumento del tipo `_vector` (línea 2 del fragmento de código que sigue), la cual sería tratada de la forma habitual. Nótese, que como se ha indicado anteriormente, “ctypes” se comunica directamente con las funciones en C, por tanto, es necesario indicar al compilador qué funciones van a ser accedidas desde Python. Para tal fin se utiliza la expresión `extern ‘‘C’’` (línea 6 de siguiente código); en la línea 7 del siguiente cuadro se declara una función denominada `simularC` que va a ser accedida desde el exterior; una vez invocada, esa función llamará a la función de C/C++ `simular`.

```

1 | //Funciones en C/C++
2 | void simular(struct _vector *miVector){
3 |     //Hacer algo

```

```

4  }
5  //Funciones visibles desde el exterior
6  extern "C" {
7      void simularC(struct _vector *miVector);
8          return simular(miVector);
9  }
10 }
```

En consecuencia, desde el wrapper de python podría llamarse a la función de C, a la cual se le pasaría como argumento el objeto `miVector` (línea 6) visto anteriormente y que “ctypes” se encargaría de convertir en una estructura como la declarada en el código C/C++. Finalmente, procederíamos de la forma:

```

1  #función que utilizaremos en python
2  def simPython(vectorNumpy):
3      #vector con los tipos de C asignados
4      miVector = vector(vectorPython)
5      #función de la librería
6      sim.simularC(miVector)
```

De esta forma, declararíamos el resto de estructuras para abarcar todos los datos de simulación, siendo posible la construcción de estructuras anidadas. Un ejemplo podría ser el parámetro que contiene las posiciones de los cortes de la región anatómica; este parámetro sería una matriz y se encontraría dentro de la estructura que contiene los datos de la planificación geométrica.

### 5.4.3 SECUENCIAS DE PULSOS

En este apartado se detallan las fórmulas utilizadas para el cálculo de las imágenes de contraste de cada secuencia. Donde:

- $A$  es la amplitud de la señal resultante.
- $M_z^0$  vector de magnetización en el instante inicial, en nuestro caso se correspondería con PD.
- $TE$  tiempo de echo (TE).
- $TR$  tiempo de repetición (TR).
- $\alpha$  ángulo de giro o flip angle (FA).
- $TI$  tiempo de inversión (TI).
- $T_1$  tiempo de recuperación de la componente longitudinal ( $T_1$ ).
- $T_2$  tiempo de relajación de la componente trasversal ( $T_2$ ).
- $T_2^*$  tiempo de relajación de la componente trasversal afectado por la inhomogeneidad de campo. Se calcularía con la ecuación 2.13.

- $\Delta B$  inhomogeneidad de campo.
- $\gamma$  constante giromagnética del hidrógeno.

## SE

Ecuación 7.26 del libro de [Zhi-Pei Liang \(2000\)](#):

$$A = M_z^0 \left(1 - e^{-\frac{TR}{T_1}}\right) e^{-\frac{TE}{T_2}} \quad (5.1)$$

## *Inversion Recovery Spin-Echo (IRSE)*

Ecuación 7.28 del libro de [Zhi-Pei Liang \(2000\)](#):

$$A = M_z^0 \left(1 - 2e^{-\frac{TI}{T_1}} + e^{-\frac{TR}{T_1}}\right) e^{-\frac{TE}{T_2}} \quad (5.2)$$

## GRE

Ecuación 7.36 del libro de [Zhi-Pei Liang \(2000\)](#):

$$A = M_z^0 \frac{\sin \alpha \left(1 - e^{-\frac{TR}{T_1}}\right)}{1 - \cos \alpha e^{-\frac{TR}{T_1}}} e^{-\frac{TE}{T_2}} \quad (5.3)$$

## *Spoiled Gradient Echo (SPOILED GRE)*

Ecuación 14.8 del libro de [Bernstein et al. \(2004\)](#):

$$A = M_z^0 \frac{\sin \alpha \left(1 - e^{-\frac{TR}{T_1}}\right)}{1 - \cos \alpha e^{-\frac{TR}{T_1}}} e^{-\frac{TE}{T_2}} \quad (5.4)$$

## *Steady State Free Precession (SSFP)-Free Induction Decay (FID)*

Ecuación 14.15 del libro de [Bernstein et al. \(2004\)](#):

$$A = M_z^0 \tan\left(\frac{\alpha}{2}\right) \left(1 - \frac{(E_1 - \cos \alpha)(1 - E_2^2)}{\sqrt{p^2 - q^2}}\right) \quad (5.5)$$

Donde:

$$E_1 = e^{-\frac{TR}{T_1}} \quad (5.6)$$

$$E_2 = e^{-\frac{TR}{T_2}} \quad (5.7)$$

$$p = 1 - E_1 \cos \alpha - E_2^2(E_1 - \cos \alpha) \quad (5.8)$$

$$q = E_2(1 - E_1)(1 + \cos \alpha) \quad (5.9)$$

## SSFP-ECHO

Ecuación 14.15 del libro de [Bernstein et al. \(2004\)](#):

$$A = M_z^0 \tan\left(\frac{\alpha}{2}\right) \left(1 - \frac{(1 - E_1 \cos \alpha)(1 - E_2^2)}{\sqrt{p^2 - q^2}}\right) \quad (5.10)$$

## BALANCED SSFP

Con alternancia de signos, ecuación 14.23 del libro de [Bernstein et al. \(2004\)](#):

$$A = M_z^0 \sin \alpha \frac{1 - E_1}{1 - (E_1 - E_2) \cos \alpha - E_1 E_2} e^{-\frac{TE}{T_2}} \quad (5.11)$$

Sin alternancia de signos, ecuación 14.24 del libro de [Bernstein et al. \(2004\)](#):

$$A = M_z^0 \sin \alpha \frac{1 - E_1}{1 - (E_1 + E_2) \cos \alpha + E_1 E_2} e^{-\frac{TE}{T_2}} \quad (5.12)$$

## SECUENCIAS CON ETL

En las secuencias como FSE, el parámetro más importante es el ETL, que indica el número de líneas que van a ser adquiridas por cada TR. En este momento, el TE pasa a convertirse en  $TE_{eff}$  e indicaría el momento de adquisición del centro del espacio k. La secuencia quedaría totalmente definida mediante el parámetro *espacio entre ecos* (ESP). De esta forma, en nuestro algoritmo, las líneas del espacio k se escribirían por bloques, donde el número de bloques depende del ETL, y donde cada bloque estaría formado por la *transformada inversa de Fourier o Inverse Fourier Transform* (IFT) de una imagen de contraste en un determinado TE.

$$TE = TE_{eff} - \left(\frac{ETL - 1}{2} - n\right) * ESP; \quad n = 0, 1, \dots, ETL - 1 \quad (5.13)$$

Donde la ecuación para el cálculo de la imagen siguiendo la secuencia FSE sería la correspondiente con la Ecuación 5.1.

### 5.4.4 IMÁGENES GENERADAS

En este apartado se explican detalles sobre la implementación del FR 1.c, que especifica la generación de una imagen médica para cada adquisición simulada.

Las imágenes médicas de manera habitual, están formadas por dos partes: lo que podríamos llamar cuerpo, y que se compone por una matriz n-dimensional, y la cabecera, que contiene información referente a la adquisición de la imagen y a cómo se estructura la información del cuerpo.

En primer lugar, vamos a dedicar especial atención a la información geométrica de la imagen almacenada en la cabecera, pero antes, es necesario poner de manifiesto los distintos sistemas de coordenadas que vamos a tratar, y que se muestran en la Figura 5.14. Éstos son:

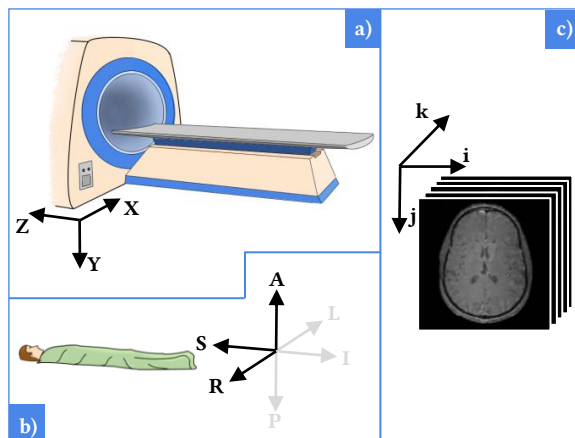


FIGURA 5.14: Sistemas de coordenadas. a) es el sistema de coordenadas del mundo, b) es el sistema de coordenadas del paciente y c) es el sistema de coordenadas de la imagen.

1. *Sistema de coordenadas del mundo* (Figura 5.14a). Define cómo se coloca el paciente en la máquina y, además, fija un acuerdo con respecto a la dirección de los campos magnéticos, donde el campo magnético principal  $B_0$  va a lo largo del eje Z y el campo transversal  $B_1$  sigue el eje X. En la MR se usa habitualmente el sistema coordenadas cartesianas (XYZ), aunque puede variar según la modalidad.
2. *Sistema de coordenadas del paciente* (Figura 5.14b), también llamado sistema de coordenadas anatómicas (Padgaonkar et al., 1978). Se utiliza para describir la posición estándar del cuerpo humano que se define mediante tres planos: axial (que se desplaza a lo largo del eje superior-inferior o S-I), sagital (derecha-izquierda o R-L) y coronal (anterior-posterior A-P). En la MR, el sistema de coordenadas habitual es el llamado *derecha-anterior-superior* o *Right-Anterior-Superior* (RAS).
3. *Sistema de coordenadas de imagen* (Figura 5.14c). Permite movernos a través de la matriz de datos. El origen de coordenadas se encuentra en la esquina superior izquierda y crece hacia a la derecha, hacia abajo y hacia atrás. En las imágenes tratadas, cada celda contiene un vóxel con un valor de intensidad y una coordenada  $ijk$  asignada.

La relación entre el sistema RAS y el de coordenadas de la imagen viene determinado por el tamaño del voxel, típicamente en milímetros y que se traduce en un escalado ( $s$ ); la posición de la imagen, que supone una traslación ( $t$ ); la inclinación de la imagen, definida por una rotación ( $r$ ). En consecuencia, el cambio de base entre las coordenadas RAS y las coordenadas de la imagen vienen dado por la Ecuación 5.14.

$$\begin{bmatrix} R \\ A \\ S \end{bmatrix} = \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (5.14)$$

Por otro lado, para realizar el cambio entre RAS y las coordenadas del mundo, sólo sería necesario invertir los ejes derecha-izquierda y anterior-posterior.



Una vez conocido esto, en el cliente, la biblioteca XTK utilizada las coordenadas del mundo, mientras que en el servidor, el núcleo de simulación usa por defecto las coordenadas RAS. Además, durante todo el proceso de simulación, se utilizan el sistema de coordenadas de la imagen. De forma más detallada, el sistema de coordenadas durante la simulación sería el siguiente:

1. La información relativa a la ROI, proveniente del cliente, estaría en coordenadas del mundo, por lo que será transformada a coordenadas RAS.
2. En la simulación, el modelo anatómico se carga en coordenadas de la imagen, aunque la información de la cabecera sea usada por defecto en coordenadas RAS. Por tanto, la información de la ROI, ya en coordenadas RAS, se transforma a coordenadas de la imagen.
3. Al finalizar la simulación, a la imagen resultante se le debe añadir nuevamente la información geométrica en la cabecera. Esta sería calculada usando la información de la ROI.

Por otra parte, al final de la simulación llevamos a cabo un reescalado de la intensidad entre 0 y 255, lo cual se traduce en ventajas para su descarga, almacenamiento y carga para la visualización posterior en el cliente. Además, también llevamos a cabo una compresión estándar.

En la simulación, se generan dos imágenes. Uno de ellas usa el formato *Neuroimaging Informatics Technology Initiative* (NIfTI), que tiene como propósito la visualización en el cliente, y sólo contiene la información geométrica y los datos de la imagen. La segunda imagen usa un formato DICOM, que contiene, además, información sobre la adquisición y los datos del paciente. Esta última imagen es la que puede ser descargada por el usuario.

## 5.5 RESULTADOS

Los resultados pueden verse en la Figura 5.15. Esta figura muestra el proceso de adquisición de una imagen en nuestra herramienta, el cual pasamos a describir. En primer lugar se comenzaría introduciendo la información del paciente, incluyendo el peso, que es un parámetro relevante para la máquina de resonancia. A continuación se introduce la posición del paciente, tanto la posición de entrada como la parte anatómica (cabeza/pies) que es introducida primero. Posteriormente se selecciona la bobina que va a emplearse y que debe ser coherente con la parte del cuerpo que vaya a ser adquirida. En este momento debe elegirse el protocolo, el cual depende de la patología buscada y la zona donde ésta se espera.

El Panel MR Manager es rellenado con las secuencias que componen el protocolo cuando éste ha sido confirmado. La primera secuencia siempre será el *survey*, la cual, al ser ejecutada, realiza una adquisición rápida del área global cubierta por la antena. Estas imágenes serán usadas posteriormente para la planificación espacial.

Una vez obtenido el *survey*, es el momento de adquirir las siguientes secuencias. En la Figura 5.15 se muestra cómo sería adquirida la última secuencia del protocolo seleccionado; para el resto de las secuencias el proceso sería similar. En primer lugar, se seleccionan los tiempos que definen la secuencia, así como otros parámetros con el ancho de banda o el NEX. Una vez que estos valores son los adecuados, se pasa a realizar la planificación espacial, la cual puede ser realizada sobre el *survey* o sobre cualquier otro volumen ya adquirido. La planificación que se muestra en la figura 5.15 ilustra el proceso. En el momento en que todos los parámetros han sido fijados, es

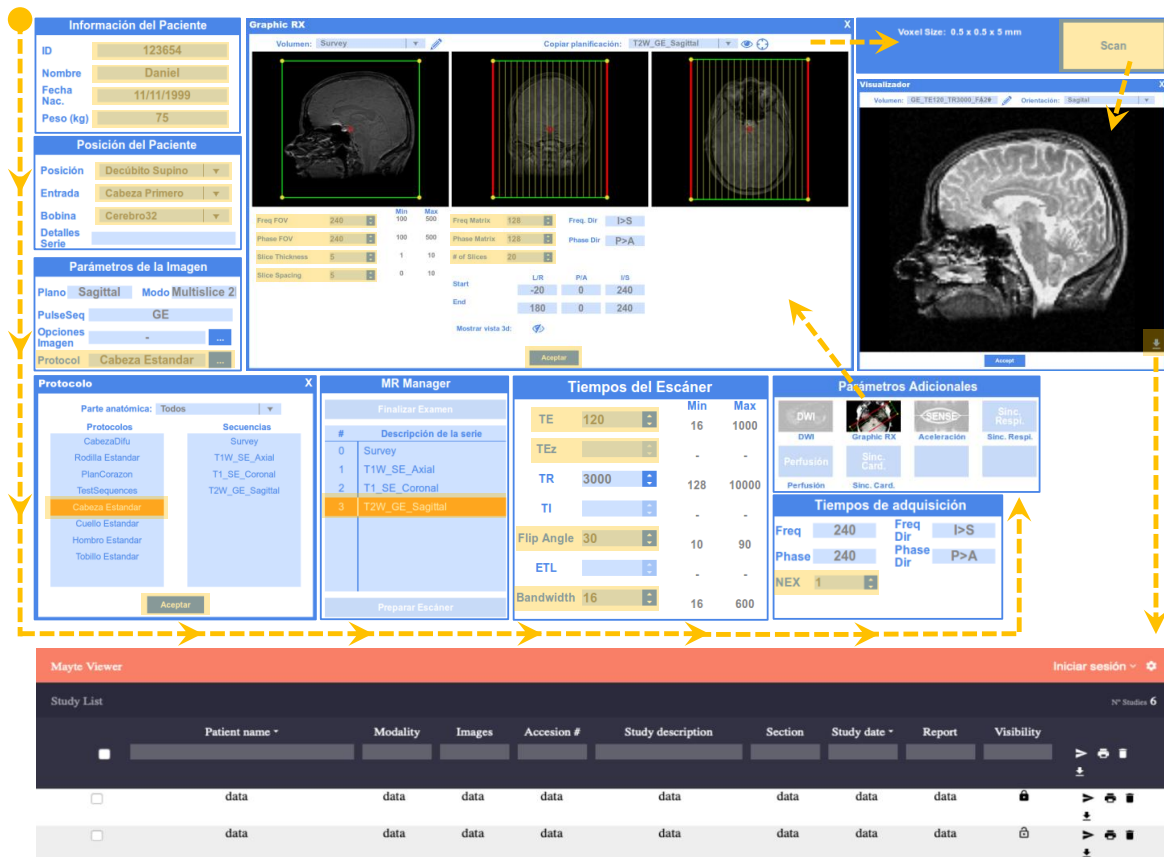


FIGURA 5.15: Ejemplo del manejo que realizaría un estudiante de nuestra herramienta. El camino naranja formado por las líneas y flechas discontinuas indica el orden en el que se realiza el proceso de adquisición de una zona anatómica, en este caso, un cerebro. El proceso pasa por rellenar las zonas iluminadas en los distintos paneles hasta pulsar “scan”, visualizar la imagen adquirida y guardarla correctamente.

posible iniciar la adquisición pulsando al botón “Scan”. Cuando la simulación finaliza, la imagen resultante es visible y puede ser descargada.

Finalmente, hemos habilitado un repositorio encargado de almacenar imágenes DICOM en una herramienta externa. De esta forma, cuando la adquisición sea descargada, ésta puede ser subida a este sistema. De esta forma se imita el paso 12 del proceso de adquisición (Sección 2.1.2).

También debe destacarse alguno de los efectos que pueden ser generados mediante la simulación. La Figura 5.16 muestra algunos de éstos. Para comenzar, se enseña cómo ha sido realizada la planificación —incluyendo la vista en 3D— de la imagen sobre la que van a realizar las simulaciones (Figura 5.16a). Se comienza con una imagen ruidosa debido a la bobina utilizada, la cual cuenta con una SNR baja (Figura 5.16b). La siguiente adquisición muestra el efecto de desplazamiento químico (Figura 5.16c), producido por un ancho de banda demasiado bajo.

La estimación del ADC también ha sido probada, la cual es adquirida mediante una regresión lineal en el dominio logarítmico una vez fijados los b-valores, como puede verse en la Figura

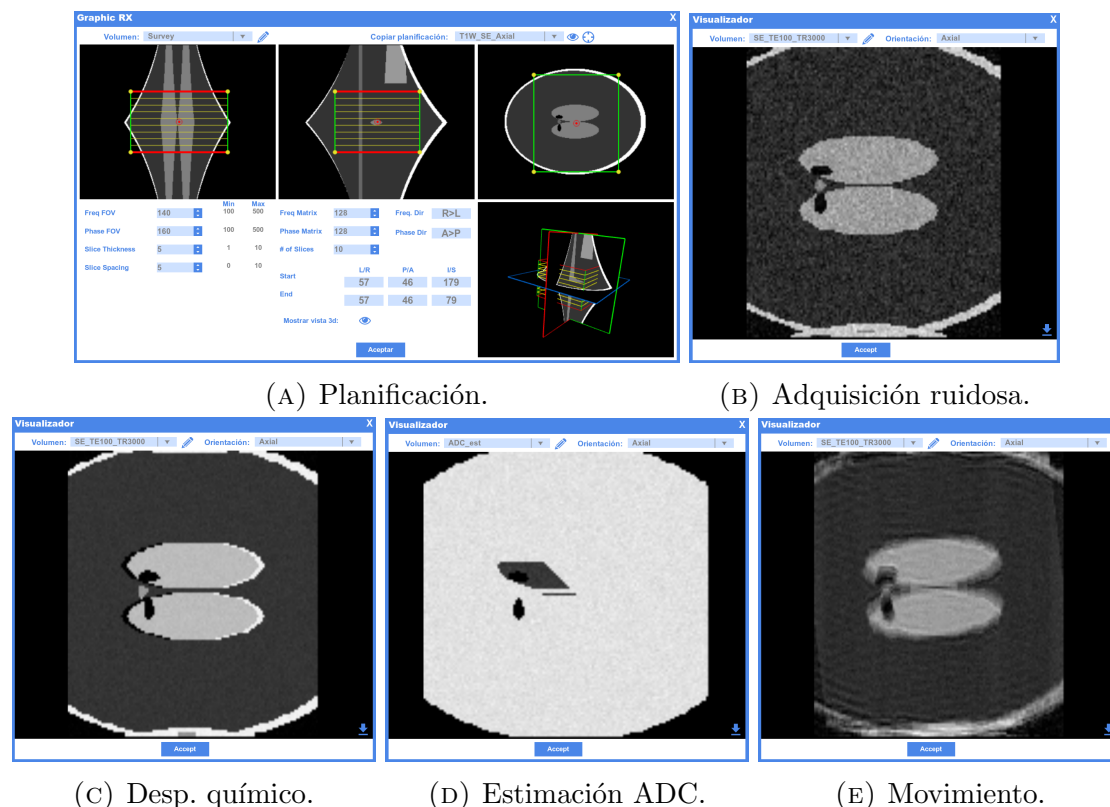


FIGURA 5.16: Simulaciones. Las figura (B), (C), (D) y (E) muestran los resultados de las simulaciones realizadas con la herramienta a partir de la planificación realizada en la figura (A).

5.16d. Para terminar, se ha probado otro de los artefactos, en este caso, el debido al movimiento del paciente durante la adquisición (Figura 5.16e).

Otro de los útiles interesantes en el aprendizaje de resonancia es el que posibilita el estudio de situaciones aisladas que se dan en esta materia. Nuestra herramienta de ejemplos actualmente permite la ejecución de los siguientes casos:

- **Submuestreo** Da la posibilidad de submuestrear el espacio  $k$  en la dirección de fase y ver su resultado sobre la imagen. Permite los factores de submuestreo (1,2,3), los cuales son de uso habitual en la práctica clínica.
- **Filtro rectangular** Permite aplicar un filtro rectangular al espacio  $k$  y ver su efecto en la imagen. El tamaño del rectángulo puede ser fácilmente variado mediante la utilización de barras deslizantes.
- **Movimiento** Esta característica es similar a la observada en la consola principal (Figura 5.16e) y permite ver el efecto sobre la imagen del movimiento del paciente durante una adquisición. Puede seleccionarse tanto el sentido del movimiento como su intensidad.
- **Escritura del espacio  $k$**  Esta herramienta realiza una simulación de cómo el espacio  $k$  está siendo adquirido y cómo sería el resultado de la imagen de intensidad en cada uno de los instantes.

- **Planificación radial** En este ejemplo es posible comprender cómo sería una adquisición radial de una imagen y entender los elementos que la conforman. Permite variar el número de cortes así como el FOV del corte.

Finalmente, el actual visor de casos permite el estudio de 22 casos clínicos de diversas zonas anatómicas, como cabeza, zonas lumbar, cervical, dorsolumbar y espina dorsal. Cada uno de estos casos cuenta con su correspondiente archivo explicativo y las adquisiciones realizadas para su diagnóstico.

## 5.6 CONCLUSIONES

Como muestra la Figura 5.15, la interfaz gráfica de nuestro simulador es una completa herramienta que mimetiza el complejo manejo de la consola de una máquina MR y que cuenta con una gran cantidad de sus útiles. Además, su desarrollo modular y estructurado, permite la mejora continua de este elemento y trata de minimizar el trabajo invertido en su mantenimiento, que es uno de los problemas más actuales en el desarrollo de software de hoy en día (Mall, 2013).

Cabe recalcar algunas características de nuestro simulador. Hasta donde sabemos, no se ha reportado el uso de una herramienta de planificación de la misma versatilidad que la nuestra en ningún simulador MR. De la misma forma, nuestra herramienta educativa está no sólo formada por el simulador propiamente dicho, sino que cuenta con otro conjunto de útiles, como los ejemplos o los casos clínicos, que aportan mayor valor a la hora de enseñar esta materia. Finalmente, uno de sus principales potenciales está en su diseño web, que permite su acceso y su ejecución desde una gran cantidad de dispositivos y localizaciones.

En conclusión, las decisiones de diseño e implementación tomadas han permitido la creación de una herramienta educativa de gran potencial, pues permite la realización del flujo completo que un técnico radiólogo debe realizar delante de la consola MRI (visto en la Sección 2.1.2), cumpliendo así uno de los principales objetivos de esta presente tesis.

Por otro lado, el núcleo de simulación hace uso de las expresiones algebraicas que serán descritas en la Sección 5.4.3. A estas ecuaciones, además, se le añaden posteriormente efectos a la imagen, los cuales surgen en función de los parámetros seleccionados. Como consecuencia, nuestro núcleo, es capaz de obtener imágenes simuladas en cortos tiempos de cálculo, de orden de segundos. Además, estas imágenes serán visibles en la interfaz poco después de su finalización, dadas las optimizaciones realizadas en su creación.

Todo ello, hace que el usuario pueda probar una gran cantidad de configuraciones, y que obtenga el resultado de cada una de ellas en un periodo corto de tiempo. De esta forma, se han cumplido cada uno de los FR y NFR vistos en la Sección 5.2, además del primer objetivo específico fijado en la Sección 1.2. Por otra parte, en el capítulo 7 mostraremos que, en efecto, la herramienta tiene interés demostrado para el aprendizaje de la adquisición de resonancia magnética con respecto a la enseñanza tradicional.

Puede encontrarse más información sobre nuestra herramienta en la guía de uso situada en el Apéndice E.

## Capítulo 6

# INCLUSIÓN DE UN ITS EN EL SIMULADOR

---

**E**STE capítulo se centra en la integración de un ITS en nuestro simulador de resonancia magnética. En primer lugar, se exponen las razones de elección del sistema de tutoría. A continuación se diseña la estrategia sobre la base de los objetivos de integración. En tercer lugar, se describen diversos detalles de implementación. Finalmente, se muestran los resultados y se indica el grado de cumplimiento de los objetivos.

## 6.1 INTRODUCCIÓN

---

### 6.1.1 ELECCIÓN DE UNA HERRAMIENTA DE TUTORÍA

---

Como ha podido verse en el Capítulo 4, los ITS han sido muy utilizados en el aprendizaje de diversas materias. Son abundantes los ejemplos aplicados a asignaturas como las matemáticas (Melis and Siekmann, 2004) o la física (Schulze et al., 2000). Sin embargo, lo que llamó nuestra atención fue el uso de estos sistemas en la instrucción de procedimientos complejos. Algunos ejemplos, como ya hemos mencionado en el Capítulo 4, son Jeon et al. (2007) y Kazi et al. (2012); recordemos que los autores del primero crearon un ITS para el adiestramiento en el uso de una máquina de anestesia y en el segundo se hace uso del aprendizaje basado en problemas para mejorar las habilidades de razonamiento clínico.

Por lo que nos ocupa en la tesis, el procedimiento para la adquisición de una imagen de MR puede dividirse en diferentes bloques, según hemos puesto de manifiesto en la Sección 2.1.2. En este caso, el punto de inicio del problema es claro, aunque la solución a éste no es necesariamente única. Hay que tener en cuenta que, en función de la patología que se esté buscando, los pasos internos que deben darse en cada bloque pueden variar. Además, para minimizar los posibles artefactos que ocurren en una adquisición, pueden emplearse diversas estrategias. No obstante, dentro de un sistema específico, las soluciones a un problema de este carácter sí son finitas y están bien delimitadas. Teniendo esto en cuenta y basándonos en lo visto durante el Capítulo 4, el problema encaja adecuadamente con el uso de un modelo cognitivo del tipo CB o RB sin la necesidad de recurrir a sistemas más complejos.

Uno de los principales problemas de los ITS es la gran cantidad de tiempo necesario en el desarrollo del sistema por cada hora de instrucción (Murray, 2003; Woolf, 2010). En consecuencia, es oportuno el uso de una herramienta que simplifique la creación de un ITS. Para la elección de una de estas AT (véase la sección 4.4), es necesario tener en cuenta el estado del simulador de MR. En el momento en el que se hizo este planteamiento, la versión de nuestro simulador ya había sido probada y resultaba estable. Además, para su implementación, se ha hecho uso de AngularJS, como se ha indicado en el Capítulo 5. Esto crea la necesidad de usar un sistema con un bajo acoplamiento entre la interfaz y el sistema de tutoría (sección 4.5), quedando descartadas muchas de las herramientas vistas en la Sección 4.4.2. A su vez, ello hace que se centre la atención sobre herramientas como xPST, ASAT o CTAT. xPST es uno de los sistemas que mejor se alinea con nuestra exigencia pero, sin embargo, fue descartado, dado que el proyecto no cuenta con soporte desde 2009. Por otro lado, el modelo seguido por ASAT no encaja con nuestros objetivos. Finalmente, el popular sistema CTAT se ajusta al modelo buscado, cuenta con una gran documentación y tiene soporte para su ejecución en un navegador.

Todo ello determinó que la elección del tutor fuese un MT (Aleven et al., 2009b). Este sistema, además, es sencillo y rápido de crear mediante las herramientas ofrecidas por CTAT. Sin embargo, y al igual que la mayoría de las herramientas estudiadas, esta utilidad no puede ser utilizada directamente en nuestro sistema. En consecuencia, debe diseñarse e implementarse un programa intermedio que compatibilice nuestro simulador y CTAT. Este proceso se describe a lo largo del presente capítulo.

## 6.1.2 DETALLES ADICIONALES SOBRE CTAT

En la Sección 4.4.2 se afirmó que CTAT (Aleven et al., 2016) proporciona dos herramientas, CTAT HTML Editor y CTAT Behavior Recorder, y se indicó la finalidad de cada una de ellas. Esto resulta útil para hacerse una idea de cómo utilizar CTAT en la creación de un sistema de tutoría para la resolución de un problema concreto; sin embargo, es necesario entrar más en detalle. La Figura 6.1 contiene un esquema para aclarar las interacciones que CTAT lleva a cabo y ver los diferentes elementos que lo componen.

- **CTAT.** Como hemos indicado, se compone de dos herramientas para la generación de tutores :
  - *CTAT HTML Editor.* Es un editor web que permite la creación de una GUI empleando los componentes propios de CTAT. La figura 6.2b muestra esta herramienta. En la parte izquierda de la misma puede seleccionarse el componente, mientras que en la parte derecha, una vez fijado el componente, pueden modificarse sus propiedades.
  - *CTAT Behavior Recorder.* Esta herramienta es principalmente empleada para la generación del *Behavior Recorder graph* (BRD) en los tutores de *seguimiento de ejemplo o Example-Tracing* (ET) (este término se define a continuación). En la Figura 6.2a puede verse un ejemplo de la herramienta y del gráfico de comportamiento que es posible crear.
- **Interfaz del estudiante.** Es el conjunto de componentes con los que el estudiante va a interactuar. Estos componentes están escritos en JS y son propios de CTAT, por lo que

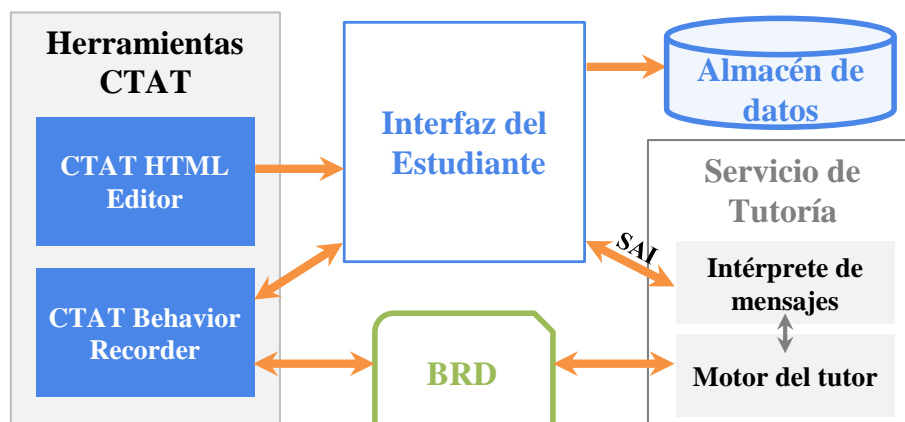


FIGURA 6.1: Esquema de CTAT que muestra las distintas partes que compondrían la creación del ITS. En la parte izquierda se sitúan las Herramientas de CTAT que permiten la creación del BRD y de la interfaz. Con la interfaz y el BRD ya creados, el estudiante haría uso (de forma transparente para él) del sistema de tutoría a través del servicio de tutoría.

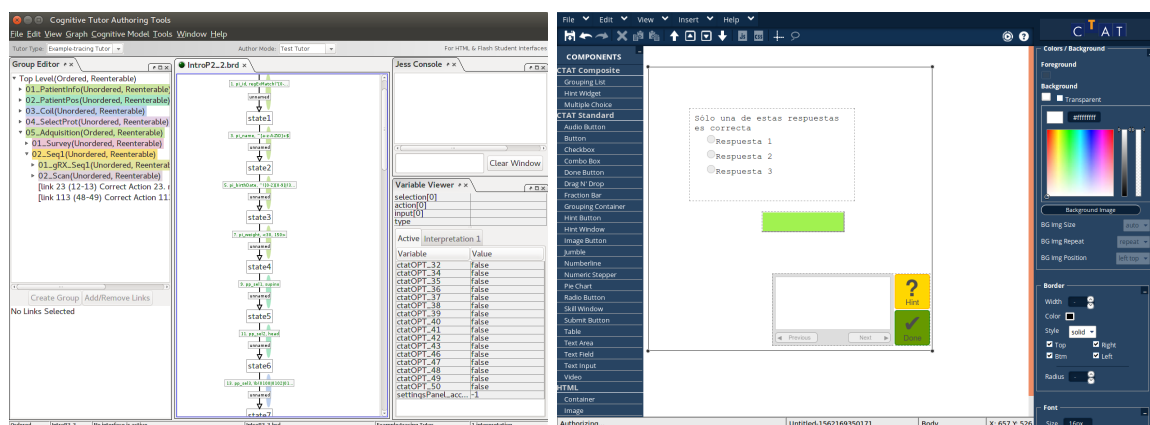
son capaces de emitir y recibir distintos eventos.

- **BRD.** *Behavior Recorder graph* (BRD) (o gráfico de comportamiento) es realmente un archivo XML. Su función es almacenar información referente a los modelos (Sección 4.2), principalmente de tutoría y de dominio. Entre estos datos pueden encontrarse los estados posibles en el problema, las acciones necesarias para la transición entre estados o los comentarios o sugerencias que pueden darse.
- **Almacén de datos.** Cada acción realizada en la interfaz es susceptible de ser almacenada. Para su almacenamiento, los datos son enviados de forma anónima a un servidor previamente configurado. Este servidor podría ser, por ejemplo, la TutorShop (descrito en la Sección 4.4.2) u otro propio. Los datos contienen información diversa sobre cómo se ha desarrollado el problema, los errores cometidos, el tiempo empleado, las peticiones de ayuda, etc. Posteriormente el conjunto de estos datos puede ser usado con propósitos investigadores.
- **Servicio de tutoría.** Es el módulo encargado de seguir la actividad realizada por el alumno. En función del trabajo percibido, interpreta el modelo contenido en el BRD y realiza las acciones pertinentes. En este servicio pueden distinguirse dos partes:
  - *Intérprete de mensajes.* Es la parte encargada tanto de emitir como de recibir eventos de la interfaz. *Selección, acción, entrada o Selection, Action, Input* (SAI) es el evento principal por el cual la interfaz interactúa con el tutor y se genera cada vez que el estudiante realiza un paso. Por otro lado, los eventos que envía a la interfaz son indicadores que resultan del paso dado por el alumno. Podrían indicar si el paso ha sido correcto o no, si es el último paso del problema, si debe iluminarse un elemento de la interfaz o si debe producirse un comentario.



- *Motor del tutor.* Este módulo tiene acceso al BRD y es el encargado de interpretar la información que recibe a través del *intérprete de mensajes* e indicarle qué evento debe mandar.

Los bloques del esquema de la Figura 6.1 no intervienen simultáneamente. La parte izquierda de la figura, las herramientas de CTAT, sólo se utilizan durante la creación del problema, que es llevada a cabo por el personal experto en la materia. La parte derecha, el servicio de tutoría y el almacén de datos, por su parte, interviene únicamente mientras el estudiante resuelve el problema. Finalmente, la parte central interviene en ambos procesos.



(A) CTAT Behavior Recorder.

(B) CTAT HTML Editor.

FIGURA 6.2: Herramientas que componen CTAT.

## 6.2 ANÁLISIS

Como hemos indicado al final de la sección 6.1.1 nuestro objetivo es la incorporación de CTAT a nuestro simulador, que debe realizarse en la parte del cliente. Con la asesoría del personal experto de la SERAM, comenzamos con la recopilación de los requisitos que dicha incorporación debía cumplir. Ello dio lugar a los requisitos que exponemos a continuación. En primer lugar, los FR son los siguientes:

1. El simulador debe ser capaz de generar un tutor ET. Para ello, su interfaz debe ser compatible con la herramienta CTAT Behavior Recorder.
2. El simulador debe permitir la ejecución de los ejercicios por parte de los alumnos. En consecuencia, los componentes del simulador deben tener una comunicación bidireccional con el servicio de tutoría.
3. Los componentes deben poder enviar la información relacionada con el problema a un almacén de datos.



4. En momentos determinados, es interesante enviar de forma simultánea —en vez de secuencialmente— un conjunto de parámetros de simulación. Por tanto, debe estar disponible la posibilidad de enviar simultáneamente varios parámetros para su registro en el ITS.
5. El requisito anterior crea una nueva necesidad, a saber, deben poderse seleccionar los parámetros que van a ser registrados por el ITS dentro del conjunto de parámetros enviados.
6. Además, teniendo en cuenta el objetivo educativo del simulador, será necesario que, para un problema dado, las propiedades iniciales del simulador sean adecuadas. Por este motivo debe ser posible la activación/desactivación de distintos artefactos o características por parte del tutor.
7. En la resolución del problema por parte del estudiante, debe mostrarse un enunciado, el cual debe poder ser consultado en cualquier momento. De esta forma, el objetivo del problema estará siempre claro.
8. El simulador debe contar con una herramienta que permita seleccionar la temática de los ejercicios y, dentro de ésta, elegir el ejercicio que va a resolverse.
9. El sistema de tutoría debe iniciarse en modo desactivado y sólo debe ser activado cuando se vaya a realizar un ejercicio.

Así mismo, planteamos los siguientes NFR:

1. El sistema de tutoría no debe afectar ni al rendimiento ni al funcionamiento de la interfaz.
2. El desarrollo de la integración del ITS debe ser independiente del desarrollo del simulador.

Estos dos requisitos parecen bastante obvios ya que el ITS está destinado a ayudar a los estudiantes, por lo que la interacción entre el simulador debe mantenerse para cumplir con sus propios requisitos (como se indica en la sección 5.2); de lo contrario, el ITS en sí mismo puede evitar que los estudiantes usen el simulador, una consecuencia que haría que este esfuerzo no valiese la pena. Además, el simulador sigue evolucionando, por lo que deben poderse incorporar funcionalidades adicionales en cualquier momento; por lo tanto, el ITS no debería ser un obstáculo para la mejora del simulador.

## 6.3 DISEÑO

---

Al igual que en el Capítulo 5, se comenzará por un diseño de alto nivel del sistema, donde se detalla la estrategia general y el diseño de los paneles en caso de ser necesario. A continuación, se detalla el diseño del programa.

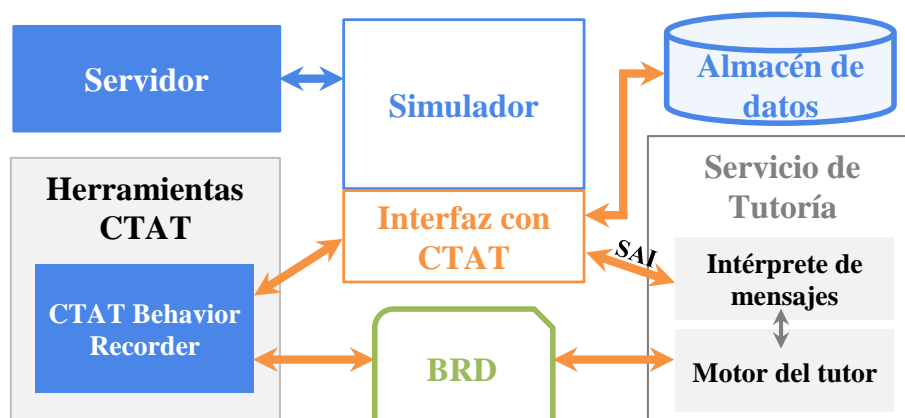


FIGURA 6.3: Diseño para la integración de CTAT en nuestro simulador. Con respecto a la figura 6.1, se ha prescindido de la herramienta que permite la creación de la GUI, puesto que ya se cuenta con una propia. Además, se ha añadido a nuestra GUI una interfaz que permite la comunicación entre nuestra GUI y los servicios ofrecidos por CTAT.

### 6.3.1 DISEÑO DEL SISTEMA

La forma de afrontar el problema es mediante el diseño de un programa intermedio que compatibilice nuestro simulador y CTAT. Esta idea se recoge en la Figura 6.3 la cual, si bien inspirada en la Figura 6.1, incluye los bloques de contexto en los que debe incluirse CTAT, así como el bloque de solución que se propone en esta tesis. Concretamente, se ha introducido nuestro sistema de simulación, formado por el bloque *Simulador*, que representa la interfaz, y el bloque *Servidor*, que representa nuestro servidor. En segundo lugar, dado que no es necesario crear una interfaz, se ha eliminado la herramienta CTAT HTML Editor. Finalmente, el bloque llamado *Interfaz con CTAT* simboliza el programa que se va a crear. Este bloque será el encargado de gestionar las interacciones entre el simulador y los servicios de CTAT.

La solución propuesta en la Figura 6.3 permite de forma natural hacer que nuestro sistema sea independiente del desarrollo del simulador (NFR 2) y que no intervenga en el funcionamiento de la interfaz (NFR 1).

### 6.3.2 DISEÑO DEL PROGRAMA

#### CONSIDERACIONES SOBRE CTAT

Para poder diseñar la aplicación, es necesario conocer más detalles sobre cómo se ejecuta CTAT. Este programa está pensado para que se ejecute en dos etapas:

- La primera etapa tiene lugar durante la carga de la página y se realiza de forma automática. En ésta, primero se inicializa CTAT según la configuración preestablecida y posteriormente se crean todos los componentes propios de CTAT que van a ser utilizados durante el problema. La creación tan prematura de los componentes puede generar inconvenientes a

la hora de utilizar componentes dinámicos, es decir, componentes que sean creados durante la resolución del problema —y no necesariamente al principio—.

- La segunda etapa abarcaría o bien la resolución del problema, o bien su creación. El modo determina cuál de ellas se está ejecutando.
  - El modo por defecto fijaría la resolución del problema, donde los componentes envían y reciben información del servicio de tutoría. Existe, no obstante, una particularización de este modo llamada “Quiz”, la cual permite la evaluación de un ejercicio sin que el usuario reciba comentarios o indicaciones.
  - El segundo modo, también llamado *autoría* (Authoring) determina que el ejercicio se está creando. Durante este modo pueden existir tres estados; el primero habilita la creación del estado inicial del problema, en el segundo se crea el BRD y finalmente, el tercer modo permite probar el funcionamiento del ITS para este problema.

## CONSIDERACIONES PARA LA IMPLEMENTACIÓN

Durante la primera carga de la página, AngularJS, de forma similar a CTAT, genera en su inicialización todos los componentes de la aplicación y ejecuta parte de su código. Por tanto, en la carga inicial de la página deben tenerse en cuenta las siguientes consideraciones de cara a la implementación:

1. Determinar el orden correcto de carga entre CTAT y AngularJS. Este orden debe respetarse de forma estricta para evitar fallos. El orden que hemos establecido de sucesión es primero AngularJS y posteriormente CTAT.
2. Antes del lanzamiento de CTAT, el sistema debe ser capaz de identificar si es para llevar a cabo la resolución de un problema (modo por defecto) o bien es para su creación (modo autoría) y, en este segundo caso, distinguir entre los tres estados comentados.

Por otro lado, la generación de componentes dinámicos en nuestro simulador es habitual. Un caso claro se da cuando se carga un protocolo en el panel RMMManager, aunque también sucede cuando se abren algunos de los pop-up, como el panel de protocolos. Sin embargo, CTAT presupone que todos los componentes necesarios existen antes de llamar a la función *initTutor()*. Como solución, en la documentación de CTAT se propone la implementación de una función que resuelva específicamente esta problemática. En consecuencia, nuestra idea de diseño inicial fue usar la metodología sugerida por CTAT para los componentes dinámicos.

## FLUJO DEL SIMULADOR CON ITS

Teniendo en cuenta las consideraciones previas, el diseño del flujo de funcionamiento, suponiendo que CTAT esté activado, sería el siguiente:

1. Creación de los componentes. Una vez cargado AngularJS, los componentes de CTAT serán creados mediante el uso de directivas (Sección B.5).

2. Inicializar CTAT. En función de la configuración, esta inicialización lleva a dos estados:

Creación de un problema por parte del personal experto (modo autoría):

- a) Establecer el estado inicial del problema.
- b) Resolver el problema para generar el gráfico de decisiones. Durante este apartado se produciría la creación de nuevos componentes.

Resolución del problema por parte del estudiante (modo por defecto):

- a) Presentar el enunciado.
- b) Resolver el problema hasta que el tutor le indique que es correcta la solución. Durante este punto se produciría la generación de componentes dinámicos.

Más en detalle, durante la creación del problema, como se ha visto, el primer paso es establecer el estado inicial de tal problema. En nuestro caso se espera que se elijan los artefactos o las propiedades de simulación que se estimen adecuadas en el panel de configuración del simulador (FR 6). Adicionalmente, y para cumplir el FR 5, durante este paso se establecerán los elementos que van a ser registrados por el simulador. Para ello, se añadirá una opción en el selector del panel de configuración y se llamará CTAT. Nos referiremos a él como panel de registro de componentes. Al activar esta opción, aparecerán en el panel las opciones de componentes registrables. Por otro lado, estas decisiones deben ser transparentes para el estudiante; por lo tanto, durante la ejecución del problema estas opciones no deben estar disponibles para el usuario. En consecuencia, la opción de elegir los elementos que serán registrados sólo será visible cuando CTAT esté activado y en modo autoría.

El siguiente paso es la creación del BRD. Para tal fin el experto utiliza el simulador mientras la herramienta CTAT Behavior Recorder (FR 1) registra los pasos que el experto realiza. Por supuesto, los pasos registrados sólo afectarán a los componentes elegidos y sólo deben funcionar cuando CTAT esté activado —y en modo autoría—. La creación de estos componentes se llevará a cabo mediante el uso de directivas. Estas directivas, si están bien implementadas, deberían solventar los FR 1, 2 y 3. Se debe crear un tipo de directiva por cada tipo de componente empleado y éstas deben ser capaces de realizar las mismas acciones que los componentes de CTAT.

En la resolución del problema, el FR 7 habla de un enunciado. Este panel es muy similar a los ya descritos en el Capítulo 5.4.1. Su colocación más adecuada sería en el panel del administrador, ya que de esta forma no puede interpretarse como parte del proceso de simulación. El enunciado, además, debe aparecer desplegado al inicio del problema para que sea lo primero que los alumnos vean. Por tanto, es necesario que el panel detecte el estado de CTAT y sólo aparezca cuando trate de resolverse un problema concreto. Además, durante este estado deben seguir cumpliéndose los FR 1, 2, 3.

Todos los elementos descritos deben ser conocedores, en cierta medida, de la información sobre el estado del tutor. Por tanto, en la implementación debe tenerse en cuenta la existencia de un elemento en común que pueda comunicar con las piezas necesarias para la *Interfaz con CTAT*. Este elemento encaja con la definición de servicio de AngularJS (Sección B.4).

Para el cumplimiento del FR 9 se diseña la aplicación de forma que, por defecto, el ITS esté desactivado y, cuando se quiera activar, el usuario debe acceder a la zona donde el simulador se cargue directamente con el ITS activado.

## PANELES

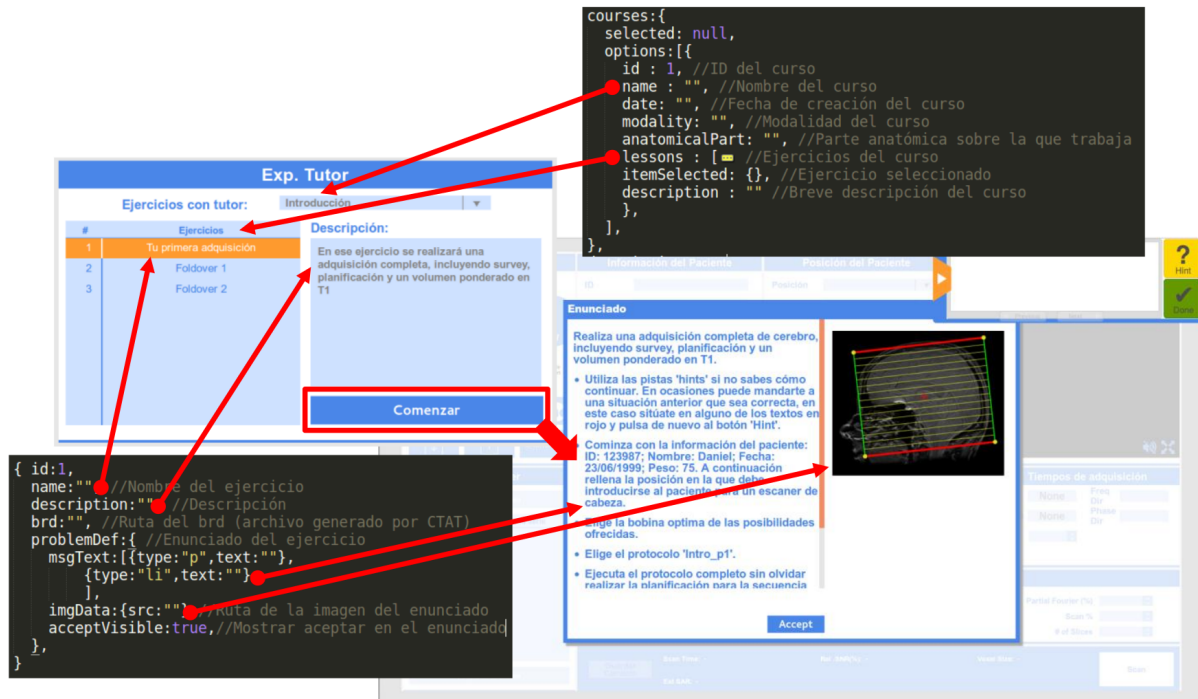


FIGURA 6.4: Elección del problema y enunciado. La figura muestra cómo el código empleado en el panel de selección del ejercicio de tutoría es también utilizado durante la ejecución del ejercicio para el enunciado.

Con la incorporación de CTAT los FR 5–8 requieren del diseño de nuevos tipos de paneles o la modificaciones de los ya existentes:

- Panel de elección del problema** (Figura 6.4) El diseño de este panel se estructura de arriba abajo y de izquierda a derecha. Se comenzaría con la selección de la temática mediante un panel desplegable situado en la parte superior. A continuación, se muestran en la parte izquierda, los ejercicios contenidos en la temática seleccionada. Para conocer más detalles sobre el ejercicio, sería posible ver una descripción a la derecha. Finalmente, en la parte inferior derecha, se habilita un botón para comenzar con el ejercicio.
- Panel del enunciado** (Figura 6.4) Debe contener el enunciado del problema que va a realizarse. Además, se diseña para que el enunciado sea definido una única vez para cada problema, como veremos a continuación. Su diseño se divide en dos, de forma que a la izquierda quede contenido el enunciado y a la derecha se muestren imágenes que puedan servir como guía al usuario.
- Panel de registro de componentes** (Figura 6.5) Este panel debe enumerar todos los componentes registrables del problema (FR 4). Esta opción se dará en dos casos: en el panel de la planificación geométrica y cuando se envíen los datos para la realización del escáner. Para comodidad del usuario, todas las opciones deben ser cargadas con un valor por defecto.



FIGURA 6.5: Panel de registro de componentes.

Además, en la Figura 6.4 se especifica el diseño que debe seguir la información, de forma que la temática, los ejercicios y el enunciado sólo estén definidos una vez. La estrategia consiste en la creación de datos anidados, donde la relación entre estos 3 elementos sea inequívoca.

## 6.4 IMPLEMENTACIÓN

En esta sección se describe la implementación de las diferentes partes que deben crearse para cumplir el diseño. En este nivel ya se entrará en detalles del código. En primer lugar describiremos el servicio compartido por los diferentes elementos que conforman la interfaz con CTAT. En segundo, detallaremos las directivas implementadas y su funcionamiento.

### 6.4.1 SERVICIO

El servicio llamado *dtCTATTools* ha sido creado para compartirse a través de los diversos componentes que conformen la interfaz con CTAT. Su finalidad es contener variables en común y proporcionar herramientas para la configuración, inicialización y seguimiento de CTAT. A continuación se muestran las principales variables y las funcionalidades de este servicio:

```

1 service('dtCTATTools', ['$rootScope', 'coursesData', function
  ($rootScope, coursesData){
2
3   //Variables principales
4   //-----
5   var data = {}; //Información que podrá ser accedida a
      través de los diferentes servicios.
6   data.CTATEnabled = true/false; //Determina si CTAT está
      activado o no

```

```

7   data.CTATMyVars = {}; //Objeto que contiene la
      configuración de CTAT
8   data.myProblem = 1; //Número que indica el problema al
      que debe llamarse
9   data.myCourse = 1; //Número que indica el curso al que
      pertenece el problema
10  data.graphConfig = null; //Meta-información del gráfico (
      highlight, feedback, etc)
11  data.problemConfig = null; //Meta-infomración del
      problema "Set Start State","Demonstrate","Test Tutor"
12  data.recordAssociatedRules = { //Las reglas asociadas
      contienen diversa información sobre el funcionamiento
      de CTAT. Entre esta información podría encontrarse si
      el paso dado ha sido interpretado como correcto o no.
13    flag:true/false, //Determina si las reglas van a ser
      grabadas o no
14    recorded: [], //Vector que contiene las reglas grabadas
      , cada una será un objeto diferente
15  };
16
17  //Funcionalidades principales
18  //-----
19  var dtCTATToolsPointer = {}; //Objeto contenedor de los
      diferentes servicios. Da acceso a la variable 'data'
      mediante funciones para su lectura y escritura. De é
      stas, sólo serán comentadas setCTATAuthoring y
      setCTATDefault.
20  dtCTATToolsPointer.setCTATAuthoring = function(){ ... };
      //Establece la configuración para el modo de autoría.
21  dtCTATToolsPointer.setCTATDefault = function(){ ... }; //
      Establece la configuración para el modo por defecto.
22  dtCTATToolsPointer.isFeedbackAllowed = function(){ ...
      }; //Función que determina si el problema se está
      resolviendo en modo Quiz.
23  dtCTATToolsPointer.initListener = function(){ ... }; //
      Permite la escucha de los eventos que suceden en CTAT
24  dtCTATToolsPointer.CTATComponentInitialized = function(){
      ... }; //Permite la comprobación de los componentes
      inicializados
25  dtCTATToolsPointer.initCTATLoader = function(){ ... }; //
      Función encargada de inicializar CTAT
26  dtCTATToolsPointer...
27  }]);

```

En la línea 1 a este servicio se le inyectan los objetos `$rootScope` y `courses-Data`. Se han

destacado únicamente estos dos porque son importantes para entender el funcionamiento. El objeto `$rootScope` se utiliza para forzar la actualización de componentes del ámbito principal ya que en algunos casos AngularJS no era capaz de lanzar el bucle de procesamiento. Por otro lado, `coursesData` se utiliza para saber si el problema del curso seleccionado existe. Las variables contenidas en el objeto `data` sólo son accesibles a través del objeto `dtCTATTool- sPointer` (línea 19), que es el único instanciado al inyectar este servicio. Las variables que alberga pueden ser definidas en distintos instantes. Al comienzo de la carga de la página se define `CTATEnabled` (línea 6) y sirve para habilitar CTAT. La variable `CTATMyVars` (línea 7) contiene todos los parámetros necesarios para la configuración de CTAT y se escribe justo antes de que CTAT sea inicializado. Las variables `myProblem` y `myCourse` (líneas 8 y 9 respectivamente) determinan el problema que va a ser resuelto en el modo por defecto; en el modo de autoría no tienen uso. La existencia del problema es verificada mediante el servicio `coursesData`. Por otro lado, `graphConfig` y `problemConfig` (líneas 10 y 11 respectivamente) albergan información del gráfico y del problema, respectivamente. Finalmente, la variable `recordAssociatedRules` (línea 12) va a ser usada para almacenar todos los mensajes provenientes del servicio de tutoría.

Continuando con el código, en las principales funciones mostradas se encuentra `isFeedback-Allowed` (línea 22), la cual permite saber si el ejercicio se está ejecutando en modo Quiz o no, de forma que podamos así cumplir una de las consideraciones del diseño.

Hagamos un análisis más detenido del resto de funciones; la función `initListener` (línea 23 del panel superior y un fragmento de su código se encuentra en el panel siguiente) se detalla a continuación y es la que permite escuchar los eventos que suceden entre los componentes y el servicio de tutoría. Para nuestro desarrollo sólo es necesario la escucha de tres eventos:

- `StateGraph` (línea 4, panel siguiente) que nos proporciona información sobre el estado del gráfico de comportamiento, tanto si está siendo ejecutado en modo Quiz o no.
- `StartProblem` (línea 5), que proporciona información sobre el problema, es decir, si está o no en modo autoría o en qué estado se encuentra, a saber, estado inicial, creación del gráfico o probando el ejercicio.
- `AssociatedRules` (línea 6) escucha los eventos de las reglas asociadas. En este caso el indicador contiene información sobre si es correcta o no y la selección nos proporciona la id del elemento que ha sido analizado.

Esta función finaliza incluyendo en el procesador de eventos de CTAT (línea 21).

```

1 dtCTATToolsPointer.initListener = function() {
2   var startStateEndListener = {
3     processCommShellEvent: function (anEvent, aMessage) {
4       if(anEvent.trim() == "StateGraph"){... }
5       if(anEvent.trim() == "StartProblem"){...}
6       if("AssociatedRules" == anEvent ) {
7         var indicator = aMessage.getIndicator();
8         var sai = aMessage.getSAI(); // selection-action-
          input from tutor engine
9         var selection = (sai ? sai.getSelection() : "
          _noSuchComponent_");

```



```

10     if(data.recordAssociatedRules.flag==true){
11         var assRule = {
12             id:selection,
13             indicator: indicator,
14         }
15         data.recordAssociatedRules.recorded.push(assRule)
16     }
17 };
18 }//End processCommShellEvent
19 };//End startStateEndListener
20
21 if(CTATCommShell.commShell!=null){
22     CTATCommShell.commShell.addGlobalEventListener(
23         startStateEndListener);
24 }

```

CTATComponentInitialized (siguiente fragmento de código) permite comprobar si un componente ha sido inicializado antes de volver a crearlo. Primero comprueba su existencia dentro de la lista de componentes de CTAT. En caso de no existir es creado, teniendo también en cuenta si el tutor se encuentra en el modo Quiz o no.

```

1 dtCTATToolsPointer.CTATComponentInitialized = function(name
2     ,element,CTATComponentType){
3     var exist = false;
4     if(CTATShellTools!=null && CTATTutor.tutorInitialized==
5         true){
6         var compList = CTATShellTools.component_descriptions;
7         for(var key in compList){
8             if(key==name){
9                 exist = true;
10            }
11        }
12    }
13    if(!exist){
14        var CTATComponent = CTAT.ComponentRegistry[
15            CTATComponentType];
16        var ctat_component = new CTATComponent();
17        ctat_component.setName(name);
18        ctat_component.setDivWrapper(element);
19        ctat_component.processAttributes();
20        ctat_component.init();
21        ctat_component.processTabOrder();
22        ctat_component.setEnabled(ctat_component.getEnabled()
23            );
24    }

```

```

20     if (ctat_component.isFeedbackComponent() == false) {
21         var compEntry = new CTATComponentDescription();
22         compEntry.type = ctat_component.getClassName();
23         compEntry.name = ctat_component.getName();
24         compEntry.setComponentPointer(ctat_component);
25         CTATShellTools.registerComponentDescription(
                compEntry);
26     }
27     CTATShellTools.listComponents();
28 }
29 }
30 };

```

Este servicio también contiene la función `initCTATLoader` (siguiente fragmento de código) que es la función que utiliza CTAT para su inicialización. Ésta se sitúa aquí para que pueda ser utilizada desde AngularJS. Además cuenta con la introducción de algunas modificaciones en su inicio que incluyen la función `dtCTATToolsPointer`.

```

1 dtCTATToolsPointer.initCTATLoader = function(myVars) {
2     function ctatOnload() {
3         initTutor(myVars);
4         dtCTATToolsPointer.initListener();
5     }
6     function startCTAT() {
7         initTutor(myVars);
8         dtCTATToolsPointer.initListener();
9     }
10    function initOnload() { ... };
11 };

```

## 6.4.2 DIRECTIVAS

La definición de directiva puede verse en la Sección B.5. En este apartado se explotan sus características para la creación de componentes compatibles con la tecnología de CTAT. Todas ellas tienen acceso al servicio `dtCTATTools`, e independientemente de su propósito, todas comienzan con el siguiente código:

```

1 var ctatEnabled = dtCTATTools.isCTATEnabled();
2 if(!ctatEnabled){
3     return {
4         link: function() {},
5     };
6 }

```

De esta forma, cuando CTAT está inactivo, ninguna de las directivas interviene en la interfaz, resultando totalmente transparentes para el sistema.

## DIRECTIVA PARA LA CARGA DE CTAT

Esta directiva está restringida a ser un elemento de HTML y se sitúa en la página principal para que sea cargado con el resto de los elementos.

```
1 <dt-ctat-loader></dt-ctat-loader>
```

Durante la carga inicial, se lee la URL de la página web. Si la URL finaliza con /tutor/, CTAT debe ser activado en modo de autoría. Por el contrario, si la URL finaliza con /tutor/?c=1&p=1, CTAT debe ser activado en modo por defecto, siendo el problema 1 del curso 1 el que ha de ser resuelto. En el modo autoría, la herramienta CTAT Behavior Recorder se comunica con la interfaz del simulador mediante un WebSocket. Por tanto, en su configuración es necesario proporcionarle información sobre el servicio de comunicación, la URL y el puerto, de la forma que mostramos a continuación:

```
1 var myVars = {
2   tutoring_service_communication: 'websocket' ,
3   remoteSocketURL: 'http://localhost' ,
4   remoteSocketPort: '20080' ,
5 };
```

En el caso de que el modo sea por defecto, hay que proporcionar a CTAT información sobre el gráfico de comportamiento que debe cargarse, la comunicación entre el servicio de tutoría y la interfaz y los parámetros del logging para registrar los datos en un almacén de datos. En los parámetros del logging se incluyen aquellos usados para registrarse en el servidor y su dirección (línea 16), para identificar el estudio, el problema o los alumnos.

```
1 var questionFile = this.getQuestionFile(problem, course);
2 var myVars = {
3   //Parámetros del tutor
4   question_file: questionFile,
5   tutoring_service_communication: 'javascript'
6   //Parámetros para el Logging
7   auth_token:'tokeUnico', //Token necesario para la
   identificación
8   dataset_level_name1:'mriBasica',
9   ...
10  dataset_level_name9:'mriAvanzada',//el nombre que
   identifica el nivel
11  dataset_level_type1:'tema1',
12  ...
13  dataset_level_type9:'tema9',//el nombre que identifica el
   nivel
14  dataset_name:'myDataSet',//Identificación del nombre de
   los datos
15  Logging:'ClientToLogServer',//Requerido en la TutorShop
16  log_service_url:'http://pslc-qa.andrew.cmu.edu/log/server
   ',//Específico para la TutorShop
```

```

17 |   problem_name: 'problema1', //Nombre del problema
18 |   ...
19 |   school_name: 'Nombre1', //Identificación de la escuela
20 |   user_guid: '12345', //Identificación única del usuario
21 |   ...
22 | };

```

Finalmente, el código de esta directiva presentaría una apariencia similar a la que se ve en el panel a continuación. La variable '\$timeout' nos permite esperar que AngularJS haya terminado de cargarse por completo para comenzar su ejecución

```

1 | .directive('dtCtatLoader', ['$timeout', 'dtCTATTools',
   |   function($timeout, dtCTATTools){
2 |   //Codigo CTAT habilitado
3 |   ...
4 |   //Configurar CTAT de la forma explicada anteriormente
5 |   ...
6 |   dtCTATTools.setCTAT();
7 |   //Configuración
8 |   var myVars = dtCTATTools.getCTATMyVars();
9 |   function main(scope, element, attr){
10 |     $timeout(function(){
11 |       //Inicializar CTAT
12 |       dtCTATTools.initCTATLoader(myVars);
13 |     });
14 |   } //endMain
15 |   return {
16 |     restrict: "E",
17 |     link: main,
18 |   };
19 | }]);

```

## DIRECTIVA PARA LOS COMPONENTES DE CTAT

Todas las directivas utilizadas para los componentes de CTAT deben constar como atributos en el *Document Object Model* (DOM). Así pues, cuando se declare un elemento que sea tutorizable, se deberá añadir el atributo correcto al elemento, como se muestra en este fragmento de código, donde el atributo sería "data-dt-ctat-text":

```

1 | <input data-dt-ctat-text
2 |       type="text"
3 |       id="pi_name"
4 |       data-ng-model="...">

```

Actualmente existen implementadas las siguientes directivas: dtCtatSelect, dtCtatCheckbox, dtCtatListItem, dtSpinBox, dtCtatText, dtCtatButton. Todas ellas tienen en común el código que les permite funcionar junto a CTAT. Emplearemos, como ejemplo, la directiva dtCtatSelect

(siguiente panel). La directiva comienza comprobando si CTAT está habilitado; si esto sucede, se declarará un objeto `dtDirectiveSelect` —en este caso— que podrá ser accedido incluso desde dentro de las funciones que ejecuta CTAT (línea 5). Posteriormente, se crea el tipo de componente, que se llamará `dtCTATSelect` (línea 6); éste hereda del componente “Clickable” de CTAT si bien hereda otros componentes del componente “Tutorable”. Dentro de este componente se define su SAI (recordemos que uno de los eventos enviados entre los componentes y el sistema de tutoría era el evento SAI, según se describió en la página 95). En este caso, al tratarse de un elemento seleccionable, debe ser capaz de enviar el elemento que ha sido seleccionado (si, por ejemplo, se tratase de un “input”, éste debería enviar el valor del “input”, además de su identificación). Antes de comenzar la función `main()` (línea 29) el tipo componente ya habría sido declarado dentro de los componentes de CTAT. El código que se encuentra dentro de la función `main()` se ejecuta una vez por cada directiva declarada en el DOM, esto también incluiría a aquellas directivas que se generan dinámicamente. En cada una de estas ejecuciones, iría añadiendo una nueva propiedad al objeto `dtDirectiveSelect` (líneas 38 y 39) que dependería de su identificación única. De esta forma podemos crear todos los nuevos componentes de este tipo que sean necesarios y CTAT sería capaz de registrarlos, independientemente del momento en el que sean creados.

```
1  .directive('dtCtatSelect', ['dtCTATTools', function(  
    dtCTATTools){  
2      //CTAT habilitado  
3      ...  
4      //Hasta la función main(), este código se ejecuta ú  
        nicamente una vez en la carga de AngularJS  
5      var dtDirectiveSelect = {}; //Esta es la pieza clave para  
        que el código funcione. Esta variable ha sido creada  
        en la función que hace de constructor del objeto  
        dtCTATSelect y por tanto, puede ser accedida por este  
        objeto.  
6      var dtCTATSelect = function(){  
7          //Esta línea permite heredar del componente 'Clickable'  
            de CTAT  
8          CTAT.Component.Base.Clickable.call(this, "dtCTATSelect"  
            , "dtCTATSelect");  
9          //A continuación se realiza la configuración del  
            componente  
10         this.setEnabled=function setEnabled(aValue){  
11             return 0;  
12         };  
13         //Este código se ejecuta en la creación del componente  
14         this.init=function init(){  
15             //Primero se consigue la id del componente  
16             var _id = this.getName();  
17             //Se establece el valor seleccionado, como se observa  
                siempre en función de la id
```

```

18     var select = dtDirectiveSelect[_id].returnSelect();
19     this.setComponent(select);
20     this.newChoice = dtDirectiveSelect[_id].newChoice;
21     this.setInitialized(true);
22     this.addComponentReference(this, select);
23   };
24 };
25 dtCTATSelect.prototype = Object.create(CTAT.Component.
    Base.Clickable.prototype);
26 dtCTATSelect.prototype.constructor = dtCTATSelect;
27 CTAT.ComponentRegistry.addComponentType('dtCTATSelect',
    dtCTATSelect);
28 //función main()
29 function main(scope, element, attr, ctrl){
30   //Función que se ejecuta una única vez en la carga pero
    para cada componente que tenga este atributo
31   window.setTimeout(onLoad(scope, element, attr, ctrl),
    0);
32   function onLoad(scope, element, attr, ctrl){
33     //Comprueba si el objeto tiene la propiedad con el
    nombre de la id
34     if(!(String(attr.id) in dtDirectiveSelect)){
35       Object.defineProperty(dtDirectiveSelect, attr.id, {
    value: {}});
36     }
37     //define la función
38     dtDirectiveSelect[attr.id].returnSelect = function()
    {...};
39     dtDirectiveSelect[attr.id].newChoice = function()
    {...};
40     ...
41   }//endOnLoad
42 }//endMain
43 return {
44   restrict: "A",
45   require: "ngModel",
46   link: main,
47 };
48 }])

```

Otro de los requisitos que estos componentes deben cumplir es que debe ser posible elegir si el componente va a ser registrado inmediatamente por el servicio de tutoría o si va a ser enviado junto a otros. En el primer caso, se crearían de forma ordinaria, mientras que en el segundo caso, sería necesario declararlos como `noTutorComponent`; esto segundo es lo que mostramos en el siguiente panel de código (primera línea). Por otro lado, también debe ser posible elegir qué

elemento debe ser el encargado de enviar la información de estos componentes al servicio de tutoría. En el caso que nos atañe, éstos son enviados cuando se pulse un botón; el código para su definición en HTML se encuentra a continuación.

```

1 <input data-dt-ctat-text="onWatch,noTutorComponent"
2     type="text"
3     id="at_freqDir"
4     data-ng-model="..."
5     readonly>
6 <button id="scan_b2"
7     data-dt-ctat-button=""
8     data-dt-ctat-on-correct="data.scanInfo.
9         requestScan()"
10    data-dt-ctat-target="Seq"
11    data-ng-click="!ctatEnabled && data.scanInfo.
12        requestScan()">

```

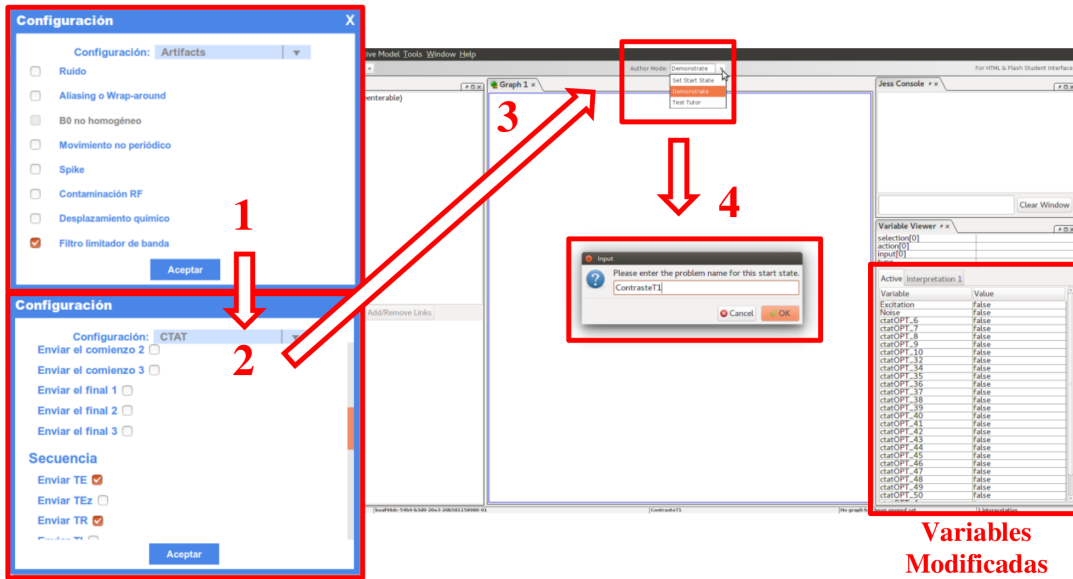
En este código se ven dos atributos, (a) `data-dt-ctat-on-correct` (línea 8) que es la función que se ejecutaría si todos los elementos enviados son correctos y (b) `data-dt-ctat-target` (línea 9) que determina el grupo de elementos que van a ser enviados. En este caso, el grupo se identifica como “Seq”, que se definiría en el panel de registro de componentes visto en la Sección 5.3.2.

Finalmente, la forma que tiene la directiva de saber si la información de los componentes registrados ha sido correcta, es mediante el uso de la función `initListener`, descrita en el servicio (Sección 6.4.1), y que almacenaría todas las respuestas a los componentes enviados y así poder comprobar si han sido correctos o no.

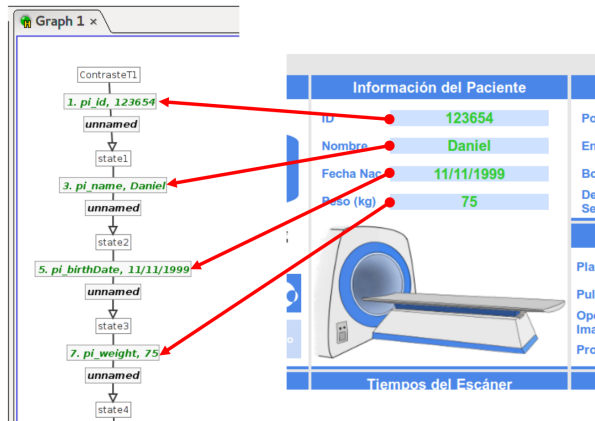
## 6.5 RESULTADOS

Mostramos unos resultados mediante el ejemplo de uso del ITS en nuestro simulador, tanto para la creación como para la resolución de un problema.

El modo de creación o “Authoring” se muestra en la Figura 6.6, donde se describe un ejemplo de creación de un problema para nuestro simulador. Primero, se debe establecer el estado inicial (Figura 6.6a); con este fin, debemos determinar los artefactos que afectarán a nuestra adquisición (6.6a1) así como los componentes que van a ser registrados (6.6a2). Supongamos que este ejemplo está relacionado con obtener algún tipo de contraste en MR; en consecuencia, los parámetros a registrar deben ser TE y TR (ver las marcas en la figura). La creación del BRD es el siguiente paso a seguir (ver Figura 6.6a3). Para este fin, cambiamos a “Demostrar” y luego se nos pide que establezcamos un nombre para el problema con el estado inicial recién definido (6.6a4)); para este ejemplo, usaremos el nombre *ContrasteT1*. Las variables que se han modificado en el estado inicial ahora se muestran como variables registradas en el lado derecho de la figura. El BRD se crea ejecutando secuencialmente pasos en el simulador (Figura 6.6b). Conforme éstos se ejecutan, el registrador de comportamiento CTAT genera el gráfico; en este caso, los pasos consisten en introducir datos personales del paciente.



(A) Creación del estado inicial.



(B) Creación gráfico de comportamiento.

FIGURA 6.6: Creación de un problema con CTAT. (A) muestra los pasos necesarios para comenzar la creación del problema, donde tanto la CTAT Behavior Recorder como nuestra interfaz deben estar ejecutándose. Una vez creado el estado inicial, será posible la creación del gráfico de comportamiento como se muestra en (B).

El modo resolución o por defecto se ilustra en la Figura 6.7, donde aparece la GUI asociada al simulador cuando el ITS está activo. En primero lugar, para poder acceder a los ejercicios que utilicen el ITS, debe seleccionarse la etiqueta “Exp. Tutor” de la barra lateral (Figura 6.7a). En este momento, la aplicación mostrará el panel para la selección del problema (Figura 6.7b). El estudiante usa este panel para seleccionar la temática de interés entre los disponibles y el ejercicio preciso de esa temática. Después de presionar el botón Comenzar, se muestra un nuevo panel, en el que se muestra tanto el enunciado del problema como el soporte gráfico adicional (consulte la Figura 6.7c). Esta figura también muestra el panel ITS (cuadro de diálogo con el



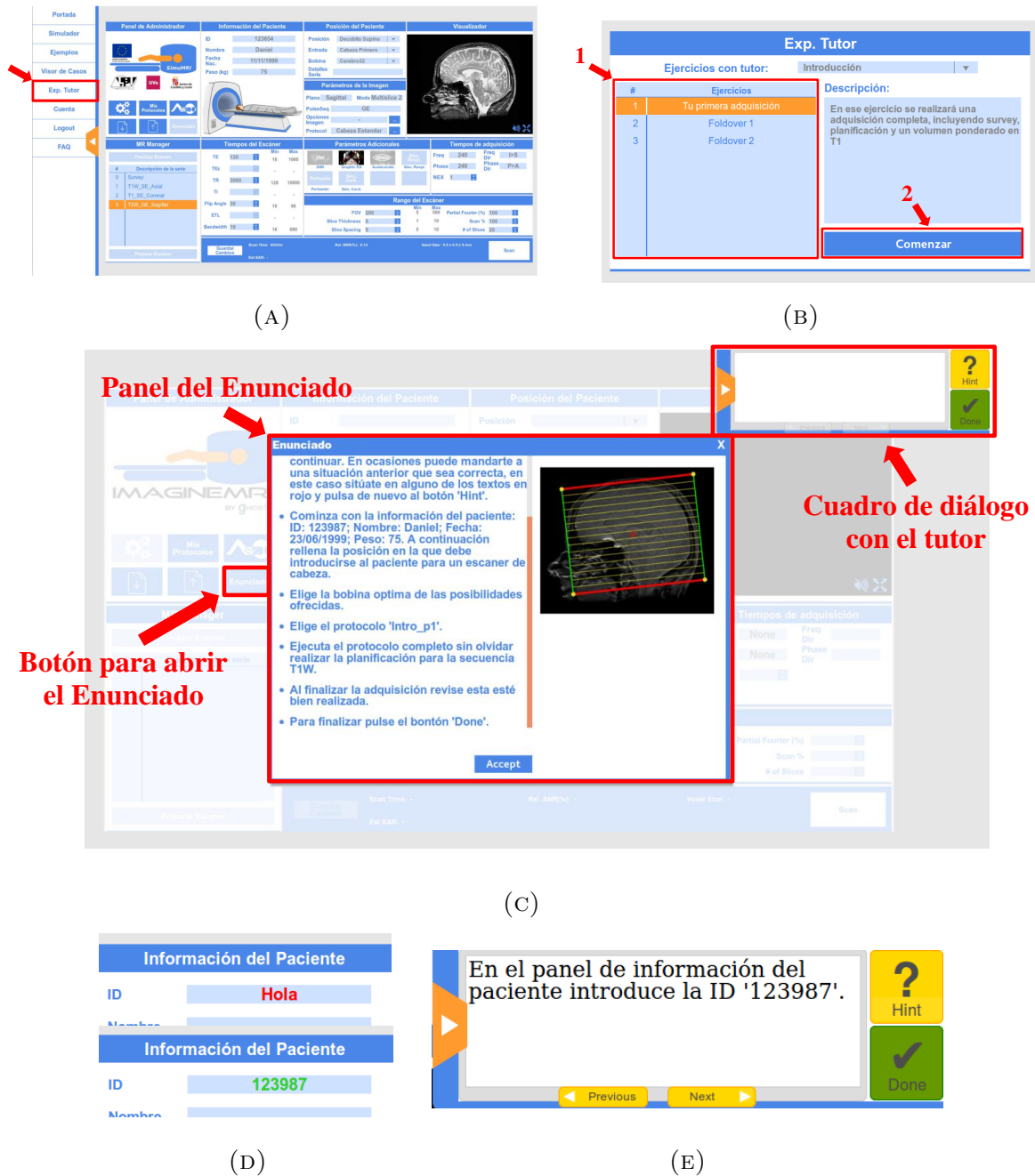


FIGURA 6.7: Realización del problema. Los pasos (A) y (B) son necesarios para acceder al problema que se desea realizar. La imagen (C) muestra el estado inicial del problema y las diferentes partes con las que cuenta el usuario para interactuar con el ITS. El ejemplo de un comentario dado por el ITS puede verse en (D) y (E).

Tutor), el cuál es utilizado por el ITS para proporcionar mensajes al alumno a medida que se ejecuta el problema.

La figura Figura 6.7d muestra una entrada incorrecta al campo particular (en este caso, ID), así como la respuesta correcta. El primero está escrito en rojo mientras que el segundo se usa color verde; se podría proporcionar un tercer indicador en amarillo (que no está mostrado en la figura) de forma que señalase la siguiente zona en la GUI donde trabajar. La Figura 6.7e muestra los comentarios de texto proporcionados por el ITS en el cuadro de diálogo del Tutor. En el caso en el que la situación errónea de la Figura 6.7d hiciese que el estudiante no supiera cómo continuar, podría pedirle una ayuda al tutor. Para ello pulsaría en el botón amarillo del cuadro de diálogo; como resultado, el tutor en este caso le indicara la respuesta correcta (que es lo mostrado en la figura). En otros casos, las pistas del tutor podrían ser menos específicas.

## 6.6 CONCLUSIONES

---

Gracias a la incorporación de CTAT a nuestro simulador, somos capaces de crear diversos ejercicios con la herramienta CTAT Behavior Recorder, para lo cual establecemos el estado inicial del problema, creamos el árbol y, a continuación, los diversos tests de funcionamiento. Un ejemplo de cómo realizarlo puede encontrarse en el Apéndice F que contiene el correspondiente manual. Una vez creados, los problemas pueden ser resueltos por los estudiantes mediante el uso habitual del simulador, con la diferencia de que ahora recibirán la correspondiente tutorización según su grado de necesidad.

# EXPERIMENTACIÓN

---

**E**N este capítulo se describe el trabajo experimental llevado a cabo sobre el simulador de resonancia desarrollado, tanto para evaluar la experiencia de los usuarios como para caracterizar su utilidad como herramienta educativa. Tras justificar la necesidad de evaluar una herramienta software como esta, se enumeran las diferentes versiones que se han ido desarrollando del simulador. La existencia de estas versiones sucesivas es una razón adicional que explica la necesidad de realizar también evaluaciones sucesivas. A continuación, se detalla el diseño experimental de las diferentes evaluaciones y se describen y analizan los resultados obtenidos.

## 7.1 INTRODUCCIÓN

---

El desarrollo de software es en general un proceso iterativo, donde el programa evoluciona a medida que avanzan sus versiones. Es habitual que dicho avance se centre únicamente en la adición de características y funcionalidades al programa, sin cuestionar si realmente se está siguiendo el camino más adecuado. De hecho, común es el caso donde se considera que añadir una nueva funcionalidad implica mejorar el software. Desafortunadamente esto no tiene por qué ser así, hasta el punto que es posible que la nueva funcionalidad perjudique ciertos aspectos del programa (Sauro and Lewis, 2016). En consecuencia, debe de contarse con una manera de comprobar si los cambios han tenido un efecto positivo o negativo sobre el programa. Para responder a este tipo de preguntas existen múltiples herramientas y procesos (Albert and Tullis, 2013) que evalúan la calidad del software.

La *experiencia del usuario* o *User eXperience* (UX) puede ser una forma de comprobar la calidad del software. Se centra fundamentalmente en los errores, la capacidad de aprendizaje y la memorización que requiera el software (Nielsen, 1994). Por otro lado, también es importante que el diseño de la interfaz sea sencillo de entender y manejar. Los sistemas informáticos relacionados con el ámbito médico, como es el caso que nos ocupa, no deben ser una excepción (Ash et al., 2004). Finalmente, la usabilidad es otro aspecto a tener en cuenta. Ésta se define por la *Organización Internacional de Normalización* o *International Standards Organization* (ISO) (Schoeffel et al., 2003) como aquella característica que representa la efectividad, la eficiencia y la satisfacción que siente un usuario en el desempeño de un conjunto de tareas en un entorno concreto.

Dado este contexto, durante su desarrollo, el simulador de resonancia ha pasado por diversas etapas que han finalizado con distintas versiones del mismo. En cada versión, siguiendo las indicaciones expuestas, se ha llevado a cabo una evaluación de la misma. Sin embargo, la valoración de cada versión ha venido dada por el objetivo de dicha versión. Este asunto es tratado más en detalle en la Sección 7.2.

Generalizando, la evaluación se ha tratado desde dos perspectivas principales: la UX, en primer lugar, y la utilidad de la herramienta en un contexto educativo, en segundo lugar. La percepción que un estudiante tiene del simulador como usuario nos permite analizar la usabilidad y la utilidad percibida. Gracias a ello, es posible hacer una valoración de la evolución del simulador a lo largo del tiempo (Sección 7.3). Por otro lado, comprobar la utilidad real del simulador como herramienta educativa es importante para el cumplimiento y la verificación del tercer objetivo visto en la Sección 1.2. Los detalles sobre esta prueba los podemos ver en la Sección 7.4.

Finalmente, a la vista del estudio realizado en el Capítulo 3, y hasta donde sabemos, este tipo de evaluación en un simulador de MR orientado al mundo educativo no se había realizado con anterioridad.

---

## 7.2 EVOLUCIÓN DEL SIMULADOR

---

Durante el desarrollo de la presente tesis, el sistema desarrollado ha ido evolucionando, y junto con ello, sus versiones. La Figura 7.1 muestra una representación espiral del desarrollo del simulador, describiendo además el ciclo del software. En el interior de esta espiral, además, pueden distinguirse tres colores, cada uno de los cuales muestra la evolución de las principales partes que conforman el simulador, a saber, el *Núcleo de la Simulación*, la *Interfaz* y la posterior incorporación del *ITS*. Por un lado, el *Núcleo de Simulación* ha evolucionando internamente para mejorar su modularidad, su rapidez y sus características. Pero, a excepción de la rapidez, el resto de estos atributos son totalmente transparentes para el usuario, por lo que su evaluación ha sido únicamente interna. En éste núcleo se han comprobado que se cumplen los conceptos teóricos, comparando los modelos obtenidos con los prototipos realizados en Matlab. Por el otro lado, la *Interfaz* y el *ITS* sí son percibidos en gran medida por el usuario y por ello, en estos casos, la evaluación por usuarios reales es verdaderamente relevante. En las siguientes secciones se analizan, con detalle sobre las cuatro versiones, las características de cada una y el proceso de evaluación que fue utilizado.

---

### 7.2.1 VERSIÓN 1

---

La versión 1 se corresponde con el primer prototipo que se desarrolló, donde la *Interfaz*, aunque similar en apariencia a la actual, estaba escrita en puro JavaScript y tenía una extensión total de unas 2000 líneas. Mientras tanto, en su primera implementación, el servidor ya utilizaba la framework Django, aunque no hacía uso de ningún sistema para el registro de usuarios. Además, el *Núcleo de la Simulación* estaba escrito en Python, por dos razones. Primera, Python es el lenguaje en el que está programado Django, y por tanto es totalmente compatible. En segundo lugar, Python incorpora un gran número de herramientas que en cierta medida lo

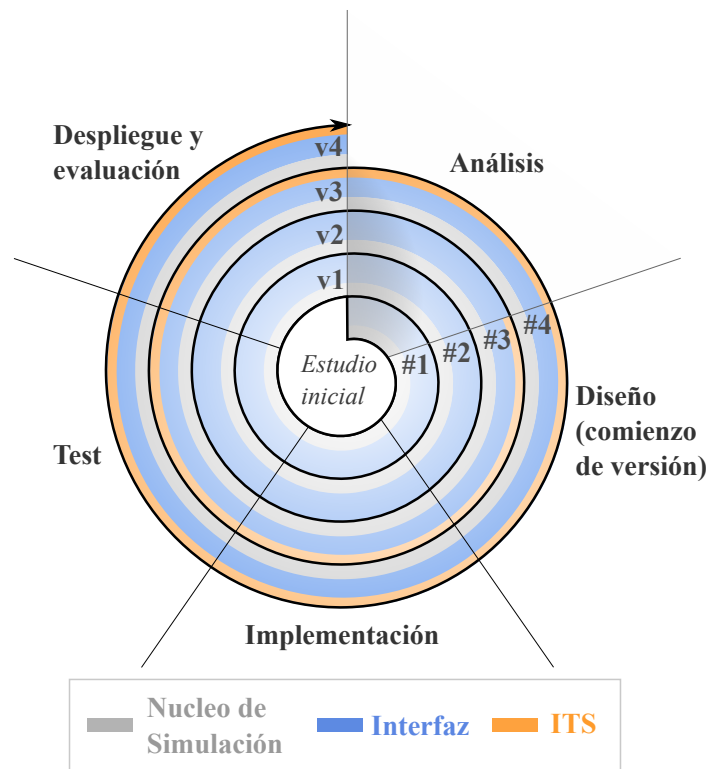


FIGURA 7.1: Evolución y versiones del simulador de resonancia. Para su detalle se utiliza un modelo incremental representado como una espiral donde se muestran las diferentes fases del desarrollo del software. Como puede verse, la espiral abarca el desarrollo de 4 versiones, donde el núcleo de simulación y la interfaz están presentes desde el inicio, y donde el ITS pasa a incluirse a partir de la tercera versión junto a la Interfaz.

hacen muy similar a Matlab, lo que facilita mucho la programación en nuestro caso. En esta versión, el objetivo principal era comprobar la viabilidad de crear un simulador de resonancia magnética como un sistema web, confirmar nuestra capacidad para simular la adquisición de imágenes con un formato médico y comprobar que éstas podían ser visualizadas correctamente en el navegador.

Para su evaluación, se concertó una reunión con personal experto perteneciente a la SERAM; en ella, se realizó una presentación en la que se describió el prototipo y se mostraron sus principales características. Posteriormente, se realizó una discusión guiada sobre las características del simulador, analizando sus fortalezas y sus limitaciones. Como resultado, la viabilidad inicial del proyecto quedó demostrada, y se comprobó una buena acogida por parte de los expertos. Además, se identificaron dos limitaciones principales. La primera limitación consistía en los tiempos de simulación, que eran del orden de 40 segundos por cada adquisición simulada. Este tiempo se consideró excesivo para permitir un manejo ágil de la herramienta, de forma que se instó a intentar reducirlo. En relación con esto, se comprobó que gran parte del tiempo empleado en la simulación se invertía en realidad en la carga de los volúmenes tridimensionales y en la realización de iteraciones a lo largo de estos volúmenes. La segunda limitación se encontraba en el abanico, reducido en ese momento, de funcionalidades disponibles en la *Interfaz* de esa

versión. En este caso, la forma de desarrollar el código imponía ciertas limitaciones. Si bien el diseño era funcional, cada cambio de cierta envergadura requería la modificación de un elevado número de elementos.

---

## 7.2.2 VERSIÓN 2

---

El objetivo principal de la segunda versión se centraba en solventar las limitaciones detectadas en la primera versión. La transición fue lenta debido al rediseño y reimplementación del programa por completo, donde el diseño resultante se corresponde con el visto en la Sección 5.3. Éste ya cuenta con la incorporación de AngularJS, que añade manejabilidad y fluidez a la *Interfaz*. Por otro lado, el sistema incorpora también un sistema de registro para los usuarios con distintos roles, como el de profesor y el de alumno. Además, el *Núcleo de la Simulación* fue reescrito en C++ haciendo uso de la librería ITK. Asimismo, la comunicación entre el servidor en Django y el *Núcleo de la Simulación* tuvo que implementarse, para lo cual se creó un *wrapper* entre Python y C++.

El objetivo de la evaluación en esta versión consistía en comprobar cómo funcionaba este software con el usuario final, e identificar el máximo número de errores tanto en el diseño como en su implementación. Para ello, el simulador fue utilizado como herramienta de apoyo en un curso online organizado por la SERAM. La versión fue finalizada antes de la celebración del curso, que se celebró entre mayo y junio de 2018 y que estaba orientado al aprendizaje básico sobre MRI. En él se identificaron algunos fallos en el diseño; éstos tenían que ver con la información que proporcionaba el simulador sobre su estado. Por otro lado, también se detectaron algunos errores en el funcionamiento. En su evaluación también se aplicó un test, como veremos más adelante.

---

## 7.2.3 VERSIÓN 3

---

Esta tercera versión avanza en funcionalidades, las cuales fueron propuestas por el personal experto. De la misma forma, trata de solventar los errores detectados en el diseño y los fallos en el funcionamiento. Además, en ella se introduce el sistema de tutoría. En esta versión, el objetivo atiende a tres necesidades: primero, evaluar si las mejoras introducidas repercuten positivamente sobre el software; segundo, tratar de responder a la pregunta de si el simulador resulta útil como una herramienta educativa de forma objetiva y, tercero, comprobar el funcionamiento del ITS.

En este caso, la evaluación se llevó a cabo *in situ* en una actividad presencial con 64 alumnos. La prueba se realizó en la Escuela de Radiógrafos del Hospital Clínico San Carlos, Madrid, en Octubre de 2018. El conjunto de estos alumnos participó en un experimento educativo encaminado a comprobar la utilidad de la herramienta durante la instrucción en MR. Por otro lado, se aplicó también el test empleado en la evaluación de la versión 2 del simulador al grupo de alumnos que había empleado el simulador de resonancia magnética durante el experimento. Finalmente, el sistema de tutoría fue puesto a prueba internamente para identificar fallos en el diseño. En este caso se identificó la necesidad de mejoras tanto en la creación de ejercicios para el alumno como en su ejecución por parte de éstos.

---

## 7.2.4 VERSIÓN 4

---

En su cuarta versión, el simulador ya ha demostrado ser un sistema robusto que, aunque sigue contando con algunas limitaciones, ya ha sido utilizado con éxito en dos pruebas. En esta última versión se añaden nuevas funcionalidades, como la creación de protocolos y de secuencias personalizadas por parte de los alumnos, o la adición de nuevas secuencias de pulsos y de artefactos a la simulación. Por otro lado, tratan de solucionarse los problemas identificados en el ITS. De las nuevas funcionalidades añadidas en este punto, la que implica un cambio de mayor alcance sobre el simulador es la adición del ITS. Por tanto, en esta versión, el objetivo principal en la evaluación es comprobar cómo repercute el ITS en el simulador.

En su evaluación, el simulador fue de nuevo utilizado en un curso *online* organizado por la SERAM que se corresponde con la reedición del curso del que se habló en la versión 2. En él se aplica el test que ya se aplicó en las versiones segunda y tercera, con la salvedad de que ahora se hace referencia explícita al conjunto de simulador—ITS.

---

## 7.3 ESTUDIO SOBRE LA EXPERIENCIA DEL USUARIO

---

La UX es un concepto de gran relevancia en la comunidad de la *interacción persona-ordenador o Human-Computer Interaction* (HCI) y, durante años, tanto su definición como sus características han sido definidas según la interpretación de cada autor (Law et al., 2008). Con el fin de aunar las opiniones, en 2008 la ISO propuso un borrador en el que se discutía la definición de la UX. La versión final fue publicada en la ISO 9241-210:2010 (DIS, 2009). En el capítulo 2, sección 2.15, se define la UX como “las percepciones y las respuestas de una persona resultantes del uso y/o uso anticipado de un producto, sistema o servicio”. Se proporcionan, además, tres notas aclaratorias sobre este concepto:

- Nota 1: la experiencia del usuario incluye todas las emociones, creencias, preferencias, percepciones, respuestas físicas y psicológicas, comportamientos y logros de los usuarios que se producen antes, durante y después del uso.
- Nota 2: la experiencia del usuario es una consecuencia de la imagen de la marca, la presentación, la funcionalidad, el rendimiento del sistema, el comportamiento interactivo y las capacidades de asistencia del sistema interactivo, el estado interno y físico del usuario como resultado de experiencias anteriores, actitudes, habilidades y personalidad, y contexto de uso.
- Nota 3: La usabilidad, cuando se interpreta desde la perspectiva de los objetivos personales de los usuarios, puede incluir el tipo de aspectos perceptivos y emocionales típicamente asociados con la experiencia del usuario. Se pueden utilizar criterios de usabilidad para evaluar aspectos de la experiencia del usuario.

Atendiendo a esta definición, y siguiendo el procedimiento que veremos en esta sección, se evaluará la UX en el simulador de resonancia. Para su valoración, se usará una combinación

CONTENIDOS	<b>PROGRAMA DEL CURSO</b>
<p><b>Bloque 1:</b> Principios de la imagen con resonancia magnética.</p> <ul style="list-style-type: none"> <li>• Introducción a la resonancia magnética</li> <li>• Fundamentos físicos de la resonancia magnética</li> <li>• El pulso de radiofrecuencia y relajación</li> <li>• Formación de imagen y el espacio K</li> <li>• Contraste en la imagen de resonancia</li> <li>• Secuencias básicas en resonancia magnética</li> <li>• Secuencias habituales</li> <li>• Secuencias y métodos de imagen avanzados</li> <li>• Calidad de imagen y artefactos en resonancia magnética</li> </ul> <p><b>Bloque 2:</b> Adquisición de resonancia magnética.</p> <ul style="list-style-type: none"> <li>• El escáner de resonancia magnética</li> <li>• Emisores-receptores de resonancia magnética</li> <li>• Seguridad en resonancia magnética</li> <li>• Posicionamiento del paciente</li> <li>• La consola de mandos de la resonancia. Cómo planificar y ejecutar una resonancia</li> <li>• Planificación geométrica</li> <li>• Parámetros del estudio</li> <li>• Factores de calidad de la imagen</li> <li>• Artefactos de imagen</li> </ul>	<p><b>Bloque 1</b> Principios de la imagen con resonancia magnética</p> <ul style="list-style-type: none"> <li>_ Introducción a la resonancia magnética</li> <li>_ Fundamentos físicos de la resonancia magnética</li> <li>_ El pulso de radiofrecuencia y relajación</li> <li>_ Formación de imagen y el espacio</li> <li>_ Contraste en la imagen de resonancia</li> <li>_ Secuencias básicas en resonancia magnética</li> <li>_ Secuencias habituales</li> <li>_ Secuencias y métodos de imagen avanzados</li> <li>_ Calidad de imagen y artefactos en resonancia magnética</li> </ul> <p><b>Bloque 2</b> Adquisición resonancia magnética</p> <ul style="list-style-type: none"> <li>_ El escáner de resonancia magnética</li> <li>_ Emisores-receptores de resonancia magnética</li> <li>_ Seguridad en resonancia magnética</li> <li>_ Posicionamiento del paciente</li> <li>_ La consola de mandos de la resonancia. Cómo planificar y ejecutar una resonancia</li> <li>_ Planificación geométrica</li> <li>_ Parámetros del estudio</li> <li>_ Factores de calidad de la imagen</li> <li>_ Artefactos de imagen</li> </ul>
(A) Curso 2018.	(B) Curso 2019.

FIGURA 7.2: Programa contenido en los trípticos de los cursos on-line impartidos por la SERAM.

de puntuaciones basadas en la percepción del usuario, en el índice de satisfacción y en una evaluación de la usabilidad mediante un cuestionario estándar explicado en la metodología. Finalmente, se expondrán los resultados y se discutirán.

### 7.3.1 METODOLOGÍA

El simulador de resonancia imita, en gran medida, a un sistema real de MRI, lo cual ha marcado fuertemente el diseño de la interfaz así como los componentes que la conforman. Estos factores, como se ha puesto de manifiesto en este capítulo, influyen en la UX y la limitan. Sin embargo, es interesante comprobar la valoración de los usuarios sobre el programa y ver así si



los cambios realizados suponen un enriquecimiento del mismo.

En esta sección se realiza un estudio longitudinal de la UX en el simulador, donde, a través de diferentes experiencias y sus respectivas versiones, se evalúa la UX y se analizan los resultados.

## TRABAJO EXPERIMENTAL

El simulador de MR ha sido empleado en tres experiencias detalladas a continuación. Durante éstas, los participantes han sido diversos, y el desarrollo del simulador ha tenido distintas fases (Sección 7.2).

1. **Curso SERAM 2018.** Este curso dio comienzo el 30 de abril de 2018 y tenía una duración de 8 semanas, con una dedicación estimada por parte de los participantes de entre 5 y 6 horas a la semana. El programa del curso puede verse en la Figura 7.2a. Al comienzo del curso se realizó una encuesta inicial, en la que se comprobó que el 72,1 % de los inscritos eran médicos radiólogos, el 21,2 % eran técnicos superiores en imagen para el diagnóstico y el resto se repartían entre ingenieros, médicos residentes y otras opciones. Los alumnos fueron inicialmente preguntados sobre su grado percibido de manejo de las herramientas informáticas, a lo cual el 51 % respondió que era básico, el 42,3 % respondió que era intermedio. Sólo un 6,7 % lo consideró como avanzado. El curso estaba formado fundamentalmente por vídeo-presentaciones, y el simulador se utilizaba como material de apoyo. En este curso participaron un total de 179 estudiantes.
2. **Presentación oral sobre MR.** Esta experiencia se llevó a cabo en un único día, el 29 de octubre de 2018, en el Hospital Clínico San Carlos, Madrid, con 32 alumnos de la escuela de radiógrafos perteneciente a esta institución. Los alumnos asistentes pertenecían a los ciclos de Técnico de Diagnóstico por Imagen y Medicina Nuclear y de Técnico de Radioterapia y Dosimetría. En esta experiencia se realizó una presentación oral dividida en dos partes. La primera parte fue una presentación de 90 minutos en la que se expusieron los principios básicos de MR. En la segunda parte se realizaron unas prácticas guiadas con el uso del simulador de MR.
3. **Curso SERAM 2019.** Este curso es equivalente al realizado en 2018 y dio comienzo el 8 de mayo de 2019. Su duración fue de 8 semanas, pero en esta ocasión las actividades iban a ser más demandantes y se estimó una dedicación mayor, de entre 6 y 7 horas semanales. El programa del curso puede consultarse en la Figura 7.2b. De la misma forma, a través de un cuestionario inicial se averiguó que el 46 % de los participantes eran médicos radiólogos, mientras que el 45,5 % eran técnicos superior en imagen para el diagnóstico. Al preguntar por su manejo informático, el 41,9 % consideró que era básico, el 50,6 % lo consideró intermedio y sólo un 8,5 % lo consideró avanzado. El total de participantes en este curso fue de 176. En este segundo curso, el simulador era utilizado como herramienta de apoyo. Adicionalmente, durante la realización de algunos ejercicios se habilitó el simulador con el ITS como ayuda complementaria, siendo totalmente optativa su utilización para el desempeño del curso.

## INSTRUMENTOS DE MEDIDA

El instrumento de medida utilizado en los tres experimentos para la comprobación de los diferentes aspectos de la UX está formado por cuatro elementos:

TABLA 7.1: Cuestionario usado en la *escala de usabilidad del sistema o System Usability Scale (SUS)*.

#	Enunciado
1	Me gustaría utilizar el simulador de resonancia de manera frecuente
2	Encuentro el simulador de resonancia innecesariamente complejo
3	Pienso que el simulador de resonancia es fácil de usar
4	Creo que necesitaría soporte técnico para utilizar el simulador de resonancia
5	Encuentro las diversas funciones del simulador de resonancia bien integradas
6	He encontrado demasiada inconsistencia en el simulador de resonancia
7	Creo que la mayoría de usuarios aprendería a utilizar el simulador de resonancia rápidamente
8	He encontrado el simulador de resonancia bastante incómodo y pesado de usar
9	Me he sentido cómodo utilizando el simulador de resonancia
10	Necesité aprender un montón de cosas antes de estar en condiciones de utilizar el simulador de resonancia

- **SUS.** La SUS (*System Usability Scale*) es un cuestionario estándar que se describe a sí mismo como “rápido y sucio” (Brooke, 1995). El comienzo de su desarrollo fue a principios de 1980 y hoy en día se ha convertido en uno de los cuestionarios más populares para la evaluación de la usabilidad (Brooke, 2013). De acuerdo con Brooke (1995), el test debe ser realizado por los participantes después de haber utilizado el software. Este test está formado por 10 preguntas, cada una evaluada en una escala de 5 pasos que van de “Totalmente en desacuerdo” a “Totalmente de acuerdo”. De las 10 preguntas, los elementos impares tienen un tono positivo, mientras que los pares usan un tono negativo. La puntuación obtenida varía entre 0 y 100. Además, en la SUS, tras la realización de un estudio retrospectivo, se identificaron dos subescalas (Lewis and Sauro, 2009). La primera escala está compuesta por 8 de sus elementos y se centra en la usabilidad, mientras que la segunda la formarían los dos restantes (las preguntas 4 y 10) y se centra en la facilidad para el aprendizaje (del inglés *learnability*). Ambas subescalas pueden ser evaluadas también con puntuaciones entre 0 y 100 puntos. Las preguntas que conforman la SUS pueden consultarse en la Tabla 7.1.
- **Probabilidad de recomendar o Likelihood to Recommend (LTR).** Los estudiantes también fueron preguntados por la LTR. Este test está compuesto por una única pregunta que enuncia: “¿Recomendaría el simulador de resonancia a otras personas interesadas en aprender sobre principios y adquisición de resonancia magnética?”. La escala de la respuesta se forma por los mismos 5 elementos que formaban la SUS.
- **Utilidad percibida.** El cuestionario de utilidad percibida consta de 2 elementos enunciados a continuación: Q1) “¿El uso del simulador de resonancia me ha resultado de utilidad

para aprender los conceptos sobre el funcionamiento de la resonancia magnética?”. Q2) “¿El uso del simulador de resonancia me ha resultado de utilidad para aprender los conceptos sobre adquisición de resonancia magnética?”. La escala de la respuesta estaba formada de igual forma que en la SUS.

- **Comentario libre.** Finalmente, para completar la opinión sobre la UX, se animaba al usuario a realizar un comentario libre. El enunciado era el siguiente: “Escriba aquí cualquier comentario y/o sugerencia que desee realizar sobre el simulador de resonancia y su experiencia con él.”

En el caso del segundo curso de la SERAM, las preguntas pasaron a hacer referencia explícita al conjunto del simulador con el ITS en lugar de únicamente al simulador de MR. Para todos los casos, los participantes fueron informados de que la respuesta a estos cuestionarios era opcional y su no cumplimiento no influía en ningún aspecto que pudiese afectar al alumno. Además, la información recogida fue anónima, salvo en el caso de la prueba presencial, que fue anonimizada posteriormente. Finalmente, cada uno de los participantes dio su consentimiento para usar esta información con propósitos investigadores.

### 7.3.2 RESULTADOS

TABLA 7.2: Resultados de la SUS sobre 100 y de la LTR sobre 5. En todos los casos se expresa su media y la desviación estándar (SD).

		Curso SERAM 2018 (Media±SD)	Curso presencial (Media±SD)	Curso SERAM 2019 (Media±SD)
SUS	Global	55.86±20.56	63.02±13.39	68.33±20.87
	Usabilidad	59.78±22.42	67.03±14.09	68.75±22.9
	Aprendibilidad	40.19±25.63	46.98±27.27	66.67±29.62
LTR		3.65±1.23	4.45±0.63	4.15±1.17

En el primer curso de la SERAM contestaron al cuestionario propuesto 116 personas de un total de 179 participantes. En el experimento presencial fueron 29 de 32. En el segundo curso de la SERAM fueron únicamente 27 de 176. Este último caso es excepcional, ya que como se ha indicado en la Sección 7.2 se trataba de una actividad adicional al curso y cuya participación no era obligatoria. Sin embargo, en todos los casos el número de respuestas es superior al aceptado comúnmente para la SUS, cuyo resultado se considera estable a partir de 12 participantes (Tullis and Stetson, 2004).

Los resultados obtenidos en los test de la SUS y de la LTR se recogen en la Tabla 7.2. La utilidad percibida se representa en la Figura 7.3. Finalmente, los comentarios han sido recogidos en la Tabla 7.3.

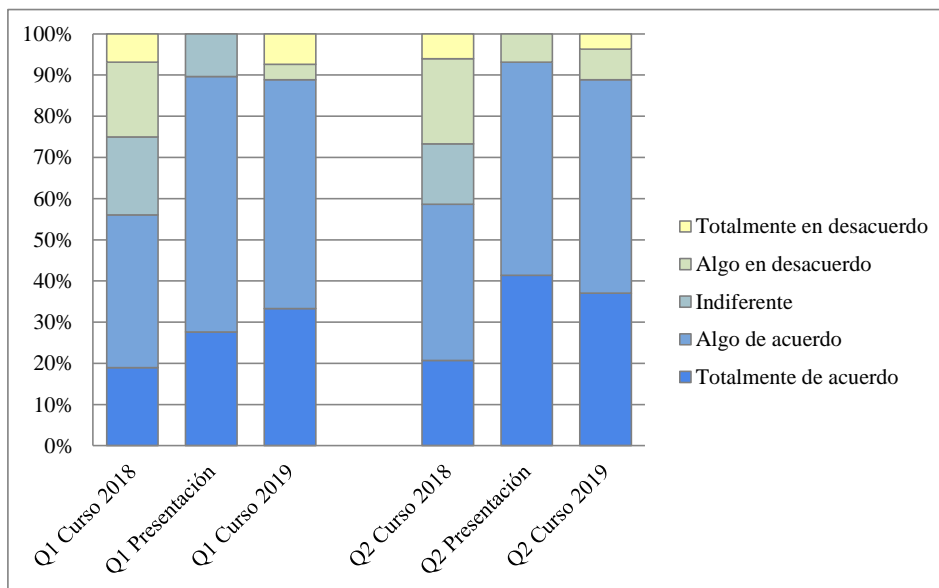


FIGURA 7.3: Resultados de la utilidad percibida. Q1 se corresponde con la pregunta: ¿El uso del simulador de resonancia me ha resultado de utilidad para aprender los conceptos sobre el funcionamiento de la resonancia magnética?. Q2 se corresponde con la pregunta: ¿El uso del simulador de resonancia me ha resultado de utilidad para aprender los conceptos sobre adquisición de resonancia magnética?

### 7.3.3 DISCUSIÓN

De los resultados expuestos se pueden extraer diversas conclusiones interesantes. En primer lugar, vamos a comparar el curso online de 2018 con la experiencia consistente en una presentación oral realizada varios meses después. Entre ambos experimentos, se observa una mejoría global en los resultados y por tanto en la UX. Este progreso se lo atribuimos a dos factores: 1) las mejoras incorporadas en el simulador, como la adición de funcionalidades y la corrección de errores, que hicieron más robusto su uso; 2) las instrucciones que se dieron sobre el uso del simulador aumentaron. En la primera experiencia, para el uso del simulador sólo se proporcionó un vídeo explicativo, mientras que en la segunda experiencia, la guía sobre el simulador se realizó en persona. Esto puede indicar que es necesario dedicar una cantidad significativa de tiempo a la capacitación sobre el uso del simulador antes de que el usuario pueda explotar todas sus capacidades. Los comentarios sobre el simulador de la primera experiencia vistos en la Tabla 7.3 refuerzan esta hipótesis.

Por otro lado, la utilidad percibida (Figura 7.3) obtuvo buenos resultados en el segundo experimento, indicando que a los estudiantes el simulador les pareció interesante para el aprendizaje en MR. Este resultado se refuerza nuevamente con los comentarios de la segunda experiencia contenidos en la Tabla 7.3. Así mismo, el resultado de la LTR aumentó, indicando una mejora en la satisfacción del alumno con el simulador.

En la puntuación de la SUS obtenida en ambas experiencias (Tabla 7.2), el resultado puede interpretarse como “OK” según (Bangor et al., 2009). Al observar sus dos escalas, vemos que la referente a la usabilidad es mayor que la que la referente a la aprendibilidad. Este resultado

TABLA 7.3: Comentarios recogidos de los alumnos en las experiencia presencial y en las experiencias on-line.

#	Comentarios del Curso 2018 sobre el simulador
1	El simulador es una herramienta muy valiosa que no ha sido bien integrada en el curso.
2	Añadiría más ejercicios para realizar en el simulador.
3	Debería haber un mayor número de ejercicios que resolver mediante la utilización del simulador.
4	Me parecería muy interesante integrar más el simulador de resonancia con los conceptos explicados.
5	El simulador es una gran herramienta, muy conseguida.
6	En mi opinión hay que centrarse en la adquisición de la imagen, la forma correcta de hacerlo, buena planificación geométrica, muchos más ejercicios con el simulador (el cual no me ha resultado fácil de usar ya que no conseguía adquisiciones y no excesivamente intuitivo).
7	La parte del simulador intentaría mejorarla ya que funciona un poco lento y no lo he usado mucho por este motivo, en esta parte pondría más práctica.
#	Comentarios de la Presentación sobre el simulador
1	Me ha parecido muy interesante y útil, ayuda a asentar los conocimientos teóricos.
2	Muy útil como otro medio para aprender la resonancia magnética.
3	He encontrado muy interesante el uso del simulador, pero debido a mi falta de conocimiento en este ámbito no he podido aprovecharlo en su totalidad, sin embargo, me ha servido para tener un buen comienzo en la materia.
4	Mi experiencia ha sido algo liosa, ya que aún no hemos visto en clase nada de resonancia magnética, por lo que todo era muy novedoso para mí, llegando a resultar difícil.
#	Comentarios del Curso 2019 sobre el simulador con ITS
1	Me gustaría seguir utilizando el simulador cuándo a acabe el curso. Es de gran ayuda! Gracias.
2	Es un programa muy practico, pero requiere un poco de mejoras para ser más eficiente a la hora de dar indicaciones en la ejecución de los ejercicios.
3	Me ha parecido algo increíble y maravilloso. Cuando era estudiante, me hubiera encantado haber tenido una herramienta como el simulador como el ITS.
4	Me ha parecido muy útil para poner en practica los conocimientos adquiridos. Pienso que es una herramienta de gran valor para el aprendizaje. Seria Bueno poder disponer de mas casos de este tipo.
5	El simulador es genial, nunca imaginé que se pudiera hacer algo así. Pero echo de menos alguien con quien poder comentar si lo estoy haciendo bien o resolver algunas dudas. Seguro que lo hay pero yo no estoy muy en contacto con las nuevas tecnologías y me cuesta.

sugiere que las dificultades que muestran los estudiantes para aprender a utilizar el simulador de MR limitan la puntuación general del SUS. Este es un punto importante, puesto que podría pensarse que uno de los objetivos a perseguir en el diseño de un simulador de resonancia debiera ser conseguir una gran facilidad de aprendizaje y utilización por parte de los usuarios. Sin embargo, un simulador de resonancia con el propósito educativo de preparar a los usuarios para realizar adquisiciones en sistema real de resonancia magnética debe mimetizar, al menos hasta cierto punto, el uso de una máquina real. Una máquina real, sin embargo, es complicada de manejar, de forma que esta dificultad debe ser heredada por el simulador para que los usuarios consigan adquirir los conocimientos en un escenario tan realista como sea posible. Es, por lo tanto, inevitable hasta cierto punto que la usabilidad de un simulador dependa tanto del propio simulador como de la usabilidad del sistema que se simula. Esto pone de manifiesto una limitación de la SUS como instrumento de medida en este tipo de escenarios.

En segundo lugar, se analizan los resultados obtenidos en el tercer experimento realizado durante el curso de 2019. En esta prueba el simulador incorpora el ITS. Fijándonos en los resultados de la utilidad percibida (Figura 7.3), vemos que, al igual que en el segundo experimento, la combinación de las respuestas “totalmente de acuerdo” y “algo de acuerdo” son cercanas al 90 %, por lo que la utilidad del simulador sigue percibiéndose como importante.

De la misma forma, la LTR, aunque superior a la primera experiencia, empeoró con respecto a la segunda. Creemos que la causa de este cambio es el contexto en el que se realiza la pregunta, donde el hecho de ser un curso *online*, con respecto a uno presencial, podría tener una influencia

negativa. En cualquier caso, el resultado en esta última experiencia sigue siendo aceptable.

Finalmente, los resultados del SUS han mejorado notablemente, situándose por encima de 68 y acercándose al valor de 70, el cual establece la usabilidad como “Buena” según (Bangor et al., 2009). Nuevamente, es interesante considerar las escalas que lo componen, en la que la usabilidad mejora levemente, mientras que la aprendibilidad mejora notablemente con respecto al segundo experimento. Los resultados sugieren que la incorporación de un sistema de tutoría hace que el alumno perciba el uso del simulador como más sencillo de aprender. Al alumno se le proporcionó un manual de cómo acceder al simulador con el ITS y cómo debía ser utilizado por lo que, al igual que en la segunda prueba, se trató de dar una capacitación al usuario antes de utilizar este sistema. En los comentarios sobre el simulador con ITS (Tabla 7.3), la mayoría de las impresiones fueron positivas, animando a utilizar el sistema de tutoría en más ejercicios.

## 7.4 ESTUDIO SOBRE LA UTILIDAD EDUCATIVA

Una investigación experimental es una metodología que nos permite establecer relaciones de causa y efecto entre variables, donde las variables pueden ser de índole independiente o dependiente. A la variable independiente, en la investigación experimental, también se denomina variable experimental o tratamiento y es aquella o aquellas de las que se observan los efectos. La variable dependiente, también conocida como criterio, o resultado, se refiere a los resultados del estudio y es aquella o aquellas variables a las que afecta la variable independiente. En otras palabras, partiendo de una hipótesis establecida, la investigación experimental observa los efectos de una o más variables independientes sobre una o más variables dependientes. La variable independiente podría ser el método de instrucción, el tipo de tarea o los materiales de aprendizaje, mientras que la dependiente suelen ser el interés en un tema, la motivación, la nota obtenida en una prueba, etc.

En el caso que vamos a ver a continuación, se observará la influencia de la utilización de una herramienta educativa en comparación con la carencia de la misma. En este caso la variable dependiente estará formada por un instrumento de medida creado para la ocasión. Para la creación de un instrumento de medida de esta índole, hay que tener en cuenta una serie de procedimientos que garanticen que las inferencias que se extraen en función de los resultados sean válidas y fiables.

La validez a menudo se define como la característica por la cual un instrumento mide lo que pretende medir (Kimberlin and Winterstein, 2008), y es la que justifica las interpretaciones de los resultados de la prueba. Esta medida de validez depende de la cantidad y el tipo de evidencias que existen para respaldar los datos recopilados. Esencialmente hay tres tipos principales de evidencia (Fraenkel et al., 2011) que pueden cumplirse:

- Evidencia de validez relacionada con el contenidos. Se refiere a lo apropiado del contenido y a cómo de completo es. También hace referencia al formato del instrumento. En el caso de un instrumento escrito, se trata de asegurar que las preguntas estén bien redactadas y presentadas teniendo en cuenta el lenguaje empleado y el público al que va dirigido.
- Evidencia de validez relacionada con el criterio. Esta evidencia se refiere a la relación que existe entre las puntuaciones obtenidas entre distintos instrumentos de medida. Un ejemplo puede ser el caso de la relación entre la puntuación obtenida en un instrumento que mida la actitud de el individuo realizando una operación matemática y su calificación final en dicha materia.
- Evidencia de validez basada en la construcción. Es la más amplia y menos concisa de los tres tipos. Los investigadores tratan de recopilar una gran variedad de evidencias que les permita justificar las inferencias. Un ejemplo clarificador podría ser el expuesto en Fraenkel et al. (2011). En este ejemplo se realiza un test de honestidad a un grupo de personas, donde definen que una persona honesta, si se encuentra un objeto por la calle, trataría de devolverlo. De este test podría inferirse que aquellos con una nota más alta tendrían una mayor probabilidad de devolver una cartera si se la encuentran abandonada. Entonces, una evidencia de validez basada en la construcción sería dar a todos los participantes la oportunidad de devolver una cartera que se encuentren por la calle y ver si coincide con los resultados del instrumento.

La fiabilidad se refiere a la consistencia de las puntuaciones obtenidas (Kimberlin and Winterstein, 2008). Un instrumento puede ser fiable pero no ser válido. Por ejemplo, un test destinado a evaluar la habilidad de unos estudiantes con un oficio público podría tener puntuaciones fiables, pero en realidad podría evaluar únicamente el conocimiento teórico y no la habilidad del estudiante para comunicarse con el público.

Teniendo esto en cuenta, en esta sección se planteará en primer lugar la hipótesis que trataremos de demostrar. A continuación, mediante el diseño de un experimento y la creación de un instrumento de medida tratará de demostrarse la hipótesis.

Finalmente se analizarán los resultados. En nuestro experimento, la variable independiente será el uso del simulador, mientras que la variable dependiente estará formada por las medidas del instrumento.

## 7.4.1 METODOLOGÍA

Dado un simulador de MRI que cumpla los requisitos definidos en la Sección 5.2, y siendo aplicado en el ámbito radiológico para el aprendizaje de conceptos teóricos y prácticos de MRI, la pregunta a la que queremos responder es: ¿puede el uso de esta tecnología marcar una diferencia con respecto a métodos de enseñanza tradicionales como presentaciones apoyadas por diapositivas? En esta sección se describen los materiales y métodos empleados para alcanzar este objetivo.

### PARTICIPANTES

Los estudiantes fueron reclutados entre los matriculados en dos grados de la Escuela de Radiógrafos del Hospital Clínico San Carlos, Madrid, España, a saber, “Técnico de diagnóstico por imagen y medicina nuclear” y “Técnico de radioterapia y dosimetría”. El número total de estudiantes entre ambos grados era de 80, de los cuales 64 participaron voluntariamente en el experimento. Por otro lado, la mayoría de estos estudiantes no tenía conocimientos previos sobre MR, a excepción de un grupo de 30 estudiantes que habían recibido previamente 4 horas de formación introductoria en esta materia.

### DISEÑO EXPERIMENTAL

La configuración del experimento fue diseñada en colaboración con los miembros de la Escuela de Radiógrafos del Hospital Clínico San Carlos. La experiencia educativa se llevó a cabo en un día durante el horario escolar, desde las 9:00 hasta las 14:00 aproximadamente. Esta elección atendía a un doble propósito. En primer lugar, facilitando su presencia, el número de asistentes se maximizaría. En segundo lugar, al realizarse el experimento de forma controlada, se impedirían interacciones y por tanto contaminación entre los dos grupos en los que los alumnos se dividirían. La Figura 7.4 muestra el diseño experimental de la prueba, que se corresponde con un experimento con una selección aleatorizada y con pre-test y post-test, donde en ambos casos se distingue entre *grupo control* (CG) y *grupo experimental* (EG) (Fraenkel et al., 2011). Los pasos de esta secuencia se describen a continuación:



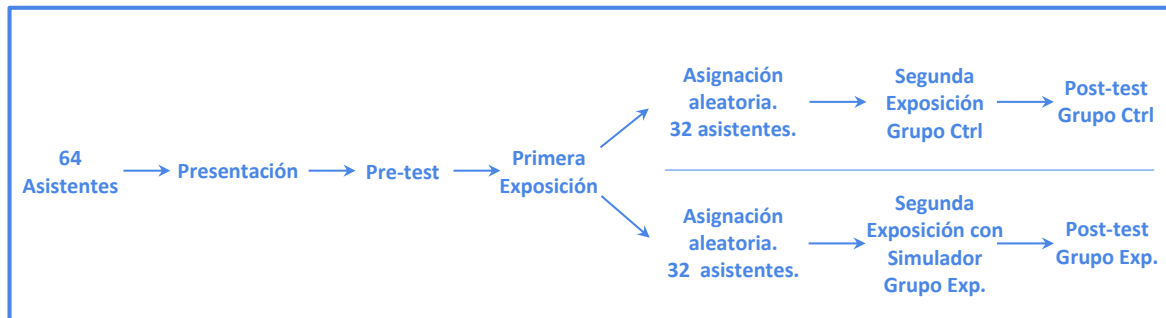


FIGURA 7.4: Diseño del experimento.

- **Presentación:** Se dio una charla introductoria a todos los participantes, en la que se explicaron los aspectos esenciales del diseño experimental. Además, se estableció explícitamente su carácter opcional y se garantizó a los estudiantes la preservación del anonimato y los efectos nulos de este experimento en sus calificaciones finales. Luego, se repartieron los consentimientos escritos siguiendo los estándares éticos de la universidad y una vez firmados por los estudiantes, fueron recogidos.
- **Pre-test:** Se realizó una prueba previa utilizando el instrumento que se describe en la Sección 7.4.1.
- **Primera Exposición:** Se impartió una charla de 90 minutos de duración a todos los participantes. La impartición fue oral con el apoyo de diapositivas, y los temas tratados fueron: las propiedades magnéticas de los tejidos, el concepto de resonancia magnética, los pulsos de RF y gradientes en MRI, el formalismo del espacio k y la formación de imágenes, secuencias de pulsos SE y GRE y algunas pautas de seguridad.
- **Asignación aleatoria:** Los estudiantes fueron divididos al azar entre CG y EG. Posteriormente, EG fue guiado a una sala de ordenadores ubicada en un edificio cercano por el mismo profesor que realizó la primera exposición. Mientras tanto, el CG permaneció en el aula y se le otorgó un breve descanso para permitir la sincronización entre ambos grupos.
- **Segunda Exposición:** A cada uno de los grupos se les impartió una segunda charla de 90 minutos de duración. Los siguientes temas fueron incluidos: la formación del espacio k, algunos parámetros relevantes en las secuencias de MRI (principalmente, TE y TR), la planificación geométrica así como sus parámetros, y algunos artefactos comunes en MRI. En el CG, estos temas se trataron mediante una presentación de diapositivas donde se ilustraron los efectos que tendría la variación de dichos parámetros. Por otro lado, en el EG, los participantes emplearon el simulador de MRI para la realización de ejercicios prácticos guiados para ver los efectos.
- **Post-test:** Se realiza la prueba nuevamente con el mismo instrumento de medida cumpliéndose la sincronía temporal entre grupos.

Como se deduce de lo anterior, dos instructores participaron en las conferencias. Uno de ellos estuvo a cargo de la primera exposición y además impartió la segunda al EG, mientras que el

otro instructor se encargó de dar la segunda conferencia al CG. Los contenidos de la segunda exposición se acordaron de antemano, cubriendo de igual manera los temas formulados en la segunda parte del instrumento de medida. Además, como ha quedado constancia, se mantuvo la comunicación para sincronizar el inicio y la finalización de la sesión.

Cabe mencionar que hubo algunos incidentes menores durante el experimento: (a) dos estudiantes del CG y otro estudiante del EG no asistieron a la segunda charla; (b) un estudiante en el EG contestó deliberadamente “No sé” a todas las preguntas del examen posterior; este participante fue excluido del análisis. Finalmente, el número de estudiantes que completaron el experimento fue de 30 para cada grupo.

## INSTRUMENTO DE MEDIDA

El instrumento de medida es un cuestionario de 20 elementos diseñado para determinar el nivel de conocimientos en MRI. Para su creación, se siguió la metodología especificada en [Fraenkel et al. \(2011\)](#) para conseguir evidencias de validez. Específicamente, esta metodología consistió en: 1) Explicar el objetivo del instrumento así como el contenido que se deseaba evaluar al personal experto, el cual, estaba formado por dos miembros del equipo docente de la escuela de radiógrafos; 2) Redactar de las preguntas por parte de los investigadores; 3) Revisión y crítica por parte de los expertos; 4) Reformulación o refinamiento de las mismas siguiendo la crítica experta. Los puntos 3) y 4) se repitieron iterativamente hasta que todos los elementos del cuestionario fueron aprobados por los expertos. Simultáneamente, para comprobar el formato del instrumento, las preguntas también fueron mostradas a estudiantes de doctorado con conocimientos en resonancia magnética que ayudaron a adaptar el lenguaje al público objetivo y contribuyeron a que las preguntas fueran inequívocas en sus respuestas.

El Apéndice D muestra el cuestionario empleado en la prueba. Cada elemento tiene cuatro posibles respuestas, además de una quinta respuesta “Lo desconozco” común en todos los elementos con lo que se pretende evitar respuestas al azar. Los primeros 10 elementos corresponden al contenido de la primera exposición comentada en la Sección 7.4.1, mientras que los 10 elementos restantes corresponden al contenido de la segunda charla. Dado que el contenido de la primera exposición se corresponde con aspectos teóricos en MR y su exposición fue de manera puramente explicativa, en lo sucesivo nos referiremos a las 10 primeras preguntas del examen como “parte teórica” o “Parte T”. De la misma forma, la segunda parte del examen poseía un contenido más práctico y por ello será referida como “parte práctica” o “parte P”. La puntuación del cuestionario es el número de respuestas correctas, en lo sucesivo, “Aciertos”, y por tanto su puntuación máxima será de 10 en cada una de las partes. Esta puntuación será la utilizada por defecto. Además, en aras de la exhaustividad, también se ha tenido en cuenta una versión de expectativa nula de la calificación, donde las respuestas aleatorias conducen a una puntuación de expectativa nula, “NE” de aquí en adelante. En esa versión, específicamente, los aciertos suman un punto, los errores restan un tercio de punto y las preguntas no contestadas o donde se seleccione la respuesta “Lo desconozco” ni suman ni restan.

Dada la naturaleza de nuestro experimento, no fue posible emplear un instrumento de medición cuya fiabilidad estuviese probada de antemano. Por lo tanto, la fiabilidad del instrumento se calculó *post hoc*, como se describe a continuación.

## ANÁLISIS ESTADÍSTICO

La fiabilidad del instrumento se calculó mediante la Fórmula Kuder-RichardsonKR-20 (Kuder and Richardson, 1937) para la calificación de aciertos, mientras que el alfa de Cronbach (Cronbach, 1951) fue usado para la nota de expectativa nula.

La inferencia estadística se realizó de la siguiente manera: se usó una prueba de normalidad Shapiro-Wilk (Royston, 1982) en las puntuaciones pre-test y post-test de cada grupo para determinar si se debe emplear el t-test o Wilcoxon signed rank test. Una vez determinado el tipo de test a emplear, este test se ejecutó en pruebas bilaterales no pareadas, donde la hipótesis nula asume que las puntuaciones coinciden para ambos grupos, mientras que la hipótesis alternativa es que las puntuaciones son distintas. Finalmente, se realizó también un análisis descriptivo donde se utilizaron la mediana, la media, la desviación estándar y el tamaño del efecto (calculado utilizando Cohen's d (Torchiano, 2018)). Todos estos cálculos fueron realizados en R.

Por consiguiente, el análisis se realizó de la siguiente forma: primero se analizó la fiabilidad de la prueba. Luego, evaluamos la existencia de diferencias estadísticamente significativas entre ambos grupos en el pre-test. En caso de existir dichas diferencias, este hallazgo implicaría que los grupos no están balanceados. Una vez comprobada la equidad en la división de los grupos, se evalúan las posibles diferencias en el post-test, las cuales podrían considerarse un resultado de nuestra intervención. Las diferencias estadísticas en el post-test se evalúan en términos de puntuación y de ganancia, definiendo la ganancia como la diferencia entre las puntuaciones entre el pre-test y el post-test.

### 7.4.2 RESULTADOS

TABLA 7.4: Análisis estadístico del pre-test para el instrumento de 10 elementos. La primera parte de la tabla (Aciertos) utiliza los aciertos en su cálculo, mientras que la segunda (NE) utiliza la nota calculada mediante el método de expectativa nula. TH: test de hipótesis, Med: mediana, SD: desviación estándar; E. size: tamaño del efecto.

Aciertos	TH	Grupo Exp.		Grupo Ctrl.		E. size
	(p-val)	Med	Media±SD	Med	Media±SD	(Cohen's d)
T	0.588	0.50	1.13±1.28	0.50	0.90±1.16	0.191
P	0.938	0.00	0.53±0.82	0.00	0.57±0.86	-0.040
NE	TH	Grupo Exp.		Grupo Ctrl.		E. size
	(p-val)	Med	Media±SD	Med	Media±SD	(Cohen's d)
T	0.552	0.33	0.74±1.29	0.00	0.51±1.23	0.185
P	0.915	0.00	0.13±0.88	0.00	0.22±0.98	-0.096

La ecuación KR-20 calculó unos valores de fiabilidad en la parte teórica del post-test de 0.446, mientras que en la parte práctica fue de 0.318. Estos valores son considerados muy bajos, y en consecuencia, se realizó un análisis de correlación con el objetivo de identificar un subconjunto de preguntas que nos proporcionara una mayor fiabilidad (Tavakol and Dennick, 2011). Como resultado de este análisis, creamos un instrumento de 10 elementos en total, seleccionando los elementos {4, 5, 6, 8, 10} de la parte teórica y los elementos {11, 12, 13, 17, 20} de la parte práctica.

TABLA 7.5: Análisis estadístico del post-test y la ganancia para el instrumento de 10 elementos. La primera parte de la tabla (Aciertos) utiliza los aciertos en su cálculo, mientras que la segunda (NE) utiliza la nota calculada mediante el método de expectación nula. TH: test de hipótesis, Med: mediana, SD: desviación estándar; E. size: tamaño del efecto.

Aciertos	TH (p-val)	Grupo Exp.		Grupo Ctrl.		E. size (Cohen's d)	
		Med	Media±SD	Med	Media±SD		
PT	T	0.540	1.50	1.93±1.48	2.00	2.17±1.51	-0.156
	P	<b>0.018</b>	2.00	2.53±1.36	2.00	1.63±1.27	<b>0.684</b>
Gain	T	0.309	1.00	0.80±1.27	1.00	1.27±1.48	-0.338
	P	<b>0.036</b>	2.00	2.00±1.62	1.00	1.07±1.36	<b>0.624</b>

---

NE	TH (p-val)	Grupo Exp.		Grupo Ctrl.		E. size (Cohen's d)	
		Med	MMedia±SD	Med	Media±SD		
PT	T	0.581	0.83	1.28±1.78	1.17	1.54±1.94	-0.143
	P	<b>0.013</b>	1.83	1.99±1.67	1.17	0.93±1.55	<b>0.655</b>
Gain	T	0.29	0.67	0.53±1.67	1.00	1.03±2.66	-0.276
	P	<b>0.01</b>	1.67	1.86±1.79	0.67	0.71±1.95	<b>0.686</b>

Como resultado, la puntuación de cada una de las partes pasó a ser de 5 puntos. La fiabilidad obtenida para este nuevo instrumento en el pre-test fue de 0.592 en la parte teórica y 0.432 en la parte práctica, mientras que en el post-test fue 0.604 en la parte teórica y 0.570 en la parte práctica. Este nuevo test fue mostrado de nuevo a la comisión experta que consideró que seguía cubriendo gran parte de la materia evaluada.

La Tabla 7.4 muestra el análisis estadístico del pre-test para ambas calificaciones, los aciertos y la expectativa nula. A pesar de las leves discrepancias en los valores medios, no se aprecian diferencias estadísticas significativas, por lo que podemos concluir que los grupos están equilibrados en su situación inicial, como era de esperar. La primera mitad de la Tabla 7.5 muestra los resultados correspondientes al post-test para la calificación basada en los aciertos (Aciertos). Específicamente, su primera columna numérica muestra los valores  $p$  de la prueba de hipótesis sobre la igualdad de las puntuaciones de ambos grupos; Los valores significativos han sido destacados en negrita. Por lo tanto, se aprecian las diferencias en la parte práctica del instrumento entre el CG y el EG, mientras que las dos muestras no presentan diferencias en cuanto a la parte teórica. Obsérvese además que el grupo experimental aprueba la parte práctica (téngase en cuenta que esta parte se evalúa sobre 5). La segunda mitad de la Tabla 7.5 muestra las mismas tendencias para la calificación de expectativa nula (NE). Curiosamente, para esta calificación alternativa se seleccionaron las mismas preguntas que en la calificación anterior para maximizar la fiabilidad.

### 7.4.3 DISCUSIÓN

El valor de fiabilidad del KR-20 para un instrumento dado suele considerarse como bueno a partir del resultado 0,7, aunque en pruebas más exigentes (Frey, 2018) la fiabilidad considerada como aceptable se establece a partir de 0,5. Nuestro instrumento inicial compuesto por 20

elementos no alcanzó este nivel en el post-test; sin embargo el subconjunto creado de 10 elementos sí logró estos requisitos como se muestra en los resultados. Debe enfatizarse además, que la fiabilidad se ve penalizada en el caso de que se produzcan respuestas aleatorias, lo que puede explicar los valores bajos en la parte práctica del pre-test teniendo en cuenta los escasos o nulos conocimientos previos de los participantes en resonancia magnética y por lo tanto las bajas puntuaciones obtenidas por los estudiantes en esta prueba previa y, específicamente, en su parte práctica (Tabla 7.4)

La Tabla 7.4 muestra que nuestro estudio parte de una situación equilibrada con el instrumento de 10 elementos, es decir, los dos grupos no tienen diferencias estadísticamente significativas en sus niveles de experiencia. La Tabla 7.5 resume los resultados de nuestra intervención. Específicamente, las medidas de tendencia central reflejan cómo la puntuación del EG en el post-test y en la ganancia es mayor en la parte práctica que en el CG; En cuanto al subconjunto teórico, se observan resultados ligeramente mejores para el CG. Realizando el cálculo de la inferencia estadística, éste muestra que los resultados de la parte práctica son realmente significativos a favor del EG, tanto para las puntuaciones obtenidas como para la ganancia. Por otro lado, en la parte teórica estas diferencias significativas no se dan. Estos resultados se acompañan de valores significativos en los tamaños del efecto superiores a 0,5, que se dan en la prueba práctica, nuevamente tanto para las puntuaciones como para la ganancia. El conjunto de estos valores nos permite afirmar que el simulador de MRI es una herramienta valiosa para entrenar a técnicos radiólogos cuando se trata de comprender conceptos prácticos.

Por otro lado, en el apartado teórico, a pesar de que los resultados no son concluyentes, podemos interpretar que el CG puede haber recibido un mayor énfasis en la parte teórica ya que la segunda charla no incluyó el trabajo directo de los estudiantes, lo cual es un proceso que consume tiempo; esta cantidad de tiempo de clase, que fue empleado por los estudiantes en el EG para familiarizarse con la herramienta, fue utilizado inevitablemente por el instructor del CG para enfatizar los conceptos de fondo.

El diseño experimental empleado tiene sin duda varias limitaciones. Una de ellas es el posible factor introducido por el hecho de que el grupo de control y el grupo experimental recibieron la segunda exposición de dos instructores diferentes. Sin embargo, esto era una consecuencia inevitable del diseño experimental empleado, que sin embargo fue elegido, entre otras razones, porque la realización de la experiencia en un solo día permitía evitar la contaminación entre el CG y el EG. Por otro lado, un diseño experimental de mayor duración resultaba extremadamente complicado de llevar a cabo por su interferencia en el funcionamiento habitual de la Escuela de Radiógrafos, además de poner posiblemente en peligro la adherencia de los participantes durante toda la experiencia. Finalmente, las conclusiones extraídas deben considerarse en el contexto del empleo puntual del simulador de resonancia en una experiencia de corta duración. Formalmente no es posible extrapolar estas conclusiones a la utilización del simulador durante un periodo más largo de tiempo, que es en el contexto en el que, probablemente, tenga un potencial mayor.

---

## 7.5 CONCLUSIÓN

---

Durante este capítulo se han detallado los experimentos realizados para la evaluación de nuestro simulador de MR. El simulador ha sido utilizado en tres experiencias educativas reales, las cuales han sido aprovechadas para su mejora y su evaluación. En primer lugar, en la Sección

7.3 se analizaron los resultados obtenidos de la UX. En consecuencia, los resultados que comparan las dos primeras experiencias prueban una mejora en la percepción del usuario entre ambas versiones. En el tercer experimento, se incorporó una cualidad única en los simuladores de resonancia magnética, que es el ITS. Como resultado, la experiencia para el estudiante se mejoró, mostrando una mayor facilidad para el aprendizaje. Por otro lado, en la Sección 7.4 el foco se pone en la utilidad del simulador como herramienta de aprendizaje. En este caso, los resultados indicaron que el simulador resulta de utilidad para el aprendizaje en MRI.

En resumen, a lo largo de estas experiencias educativas el simulador ha evolucionado y, en base a los parámetros cualitativos y cuantitativos recopilados, podemos concluir que el sistema actual propuesto es una herramienta educativa valiosa para la capacitación en los principios y la adquisición de resonancia magnética y que ha tenido una buena acogida por parte de los potenciales usuarios.

# CONCLUSIONES, CONTRIBUCIONES Y LÍNEAS FUTURAS

---

**E**N este capítulo se recogen las principales conclusiones obtenidas durante el proceso de elaboración de la tesis a partir de los resultados presentados. Asimismo, se enumeran sus contribuciones. A continuación, se exponen las limitaciones de los métodos empleados, lo cual nos lleva a definir las líneas futuras que esta tesis permite abrir.

## 8.1 CONCLUSIONES

---

La imagen por resonancia magnética es una de las modalidades de diagnóstico por imagen más utilizadas actualmente. Esta técnica aprovecha las características magnéticas de los núcleos de hidrógeno para obtener información sobre las propiedades de los tejidos blandos del cuerpo humano. Se trata de una técnica de gran flexibilidad, no invasiva y emplea radiación no ionizante. Todo ello hace que sea una modalidad de imagen de gran uso clínico, como ya hemos dicho, y que reciba una gran atención tanto desde la perspectiva investigadora como desde las perspectivas industrial y comercial.

Sin embargo, la adquisición de imágenes adecuadas a la patología buscada, y con una calidad suficiente, es un proceso complejo. La primera razón se debe al propio fenómeno NMR, que trae consigo que las técnicas de adquisición de este tipo de imágenes no resulten intuitivas y sean complejas de aprender. Por otro lado, la gran flexibilidad asociada a esta modalidad se traduce en una consola de control de máquina que cuenta con una gran cantidad de parámetros, cuyo conocimiento es fundamental para conseguir imágenes de calidad y exentas de artefactos. Además, cada fabricante emplea su propio software y nomenclatura. Por otro lado, esta modalidad tiene tanto un elevado coste como una enorme demanda asistencial; en consecuencia, las instalaciones existentes tienen una alta tasa de ocupación.

En respuesta al gran desarrollo de la MRI, y debido a la complejidad mencionada, a lo largo de los años han aparecido diversos simuladores. Éstos pueden diferenciarse según su finalidad en: 1) aquellos que han sido creados con un propósito educativo o divulgativo; 2) aquellos destinados a anticipar el comportamiento de nuevas secuencias antes de ser desplegadas en las máquinas reales.

Dentro del primer grupo, cabe destacar los destinados a la formación en MRI de técnicos en imagen para el radiodiagnóstico. Estos simuladores deben disponer de interfaces que mimeticen las interfaces de las máquinas comerciales. Por otra parte, los tiempos transcurridos entre la petición de una simulación y su resultado deben ser breves. Además, como hemos puesto de manifiesto en la tesis, estos simuladores se centran únicamente en la imagen de contraste, de forma que no prevén imitar el proceso completo que un técnico debe realizar en la consola de estas máquinas. Finalmente, hasta donde conocemos, no existen datos cuantitativos que avalen la utilidad o la usabilidad de estas herramientas de simulación.

En respuesta a las mencionadas carencias, en esta tesis se ha desarrollado un nuevo simulador para la formación de este tipo de personal. Para tal fin, hemos recabado las opiniones de radiólogos con elevados conocimientos en el manejo de máquinas de MRI y en el análisis de imágenes de resonancia. Ello nos ha llevado a recoger un conjunto de requerimientos, funcionales y no funcionales, los cuales deben ser cumplidos por un simulador que pretenda dar respuesta a las carencias identificadas. El conjunto de los requisitos ha sido implementado en nuestra herramienta educativa, facilitando además un diseño que permita el mantenimiento y la evolución de la herramienta.

Paralelamente al desarrollo de la herramienta, hemos observado que los ITS han sido utilizados para el aprendizaje de conceptos clínicos y para el uso de maquinarias complejas en diversos dominios de aplicación. Esto nos llevó a un estudio más detallado de los ITS y de las herramientas que facilitasen su creación, ya que uno de los principales problemas de los ITS es la cantidad de tiempo que debe emplearse para su implementación, en comparación con el tiempo que el software resultante sirve para la formación de estudiantes.

Este estudio nos permitió ver que la integración de un ITS en un simulador de resonancia es inédita hasta el momento actual; también vimos que tal integración en nuestro simulador sería perfectamente posible sin prácticamente alterar el código ni del simulador ni del propio ITS.

Las bondades del simulador de resonancia magnética generado han sido puestas a prueba a lo largo de tres experimentos educativos reales. En primer lugar, fue utilizado en un curso online celebrado en el año 2018, con una duración de 8 semanas, donde nuestro simulador fue usado como un complemento a la formación. En segundo lugar, fue utilizado durante una sesión presencial en una jornada de trabajo celebrada en una escuela de técnicos en imagen. Finalmente, en el año 2019, el simulador fue utilizado en una nueva edición del curso online indicado en primer lugar.

Del conjunto de las dos primeras pruebas se tomaron medidas para evaluar la experiencia del usuario. Como resultado, la usabilidad del simulador resultó como “Ok”, donde la escala que medía la aprendibilidad del sistema resultó baja. Por otro lado, la utilidad que los estudiantes percibieron sobre la herramienta fue positiva. Además, de la misma forma, la probabilidad de recomendar el uso de este simulador también obtuvo buenos resultados, y en línea con lo esperado por el resto de puntuaciones.

En el tercer curso, las pruebas se centraron en medir el aporte que suponía la incorporación del ITS. Para ello se habilitó un conjunto de ejercicios totalmente opcionales durante el curso, y se tomaron las mismas medidas de la experiencia del usuario que en los dos estudios anteriores. La conclusión principal fue que la incorporación del ITS mejoró la experiencia del estudiante. Esto es debido principalmente a que la incorporación de este sistema a nuestra herramienta mostró una mayor facilidad para el aprendizaje, como quedó reflejado en la escala de usabilidad.

Durante esta evaluación, también se recogieron comentarios libres de los usuarios, los cuales



reforzaron el resto de puntuaciones y conclusiones obtenidas.

En el curso presencial, además de medir la experiencia del usuario, se aplicó un diseño experimental, el cual consistió en un pre-test, una selección aleatorizada para la creación de un grupo control y de un grupo experimental y un post-test aplicado a cada grupo. En este caso, el grupo control recibió una formación tradicional, sin el uso del simulador, mientras que el grupo experimental trató los mismos temas que el grupo control pero utilizando nuestra herramienta. Los resultados obtenidos mostraron que el simulador había sido realmente determinante en el aprendizaje de conceptos prácticos relacionados con la resonancia magnética.

En conclusión, a la finalización de este proceso de tesis se cuenta con un sistema de simulación que ha evidenciado ser valioso en la formación de técnicos en imagen para el radiodiagnóstico. Esto ha sido posible gracias al conjunto de expertos reunido, a la iteración en su desarrollo y a los resultados que se han evidenciado a partir de los procedimientos de evaluación descritos.

## 8.2 CONTRIBUCIONES

---

Las contribuciones de esta tesis son las siguientes:

- Análisis, diseño e implementación de un simulador orientado realmente al aprendizaje de técnicos de radiodiagnóstico. Esta herramienta, debido a la gran cantidad de útiles y paneles que incorpora, a su gran accesibilidad y los bajos requerimientos para su uso, resulta de especial utilidad. Hasta donde conocemos, en el campo de la enseñanza de esta materia ninguno de los simuladores anteriormente publicados cuenta con estas facilidades.
- Incorporación de un ITS a un simulador MRI. La inclusión de un ITS en el aprendizaje del manejo de una consola MRI es enteramente inédita.
- Evaluación tanto del simulador como herramienta educativa como de la suma del simulador y el ITS en escenarios educativos reales. Hemos obtenido mediciones sobre la experiencia de los alumnos, y hemos conseguido evidencias sobre su utilidad tanto objetivas —medidas comparadas sobre cómo los alumnos han aprendido conceptos con y sin el simulador— como subjetivas —cómo los alumnos han percibido su utilidad—. La evaluación contiene también una parte cualitativa, basada en comentarios directamente obtenidos de los alumnos. Hasta donde conocemos, una evaluación de estas características tampoco ha sido descrita en la literatura.

## 8.3 LIMITACIONES

---

El simulador ha sido diseñado para que pueda producir imágenes de resonancia magnética de forma rápida. Sin embargo, esta ventaja puede ser también considerada su principal limitación, dado que con este tipo de simulación es necesario añadir los artefactos de una forma sintética. Esto entra en contraste con la forma natural en que los artefactos aparecen al aplicar las ecuaciones de Bloch de las formas 2 y 3 vistas en el capítulo 3. En consecuencia, durante el desarrollo de esta tesis, también se probó la aceleración de la simulación mediante el uso de matrices *sparse*

(Treceño Fernández et al., 2016). De este primer paso surgió posteriormente un trabajo de fin de grado, donde dicha simulación fue implementada en GPU (Moya Sáez et al., 2017). Los resultados de estas alternativas fueron razonables, pero seguían sin ser suficientemente rápidos para su implementación en nuestra herramienta educativa. Además, para la simulación en GPU, hay que tener en cuenta cuestiones adicionales, como el alto coste en un servidor con este servicio y su comportamiento ante la petición de varias simulaciones a la vez.

Otra limitación ha venido dada por la adquisición de los distintos modelos anatómicos, que implica que el voluntario del que va a adquirirse una determinada zona anatómica tenga que estar una gran cantidad de tiempo inmóvil en el interior de una máquina de resonancia. Además, los volúmenes resultantes después de aplicar técnicas de relaxometría, muestran zonas de valores inadecuados, los cuales generan simulaciones artefactadas. Como consecuencia, el resultado de la simulación tiene una alta dependencia con los volúmenes del modelo empleado.

En el estudio sobre la utilidad educativa, pese a que en el segundo instrumento se consiguió una fiabilidad aceptable, se detectó la necesidad de crear un instrumento de medida estándar para la evaluación del conocimiento en esta materia. La falta de dicho instrumento limita el alcance de los resultados obtenidos. Por otro lado, esta prueba fue diseñada en un único día con el fin de evitar la contaminación entre los alumnos que iban a utilizar el simulador y los que no iban a utilizarlo. No obstante, la corta duración de la prueba no nos permite extrapolar las conclusiones a la utilización del simulador durante un periodo de tiempo más largo.

## 8.4 LÍNEAS FUTURAS

Las líneas futuras de nuestra herramienta se plantean como una doble vertiente; por una parte, un aumento en la funcionalidad de la herramienta. Por otra, algunas contribuciones científicas que pueden mejorar sensiblemente la herramienta.

En primer lugar, dado que esta interfaz cuenta con el potencial suficiente para seguir evolucionando, en un futuro está planteado que cuente con diversas funcionalidades, tales como una herramienta que emule la señal respiratoria y permita el sincronismo, una mejora que añada la posibilidad de visualizar imágenes en 4D o el incluir el cálculo del SAR. Incorporaciones de este tipo darían un gran valor añadido a nuestra ya compleja interfaz.

Dada la diversidad de marcas (Siemens, Philips, Toshiba, General Electric, etc) que comercializan equipos de MRI, y en vista de que cada una de estas marcas implementa sus propias interfaces y utiliza su propia nomenclatura, la formación de un alumno en una máquina específica debe pasar, necesariamente, por un período de aprendizaje específico. No obstante, teniendo en cuenta el bajo acoplamiento entre la *vista* y el resto de componentes que forman la arquitectura MVVM, nuestra herramienta estaría preparada para la creación de interfaces secundarias de la consola que afectasen principalmente a aspectos estéticos y a la nomenclatura. De esta forma, agregar diferentes apariencias que guarden similitudes con una marca concreta resultaría sencillo para lograr esa formación específica de un usuario. Igualmente, esto podría ser explotado para hacer una aplicación con soporte en varios idiomas.

Por otra parte, nuestra herramienta cuenta con la base necesaria para una configuración compleja de parámetros, y además, el servidor cuenta con un *wrapper* que le permite la llamada a otras librerías en C. De esta forma, sería posible la incorporación de programación paralela en el núcleo de la simulación, lo cual permitiría generar imágenes artefactadas de forma más

natural a como se hace actualmente y en tiempos presumiblemente compatibles con la práctica educativa.

En el aspecto científico, el aumento de los modelos disponibles, sobre todo con patología, y su mejora para realizar una mejor simulación, sería otro de los objetivos en los que investigar. La gran popularidad que han obtenido los modelos de síntesis de imagen a partir de técnicas de aprendizaje profundo abre la puerta a aplicar dichas técnicas a la obtención de nuevos modelos (mapas de PD,  $T_1$  y  $T_2$ ) a partir de las distintas imágenes obtenidas de forma rutinaria en un escáner MRI. Esta posibilidad podría poblar el simulador con una gran variedad de modelos con distintas patologías y mostrar el resultado de las simulaciones sobre éstas.

Con el fin de seguir obteniendo evidencias que puedan, por un lado, darnos medidas de la utilidad del simulador y, por otro, comprobar el progreso de los alumnos en este área, la creación de un test con alta fiabilidad podría suponer un aporte al campo educativo en esta materia. Ésta es otra línea que debe ser explorada.

En esta misma línea de investigación, serían convenientes nuevas aplicaciones y evaluaciones de la utilidad del software como herramienta educativa, en este caso en situaciones de aprendizaje con una larga duración. En las nuevas experiencias sería interesante evaluar aspectos prácticos, como las guías que se da al estudiante durante el aprendizaje con el simulador o los comentarios que proporciona el ITS. De igual modo, sería valiosa la incorporación de nuevas medidas sobre la influencia del simulador en aspectos como la motivación del alumno en el aprendizaje, su implicación con la herramienta o su rendimiento en el aprendizaje.



*Apéndice A*

# CASEIB 2016

---

# Una implementación Eficiente No Paralela de Secuencias de Resonancia Magnética mediante Matrices *Sparse*

Daniel Treceno-Fernández<sup>1</sup>, Juan Calabia del Campo<sup>2</sup>, Rodrigo De Luis García<sup>1</sup>, Carlos Alberola-López<sup>1</sup>

<sup>1</sup> Laboratorio de Procesado de Imagen (LPI), Universidad de Valladolid; <sup>2</sup> Hospital Clínico Universitario de Valladolid

## Abstract

**Keywords:** Resonancia, Secuencias, Matrices *sparse*

## 1. Motivación

La simulación por ordenador de secuencias de resonancia magnética es una actividad de enorme interés para el desarrollo de nuevas aplicaciones y modalidades de la resonancia, así como para su optimización, reducción de tiempos de adquisición y adición de elementos de perturbación realistas, como inhomogeneidades de campo, desplazamiento químico, imperfección en los gradientes y otros [1]. Es una actividad que se lleva a cabo desde mediados de los años ochenta [2] pero continúa siendo un tema de actualidad mediante la introducción de nuevas y más completas funcionalidades [3].

Uno de los grandes problemas de esta actividad es su elevado coste computacional, para lo cual se han propuesto diversas posibilidades, desde la simulación compartimentada empleando el formalismo del espacio K [4], el uso de alternativas a la suma de componentes isocromáticas [5] o, más recientemente, esquemas paralelos [6] o sistemas basados en unidades gráficas de procesado (GPUs) [7].

En este artículo, aún siendo conscientes de estas propuestas ambiciosas, buscamos un planteamiento sencillo para el prototipado de secuencias para programadores no expertos en procesado paralelo. Para ello emplearemos matrices *sparse* de forma natural al llevar a cabo operaciones de rotación sobre vector apilados, de manera que se eviten bucles innecesarios así como unos requisitos de almacenamiento en memoria prohibitivos.

En lo que sigue haremos una descripción sobre una secuencia sencilla basada en spin-echo y con un recorrido del k-espacio similar al empleado en una secuencia EPI (echo planar imaging). Todas estas operaciones pueden llevarse a cabo, como indicamos en detalle, de manera sencilla, mediante operaciones matriciales [2]; estas operaciones matriciales serán implementadas, como hemos indicado, mediante matrices *sparse*.

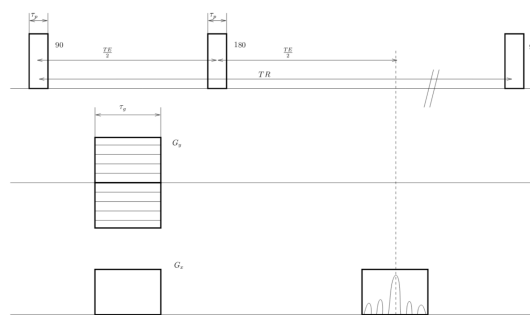


Figura 1. Diagrama de la secuencia spin-echo

## 2. Métodos

### 2.1. Spin echo monoeco

Planteamos el desarrollo para una secuencia básica, como es la secuencia spin-echo [8]. Otras secuencias seguirían unos pasos similares a los aquí descritos. La figura 1 muestra un esquema de la secuencia. Yendo paso a paso, tendríamos:

1.-Pulso de 90, de duración  $\tau_p$  segundos y amplitud constante  $B_1$ . Si existe una inhomogeneidad de campo  $\Delta B(\vec{r})$ , que sea función de cada punto del volumen excitado, tendremos un efecto *off-resonance* [8] a tener presente; en primer lugar, el valor de  $B_1$  debe calcularse de forma que

$$\alpha = \gamma \int_0^{\tau_p} B_1(t) dt = \gamma \tau_p B_1 = \frac{\pi}{2}.$$

Por otra parte, el campo efectivo observado en cada punto será

$$\vec{B}_{eff} = B_1 \vec{x}' + \left( B_0 + \Delta B(\vec{r}) - \frac{\omega_{rf}}{\gamma} \right) \vec{z}'.$$

Siguiendo [2], eso trae consigo que el eje de giro no sea el eje  $x'$  sino un eje separado de  $x'$   $\beta$  radianes, donde  $\beta$  se define

$$\beta = \arctan \left( \frac{\Delta B(\vec{r})}{B_1} \right)$$

y, en realidad, el pulso resulta ser de  $\delta$  radianes, con  $\delta$  definido de la forma

$$\delta = \gamma \tau_p \sqrt{B_1^2 + \Delta B^2(\vec{r})}$$

Por tanto el primer paso consiste en [2]

$$\begin{pmatrix} M_{x'}(0+) \\ M_{y'}(0+) \\ M_{z'}(0+) \end{pmatrix} = \mathbf{R}_{y'}(\beta)\mathbf{R}_{x'}(\delta)\mathbf{R}_{y'}(-\beta) \begin{pmatrix} 0 \\ 0 \\ M_z^0 \end{pmatrix} \quad (1)$$

donde hemos hecho uso de las ecuaciones (11) y (12). Este paso, como se ve a partir de la expresión, consiste en situar el eje de rotación sobre el eje  $x'$ , girar  $\delta$  radianes alrededor de él y devolver el eje a su posición de partida.

2.- Rotación por la inhomogeneidad del campo y los dos gradientes aplicados durante un tiempo  $\tau_g = \frac{T_{acq}}{2}$ . Esta rotación viene dada por el campo distinto a  $B_0$  que se observa, a saber

$$\Delta B_{g_1}(\bar{r}) = (\Delta B(\bar{r}) + G_x x + G_y y) \bar{z}',$$

lo cual se traduce en

$$\begin{pmatrix} M_{x'}(\tau_{g_1}) \\ M_{y'}(\tau_{g_1}) \\ M_{z'}(\tau_{g_1}) \end{pmatrix} = \mathbf{R}_{z'}(\theta_{g_1}) \begin{pmatrix} M_{x'}(0+) \\ M_{y'}(0+) \\ M_{z'}(0+) \end{pmatrix} \quad (2)$$

donde hemos hecho uso de la ecuación (13) y  $\theta_{g_1}$  se define de la forma

$$\theta_{g_1} = \gamma \Delta B_{g_1}(\bar{r}) \tau_g \quad (3)$$

3.- Efecto de relajación durante el intervalo de duración  $\frac{TE}{2}$  entre el pulso de saturación ( $90^\circ$ ) y el de inversión ( $180^\circ$ ). Denominando  $E_1 = \exp(-(\frac{TE}{2})/T1)$  y  $E_1 = \exp(-(\frac{TE}{2})/T2)$ , tenemos

$$\begin{pmatrix} M_{x'}(\frac{TE}{2}) \\ M_{y'}(\frac{TE}{2}) \\ M_{z'}(\frac{TE}{2}) \end{pmatrix} = \begin{pmatrix} E_2 & 0 & 0 \\ 0 & E_2 & 0 \\ 0 & 0 & E_1 \end{pmatrix} \begin{pmatrix} M_{x'}(\tau_{g_1}) \\ M_{y'}(\tau_{g_1}) \\ M_{z'}(\tau_{g_1}) \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 - E_1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ M_z^0 \end{pmatrix} \quad (4)$$

4.- Pulso de inversión de longitud  $\tau_p$  y amplitud  $B_2$ . En este caso, el valor  $B_2$  debe cumplir

$$\gamma \int_0^{\tau_p} B_2(t) dt = \gamma \tau_p B_2 = \pi.$$

Por otra parte, se siguen cumpliendo las condiciones de *off resonance*, donde ahora  $\beta_I$  y  $\delta_I$  son similares a los anteriores pero sustituyendo el valor de  $B_2$ . Así pues

$$\begin{pmatrix} M_{x'}^\pi(0+) \\ M_{y'}^\pi(0+) \\ M_{z'}^\pi(0+) \end{pmatrix} = \mathbf{R}_{y'}(\beta_I)\mathbf{R}_{x'}(\delta_I)\mathbf{R}_{y'}(-\beta_I) \begin{pmatrix} M_{x'}(\frac{TE}{2}) \\ M_{y'}(\frac{TE}{2}) \\ M_{z'}(\frac{TE}{2}) \end{pmatrix} \quad (5)$$

5.- En este momento, disociamos componentes longitudinal y transversal. Al respecto de la primera, la podremos llevar a término del intervalo TR, de forma que tendremos

$$\begin{aligned} M_{z'}(TR) &= E_1 \left( TR - \frac{TE}{2} \right) M_{z'}^\pi(0+) + \\ &= \left( 1 - E_1 \left( TR - \frac{TE}{2} \right) \right) M_z^0, \end{aligned}$$

con  $E_1 \left( TR - \frac{TE}{2} \right) = \exp \left( - \left( TR - \frac{TE}{2} \right) / T1 \right)$ , que es la componente en  $z'$  que se debe usar en la ecuación (1) para el siguiente intervalo.

Por su parte, la componente transversal durante la aplicación del gradiente en la dirección de codificación de frecuencia se puede escribir (según lo indicado en la ecuación (17))

$$M_{x'}^{g_2}(t) + j M_{y'}^{g_2}(t) = E_2 \left( M_{y'}^\pi(0+) + j M_{x'}^\pi(0+) \right) e^{-(\theta_{g_2} - \theta_{g_1})}$$

donde  $\theta_{g_1}$  se define según se ha indicado en la ecuación (3) y  $\theta_{g_2}$  es ahora función del tiempo y resulta ser:

$$\theta_{g_2} = \gamma \Delta B_{g_1}(\bar{r}) t = \gamma (\Delta B(\bar{r}) + G_x x) t$$

y donde la variable  $t$  empieza a contar (desde cero) a partir de la aplicación del segundo gradiente en la dirección de codificación de frecuencia.

Con ello, para cada punto de la zona excitada, tenemos

$$\begin{aligned} M_{x'}^{g_2}(t) + j M_{y'}^{g_2}(t) &= \\ E_2(t) M_{x' y'}^\pi(0+) e^{-j\gamma((\Delta B(\bar{r}) + G_x x)t - (\Delta B(\bar{r}) + G_x x + G_y y)\tau_g)} &= \\ E_2(t) M_{x' y'}^\pi(0+) e^{-j\gamma\Delta B(\bar{r})(t - \tau_g)} e^{-j\gamma(G_x(t - \tau_g)x + G_y\tau_g y)} &= \\ E_2(t) M_{x' y'}^\pi(0+) e^{-j\gamma\Delta B(\bar{r})(t - \tau_g)} e^{-2\pi\frac{\gamma}{2\pi}(G_x(t - \tau_g)x + G_y\tau_g y)} &= \\ E_2(t) M_{x' y'}^\pi(0+) e^{-j\gamma\Delta B(\bar{r})(t - \tau_g)} e^{-j2\pi(k_x(t)x + k_y(t)y)} &= \end{aligned} \quad (6)$$

donde hemos empleado la abreviatura  $M_{x' y'}^\pi(0+) = M_{y'}^\pi(0+) + j M_{x'}^\pi(0+)$  y, además, se han definido

$$k_x(t) = \frac{\gamma}{2\pi} G_x (t - \tau_g) \quad (7)$$

$$k_y(t) = -\frac{\gamma}{2\pi} G_y \tau_g \quad (8)$$

Además,  $E_2(t) = \exp(-(\frac{TE}{2} - (T_{acq}/2 - t))/T2)$

Si el resultado anterior se suma para todo el slice excitado, tendremos la transformada de Fourier espacial de la señal  $E_2 M_{x' y'}^\pi(0+) e^{-j\gamma\Delta B(\bar{r})(t - \tau_g)}$  evaluada en el punto  $(k_x(t), k_y(t))$  indicado. Nótese que esto da lugar a una trayectoria cartesiana de muestreo del  $k$  espacio, donde la ubicación de la componente vertical (codificación de fase) viene dada por el valor de  $G_y$ , de forma que si éste es positivo muestreamos frecuencias verticales negativas y viceversa. Por otra parte, la componente horizontal se muestre de izquierda a derecha, según va aumentando el valor de  $t$  en el intervalo  $0 \leq t \leq T_{acq}$ .

6.- Finalmente, debemos introducir la relajación de la componente transversal hasta final del ciclo, de forma similar a como hemos hecho en el caso de la componente longitudinal. Probablemente en este caso es irrelevante, puesto que en pocos instantes esta componente se pierde, pero incluimos este efecto por si queremos estudiar la influencia de TR en la componente transversal también. Así pues, tendremos:

$$M_{x' y'}(TR) = E_2(TR - TE - T_{acq}/2) \left( M_{x'}^{g_2}(T_{acq}) + j M_{y'}^{g_2}(T_{acq}) \right)$$

con  $E_2(TR - TE - T_{acq}/2) = \exp(-(TR - TE - T_{acq}/2)/T_2)$ .

Así pues, los siguiente ciclos de la secuencia deben simularse a partir de la ecuación (1), donde el vector Spin echo monoeco

$$\begin{pmatrix} 0 \\ 0 \\ M_z^0 \end{pmatrix}$$

debe sustituirse por el vector

$$\begin{pmatrix} M_{x'}(TR) \\ M_{y'}(TR) \\ M_{z'}(TR) \end{pmatrix}$$

## 2.2. Secuencia EPI

La secuencia EPI es una secuencia rápida que se caracteriza por la consecución de un tren de ecos dirigidos por gradientes en vez de por pulsos de RF; por este motivo se consiguen imágenes en tiempos de captación muy breves (decenas de milisegundos) en comparación con otras secuencias como la fast spin echo (también llamada RARE) o las secuencias rápidas de eco de gradiente (por ejemplo, FLASH) [9]. Esta secuencia se puede implementar a partir de otras básicas como el spin echo descrito en este artículo, el eco de gradiente y otras basadas en el paradigma inversión recuperación.

El tren de gradientes en la dirección de codificación de frecuencia junto con los gradientes *blip* en la dirección de codificación de fase, permite la adquisición de múltiples líneas del espacio K a partir de un único pulso de RF. Un ejemplo de transición por el k-espacio para esta secuencia se muestra en figura 2

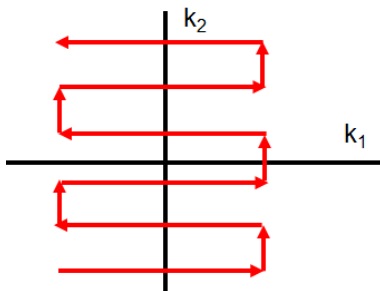


Figura 2. Recorrido del k-espacio en la secuencia EPI.

Para conseguir este efecto basta con multiplicar por unas matrices similares a las indicadas en la expresión 2, si bien ahora el signo de  $G_x$  debe ir alternando línea a línea del espacio K para conseguir barridos en las dos direcciones indicadas por las flechas de la figura 2.

## 2.3. Implementación mediante matrices sparse

Las operaciones que se han indicado en las secciones anteriores deben aplicarse a cada uno de los puntos del cuerpo excitado para dar lugar a la formación de la imagen. Ello trae consigo que las mismas operaciones deban repetirse para cada uno de los puntos de la imagen, lo cual se traduce

en una enorme carga computacional si no se acude a esquemas paralelos, como los que ya hemos descrito en la introducción. No obstante, para el caso de una implementación en una máquina secuencial en la que se desee hacer un primer prototipado de secuencias de resonancia, se pueden conseguir una mayor eficiencia mediante el empleo de rutinas que hagan uso de matrices *sparse*. En lo que sigue tendremos en mente una implementación en Matlab, si bien lo que aquí exponemos es de uso general en librerías que manejen este tipo de matrices.

El procedimiento seguido consiste en evitar bucles mediante la aplicación de las operaciones matriciales a todos los puntos de la imagen simultáneamente. Si la imagen a construir tiene  $N_x \times N_y$  puntos, creamos un vector, pongamos  $\mathbf{z}$ , de  $3 \times N_x \times N_y$  componentes, procedentes de apilar vectores  $\mathbf{z}_i$  como el vector del miembro derecho de la igualdad 1. La aplicación de la matriz  $\mathbf{A}(\theta)$  a cada uno de esos vectores, realizado como operación sobre  $\mathbf{z}$ , se puede llevar a cabo mediante un esquema de matrices bloque

$$\begin{bmatrix} \mathbf{A}(\theta_1) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(\theta_1) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}(\theta_{N_x \times N_y}) \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_{N_x \times N_y} \end{bmatrix} \quad (9)$$

donde la matriz  $\mathbf{A}(\theta_i)$  es una de las matrices indicadas en las ecuaciones (11)—(13). La matriz resultante es una matriz tridiagonal; por concreción, si escogemos la matriz (11) como referencia, los valores de las diagonales (escritos como columnas en lo que sigue) son

$$\begin{bmatrix} 0 & 1 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & \sin(\theta_1) \\ 0 & \cos(\theta_1) & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ -\sin(\theta_{N_x \times N_y}) & \cos(\theta_{N_x \times N_y}) & \sin(\theta_{N_x \times N_y}) \\ 0 & \cos(\theta_{N_x \times N_y}) & 0 \end{bmatrix} \quad (10)$$

Estas matrices se pueden construir, almacenar y procesar de forma inmediata en Matlab mediante el empleo de la orden `spdiags`, tomando como entrada la estructura indicada en (10). Caso contrario, el tamaño de estas matrices haría prohibitivo su uso directo.

Para las matrices de relajación y recuperación (véase ecuación 4) al ser directamente diagonales, la creación de matrices *sparse* es inmediata.

## 3. Resultados

Se ha realizado un experimento de reconstrucción de una sección de una rodilla en el entorno Matlab (version R2014a) empleando la implementación *sparse* aquí propuesta y una implementación secuencial. La imagen tiene un tamaño de  $256 \times 256$  y se han empleado dos diezmos, a saber, por dos y por cuatro en cada dimensión. El procesado empleado es un Intel Xeon a 2.30 Ghz con XX núcleos y dispone de 64 Gb de memoria ram. Los tiempos de ejecución (en segundo) son los que siguen:



	×2	×4
Secuencial	2694	179,37
Sparse	1037	56,27

El procedimiento consigue una reducción de tiempos por un factor de 2,5 – 3 según las dimensiones de la imagen.

#### 4. Conclusiones y líneas futuras

La exposición que se ha hecho de la implementación de las secuencias adolece de ciertas limitaciones: en particular, no hemos tenido en consideración los desfases que puede introducir la inhomogeneidad del campo magnético en los intervalos (véase figura 1) a que no hemos hecho mención, a saber, desde final del pulso de saturación hasta comienzo de los gradientes de  $G_x$  y  $G_y$ , desde el final de éstos hasta el comienzo del pulso de inversión etc ... habría que validar la hipótesis de que este hecho no es relevante. Por otra parte, no se ha tenido en cuenta que las diversas líneas de codificación de fase tienen valores de  $E_2$  distintos, tal y como se escribe en la ecuación (6). En todo caso, hemos cuantificado unas ganancias notables con respecto a una implementación estrictamente secuencial, lo cual permite, como se hemos afirmado en la introducción, que una operativa sencilla permita abordar el análisis de secuencias con unos tiempos de ejecución moderados.

#### 5. Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos TEC2013-44194-P, TEC2014-57428-R (ERDF-FEDER) y VA136U13 y la empresa Giveme5D.

#### Referencias

- [1] Benoit-Cattin H., Collewet G., Belaroussi B., Saint-James H., and Odet C. The SIMRI Project: a versatile and interactive MRI simulator. *Journal of Magnetic Resonance*, 173:97—115, 2005.
- [2] Bittoun J., Taquin J., and Sauzade M. A Computer Algorithm for the Simulation of Any Nuclear Magnetic Resonance (NMR) Imaging Method, Magnetic Resonance Imaging. *Magnetic Resonance Imaging*, 2:113—120, 1984.
- [3] Xanthis C. G., Venetis I.E., Chalkias A. V., and Aletras A. H. Block-Based MRI System Simulator Considering Realistic Electromagnetic Fields for Calculation of Signal, Noise and Specific Absorption Rate. *Magnetic Resonance in Medicine*, 72:237—247, 2014.
- [4] Petersson J. S., Christofferson J. O., and Golman K. MRI Simulation Using the k-Space Formalism. *Magnetic Resonance Imaging*, 11:557—568, 1993.
- [5] Jochimsen T. H., Schäfer A., and Moseley M. E. Efficient simulation of magnetic resonance imaging with Bloch Torrey equations using intra-voxel magnetization gradients. *Journal of Magnetic Resonance*, 180(1):29—38, 2006.
- [6] Stöcker T., Vahedipour K., and Shah N.J. Pflugfelder D. High-performance computing MRI simula-

tions. *Magnetic Resonance in Medicine*, 64(1):186—193, 2010.

- [7] Xanthis C. G., Venetis I.E., Chalkias A. V., and Aletras A. H. MRISIMUL: A GPU-Based Parallel Approach to MRI Simulations. *IEEE Trans. on Medical Imaging*, 33(3):607—617, 2014.
- [8] Z. P. Liang and Lauterbur P. C. *Principles of magnetic resonance imaging. A signal processing perspective*. IEEE Press Series in Biomedical Engineering, 2000.
- [9] Bernstein, M. A. and King F. and Zhou X. J. *Handbook of Pulse Sequences*. Elsevier Academic Press, 2004.

#### A. Apéndice

##### A.1. Operaciones de rotación

Las matrices de rotación en torno a cada uno de los ejes en el sistema móvil de referencia son las siguientes (véase [8], pág. 81):

$$\mathbf{R}_{x'}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (11)$$

$$\mathbf{R}_{y'}(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \quad (12)$$

$$\mathbf{R}_{z'}(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

##### A.2. Aplicación de pulso de inversión

La aplicación de un pulso de inversión en la dirección del eje  $x'$  produce un giro de 180 grados alrededor de dicho eje. Ello se traduce en

$$\mathbf{R}_{x'}(\pi) \begin{bmatrix} M_{x'} \\ M_{y'} \\ M_{z'} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} M_{x'} \\ M_{y'} \\ M_{z'} \end{bmatrix} = \begin{bmatrix} M_{x'} \\ -M_{y'} \\ -M_{z'} \end{bmatrix} \quad (14)$$

Pues bien, respecto de la componente transversal ese efecto es el de la conjugación compleja, es decir, invertir la fase del mismo. En efecto:

$$M_{x'y'} = M_{x'} + jM_{y'} \rightarrow M_{x'} - jM_{y'} = |M_{x'y'}|e^{-j\psi} = M_{x'y'}^* \quad (15)$$

##### A.3. Secuencia: giro-inversión-giro

Si aplicamos al vector de magnetización  $|M_{x'y'}|e^{j\psi}$  sucesivamente las operaciones de giro de  $\theta_1$  radianes, inversión y giro de  $\theta_2$ , tendremos:

$$\begin{aligned} & |M_{x'y'}|e^{j\psi} \xrightarrow{\theta_1} |M_{x'y'}|e^{j(\psi-\theta_1)} \xrightarrow{inv.} \\ & |M_{x'y'}|e^{-j(\psi-\theta_1)} \xrightarrow{\theta_2} \\ & |M_{x'y'}|e^{-j(\psi-\theta_1)}e^{-j\theta_2} = |M_{x'y'}|e^{-j\psi}e^{-j(\theta_2-\theta_1)} \quad (16) \end{aligned}$$

es decir,

$$|M_{x'y'}|e^{-j\psi}e^{-j(\theta_2-\theta_1)} = M_{x'y'}^*e^{-j(\theta_2-\theta_1)} \quad (17)$$



## Apéndice B

# ANGULARJS

---

## B.1 INTRODUCCIÓN

---

Angular es una framework de código abierto para JavaScript ([Google, 2019](#)) que cuenta con el soporte de Google, el cual se encarga de su mantenimiento. Esta framework sigue muy activa; su última versión estable es la 1.7.6 que fue lanzada el 17 de enero de 2019, y su objetivo se orienta al desarrollo de clientes web. En el desarrollo web, el lenguaje HTML fue ideado para la creación de documentos estáticos; por tanto, cuando HTML es utilizado para crear páginas dinámicas, las funcionalidades que aporta son menores. En respuesta, AngularJS trata de extender el vocabulario utilizado en HTML y así mejorar su desempeño en este área.

AngularJS también trata de introducir buenas prácticas en el desarrollo, como, por ejemplo, en la disposición y el orden del código. En este sentido, AngularJS posibilita un desarrollo modular y a su vez introduce pautas para hacer uso del patrón de diseño MVC. Ambos temas son afrontados en la Sección [B.2](#). Además, para la implementación de este patrón crea las herramientas llamadas Controladores y Servicios (Sección [B.3](#) y [B.4](#) respectivamente).

Sin embargo, el principal potencial de AngularJS se encuentra en las directivas, que son las herramientas que habilitan la ampliación del léxico de HTML y que son descritas en la Sección [B.5](#). Finalmente, para ahorrar trabajo recurrente AngularJS cuenta con otra utilidad llamado “Filtro”. Éste está compuesto habitualmente por una función que toma como entrada una información determinada y los parámetros necesarios, retornando en su salida la información depurada (Sección [B.6](#)).

## B.2 MODULARIDAD Y PATRÓN DE DISEÑO

---

En primer lugar, AngularJS propone una herramienta para la agrupación del código, a la que denomina “módulo”. Cada módulo podría ser considerado como un contenedor donde el programador puede situar el código de sus directivas, servicios y controladores. Esto hace que una aplicación compleja, como podría ser el simulador de resonancia, quede dividida en bloques, como podrían ser la consola principal y la aplicación para la visualización de diferentes patologías.

## B.2.1 MVC

---

AngularJS define el patrón de diseño que usa como MVC. Éste se divide en tres capas o componentes que explicamos a continuación.

- **Modelo** (model): representa a la capa que contiene los datos y donde se trabaja con ellos. Es decir, los modelos son los encargados de contener la información y dar acceso a ella. En AngularJS es habitual que sean los encargados de acceder a la base de datos del servidor y de almacenar los datos locales. De esta forma, el modelo en AngularJS quedaría representado por el objeto *\$scope* definido en la Sección B.3.
- **Vista** (view): la vista, como su nombre indica, se centra en en la presentación que van a tener los datos y la interfaz con el usuario. En AngularJS hace referencia al código extendido de HTML, que incluye el código HTML básico y los filtros y directivas de AngularJS.
- **Controlador** (controller): se encuentra a cargo de realizar las peticiones que llegan por parte del usuario a través de la interfaz, haciendo de enlace entre la Vista y el Modelo. En AngularJS está formado por parte del controlador propiamente dicho (Sección B.3) y por los servicios, principalmente los llamados *Factory* y *Service* (Sección B.4).

## B.2.2 DEPENDENCIA INYECCIÓN

---

El patrón de diseño MVC hace referencia a la arquitectura de la aplicación pero no baja al detalle de definir los mecanismo empleados para la creación de los objetos. AngularJS usa el procedimiento denominado *inyección de dependencias* o *Dependency Injection* (DI) donde un objeto proporciona las dependencias a otro objeto mediante un mecanismo de inyección de un objeto en el otro (en vez de la creación de un objeto dentro de otro). La principal motivación de este mecanismo es la posibilidad de reutilización del código y de modularidad de los objetos. Mostramos a continuación la forma de declarar la inyección de objetos en AngularJS; en este caso puede verse cómo al objeto *miCtrl* se le inyecta la dependencia *miNombre*:

```
1 | .constant('miNombre',{ nombre: "Dani", apellido:"Trece"});
2 | .controller("miCtrl", ['$scope', 'miNombre', function(
   |     $scope,miNombre){
3 |     $scope.name = miNombre.nombre;
4 |     $scope.borrarNombre = function(){
5 |         $scope.name = "";
6 |     }
7 | }]);
```

## B.3 CONTROLADOR EN ANGULARJS

---

Los controladores son la herramienta que AngularJS proporciona para manejar la lógica de la Vista. Éstos se ejecutan siempre durante la carga de AngularJS y permiten la introducción

del código para la inicialización de un determinado componente, así como para la gestión de eventos y de su flujo de comportamiento. Esencialmente trata de minimizar la escritura de código JavaScript en la Vista. El aspecto que presenta un controlado es similar al siguiente:

```

1 | .controller("miCtrl", ['$scope', function($scope){
2 |     //miCtrl sería el nombre del controlado
3 |     $scope.miData = {
4 |         var1:null,
5 |     }; //Objeto que contendría información que será
        representada en la Vista
6 |     $scope.miMetodo = function(){
7 |         //Función que puede ser invocada mediante algún evento
        en la Vista
8 |         $scope.miData['var1'] = "otra cosa";
9 |     }
10| }]);

```

La variable *\$scope* es una pieza fundamental en AngularJS. Es la encargada de contener los datos y las funciones y hacerlos visibles desde la Vista. Esto se realiza sin tener que introducir ninguna implementación adicional. Además, habilita el funcionamiento del enlace doble (*double binding*), donde la información fluye en los dos sentidos entre la Vista y el Modelo. Un ejemplo de aplicación del controlador *miCtrl* (panel superior) en una Vista podría ser el siguiente, donde al pulsar *miBoton*, en *miEntr* se mostraría el valor "otra cosa".

```

1 | <div ng-controller="miCtrl">
2 |     <input id="miEnt" type="text" ng-model="miData.var1"/>
3 |     <button id="miBoton" ng-click="miMetodo()">X</button>
4 | </div>

```

Finalmente, para mantener separados los roles, los controladores no deberían manipular el DOM de la página, ya que deben ser independientes de la Vista.

## B.4 SERVICIOS EN ANGULARJS

Los servicios en AngularJS en general están restringidos para ser instancias únicas (singleton) con el objetivo de que la misma instancia sea utilizada en la diferentes partes de la aplicación. Para su creación pueden utilizarse diferentes herramientas:

- **Constant:** sirven para almacenar datos que van a permanecer invariantes. No se permite que tenga dependencias.

```

1 | .constant('misCte',{ cte1: "Dani", cte2:"Trece"});

```

- **Factory:** es la función constructora de un nuevo servicio al cual se le permite tener dependencias. La instancia que devuelve sería la función creada por el propio servicio. Esto resulta ser una versión más completa del servicio *Value* que se ve más adelante.

```

1 | .factory('miFactoria', ['miCte', function(miCte) {
2 |     var miNombre = miCte.cte1 + ' ' + miCte.cte1;
3 |     return miNombre;
4 | }]);

```

- **Provider:** es la versión más flexible de los posibles tipos de servicios. En ocasiones, dependiendo de la circunstancia en la que se inicializa un servicio, éste puede necesitar ser configurado. Esta herramienta permite la configuración del servicio antes de que se inicialice la aplicación. Un ejemplo podría ser el de un servicio que habilite —o no— que se muestren los mensajes por consola. Por defecto, el siguiente código no se mostrarían los mensajes, a no ser que la aplicación se inicie con la variable que lo activa como *true*.

```

1 | //Configuración
2 | miAplicacion.config(['registroProvider', function(
3 |     registroProvider){
4 |     registroProvider.habilitarConsole(true);
5 | }])
6 | //Servicio
7 | miAplicacion.provider('registro', function registroProvider
8 |     (){
9 |     var flagRegistro = false;
10 |    this.enableConsole = function(flag){
11 |        flagRegistro = flag;
12 |    };
13 |    this.$get = function(){
14 |        return {
15 |            debug: function(mensaje){
16 |                if(logToConsole){ console.log(mensaje);}
17 |            }
18 |        };
19 |    };
20 | });

```

- **Service:** es un servicio al que se le pueden inyectar dependencias. La diferencia con otros servicios vistos, es que AngularJS, internamente, utiliza el método *new* sobre este servicio. Para no hacer una instancia de los datos cada vez que inyectamos un servicio, lo habitual es definir la información en un objeto interno del constructor y sólo retornar el objeto que contiene los métodos para manipular y acceder a la información.

```

1 | .service('miServicio', ['misCte', function(misCte) {
2 |     var misDatos = misCte.cte1;
3 |     servicio = {};
4 |     servicio.getMisDatos = function(){return misDatos};
5 |     servicio.setMisDatos = function(data){misDatos = data};
6 |     return servicio;
7 | }]);

```

- **Value:** sirve para almacenar variables, pero éstas no tiene por qué ser necesariamente constantes. No se permite que tenga dependencias. A continuación se muestra cómo este servicio podría retornar un número aleatorio entre 1 y 100.

```
1 | .value('num1to100', function(){
2 |     return Math.floor(Math.random()*100) + 1;
3 | });
```

## B.5 DIRECTIVAS EN ANGULARJS

Las directivas en AngularJS permiten la reutilización de código HTML. Se pueden clasificar según su origen, existiendo las directivas nativas, que son todas aquellas con las que cuenta AngularJS por defecto, y las directivas no nativas, que son las directivas propias o creadas por un tercero. Todas ellas pueden estar asociadas de forma distinta a los nodos del DOM. En la siguiente lista se enumeran estas maneras:

- Como atributo en cualquier elemento del DOM. Dentro de la directiva se usaría la letra “A” para referirse a ello (ver variable `restrict` en la línea 3 del último panel de código de esta sección).

```
1 | <div mi-atributo></div>
```

- Como una clase de CSS en cualquier elemento del DOM. Dentro de la directiva se usaría la letra “C” para referirse a ello.

```
1 | <div class="mi-clase"></div>
```

- Como un elemento del DOM. Dentro de la directiva se usaría la letra “E” para referirse a ello.

```
1 | <mi-elemento></mi-elemento>
```

- Como un comentario en el código de HTML. Dentro de la directiva se usaría la letra “M” para referirse a ello.

```
1 | <!-- directive: mi-comentario -->
```

Por otra parte, la composición de una directiva varía según su propósito, pero su estructura general se mantiene. Está formada siempre por una función anónima a la que se le podrían pasar servicios como variables y un *return* que devuelve un objeto. El objeto puede estar compuesto por las opciones que se ven en el siguiente código:

```
1 | .directive('miDirectiva', function() {
2 |     return {
3 |         restrict: 'E', //Define el ámbito del DOM en el que se
           va a utilizar. Ej: A, C, E, M o combinaciones de
           estos como AE
```

```
4   template:<div></div>, //Establece el código HTML
5   templateUrl:'miDirectiva.html', //Establece la url del
    código HTML5
6   scope: false/true/{}, //Por defecto, en AngularJS el á
    mbito asociado a un elemento, atributo, etc. es el
    de su padre. Con este parámetro es posible asociarle
    uno distinto.
7   link:[function(){}], //Establece una función donde podr
    áan definirse variables y/o métodos del elemento,
    del atributo o del scope.
8   transclude:true/false, //Permite la inclusión de código
    dentro de la propia directiva. En su estado true,
    podría habilitar la inclusión de una directiva
    dentro de otra.
9   require: 'ngModel', //Permite definir parámetros
    obligatorias dentro de la directiva.
10  controller:[function(){}], //Puede considerarse similar
    a link. Pero éste está pensado para directivas que
    cuente con una Vista y un Controlador.
11  };
12 }
```

Finalmente, a modo informativo, se mencionan algunas directivas nativas de AngularJS y la finalidad para la que han sido diseñadas.

- **ng-app**: directiva que indica el elemento base de la aplicación de AngularJS.
- **ng-controller**: directiva que asocia el controlador a una Vista, encargándose de mantener vinculado el Modelo y de la gestión de los eventos.
- **ng-model**: directiva que se encarga de vincular una variable del controlador a la Vista.
- **ng-if**: directiva que permite mostrar o ocultar elementos del DOM.
- **ng-click**: directiva que se encarga de asociar alguna funcionalidad al clic sobre el elemento.
- **ng-init**: directiva que permite evaluar una expresión en el ámbito donde se trabaja (scope).
- **ng-repeat**: directiva que permite iterar sobre un conjunto de datos. Para cada uno de los datos genera su correspondiente nodo en el DOM.

## B.6 FILTROS EN ANGULARJS

Los filtros permiten la modificación de los datos, generalmente con el objetivo de formato o presentación. Estos filtros pueden utilizarse tanto en HTML, como en directivas, controladores o servicios de AngularJS como dependencias. Sin embargo, su uso más habitual es para la



visualización de datos en la Vista. El aspecto que muestran habitualmente en el código HTML es el siguiente:

```
1 | {{ expresión | filtro1 | filtro2 }}
```

donde la expresión podría ser una cadena como “hola mundo” y donde el filtro1 podría eliminar los espacios y el filtro2 poner las letras en mayúscula, dando como resultado “HOLAMUNDO”. Al igual que sucede con las directivas, AngularJS cuenta con filtros propios ya implementados. A continuación se nombran algunos:

- **Currency:** Se utiliza para mostrar el resultado como moneda. Por defecto la moneda es el dólar americano.

```
1 | {{ 13 | currency }} --> <<$13.00>>
2 | {{ 13 | currency: 'DT' }} --> <<DT13.00>>
```

- **Date:** Se utiliza para dar formato a fechas.

```
1 | {{1288323623006 | date: 'medium'}} --> <<Oct 29, 2010
   | 5:40:23 AM>>
```

- **LimitTo:** Se utiliza para establecer un límite máximo.

```
1 | {{ [1,2,3,4,5,6,7,8,9] | limitTo:3 }} --> <<[1,2,3]>>
2 | {{ "abcdefghi" | limitTo:4 }} --> <<'abcd'>>
3 | {{ 2345432342 | limitTo:5 }} --> <<23454>>
```

- **Lowercase:** Se utiliza para poner una cadena en minúsculas.

```
1 | {{ Daniel | lowercase }} --> <<daniel>>
```

- **Uppercase:** Se utiliza para poner una cadena en mayúsculas.

```
1 | {{ Daniel | uppercas }} --> <<DANIEL>>
```

Finalmente, para la creación de un filtro personalizado se utilizaría la herramienta *filter*. Y en su interior se crearía la función deseada. Seguidamente se muestra un ejemplo implementado en el simulador. Su finalidad es recortar una cadena de caracteres a un determinado número y añadir una terminación para indicar que está inacabada.

```
1 | .filter('trimString', function () {
2 |     return function (string, maxLength, tail) {
3 |         if(!string || string.length<=maxLength ){
4 |             return '';
5 |         }
6 |         var newStr = string.substr(0, maxLength);
7 |         if(tail && (typeof(tail)=="string")){
8 |             newStr = string+tail;
9 |         }else{
```

```
10     newStr+"...";
11     }
12     return newStr;
13 };
14 })
```

Un ejemplo de uso sería:

```
1 {{ SoyUnaCadenaDemasiadoLarga | trimString:10:'NA' }} -->
   <<SoyUnaCadeNA>>
```

## Apéndice C

# PLANIFICACIÓN ESPACIAL: ANÁLISIS, DISEÑO E IMPLEMENTACIÓN.

---

## C.1 INTRODUCCIÓN

---

En la sección 2.1.2, el punto 7 describe la necesidad de especificar espacialmente la posición y el ángulo de los cortes que van a ser adquiridos. Esta localización es también denominada como planificación espacial y debe atender tanto a interacciones gráficas como textuales.

Además, en función de la zona anatómica a adquirir, su realización puede ser más o menos compleja y va a tener que ser realizada en varios pasos. Por ejemplo, en una adquisición habitual de cerebro, las imágenes adquiridas con el survey suelen ser suficientes para la localización de las diferentes zonas del cerebro. Sin embargo, en una adquisición cardiaca, los cortes deben ser adquiridos siguiendo los ejes del corazón. Esta localización necesita primero identificar dichos ejes y posteriormente trazar los cortes, por lo que no es inmediata. En la Figura C.1) pueden verse algunos de estos cortes cardiacos así como las instrucciones que da el fabricante para su localización.

Finalmente y determinado por el modo en el que van a ser tomados los cortes, la adquisición puede ser volumétrica o corte a corte, y su planificación puede ser cartesiana o radial. En este apartado nos centramos en la planificación corte a corte cartesiana.

## C.2 ANÁLISIS

---

A continuación se establecen los requisitos que la planificación nombrada en el punto C.1 debe cumplir. Para su recopilación se ha estudiado el comportamiento de la planificación en máquinas comerciales de la marca Philips y General Electric y posteriormente se han realizado reuniones con radiólogos y técnicos en imagen médica. Como consecuencia se han identificado los siguientes FR y NFR. Con el fin de simplificar las explicaciones, las imágenes que van a ser tomadas como referencia para realizar la planificación van a ser llamadas o *imágenes de referencia* o simplemente *imágenes*. El área que determina la zona y los cortes de adquisición va a ser llamada ROI. El lugar donde va a mostrarse por pantalla el conjunto de la ROI y la imagen va a ser llamado visor. A continuación se enumeran los FR:

1. La ROI debe ser dibujada correctamente sobre la imagen de referencia y mostrada en el

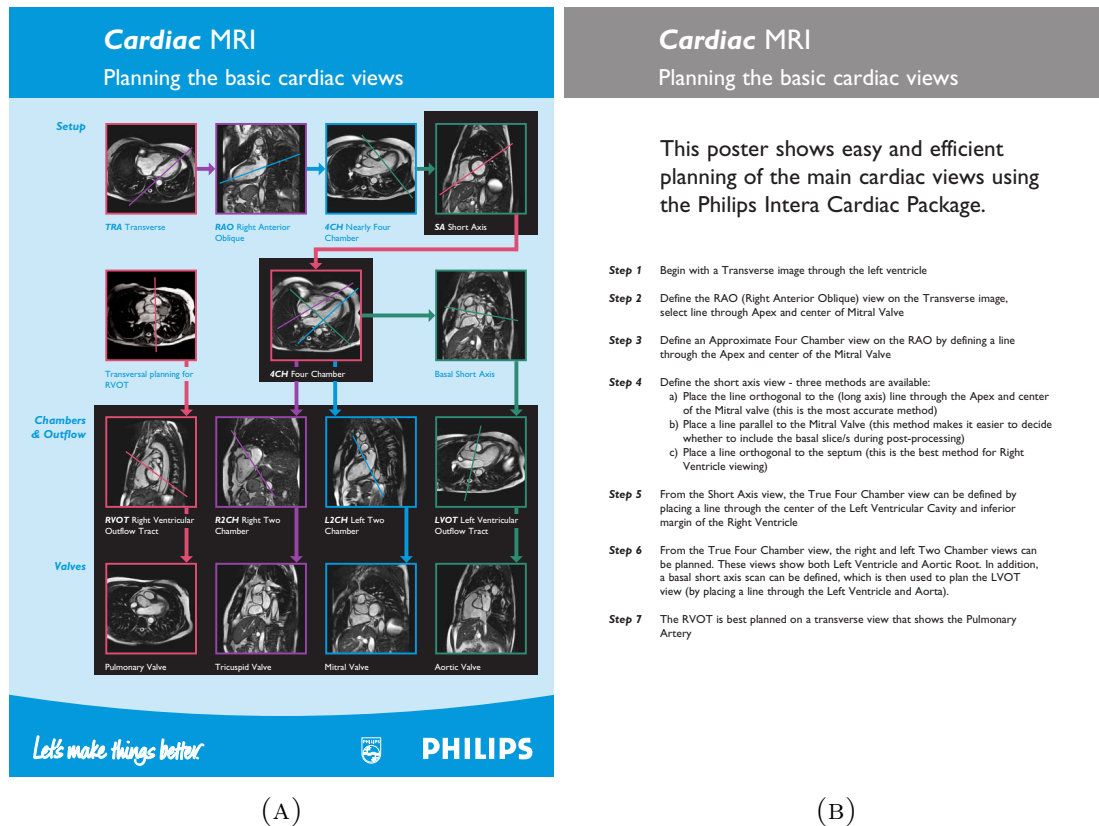


FIGURA C.1: Planificación de los planos cardiacos básicos. En la imagen (A) pueden verse los tipos de cortes usados en el diagnóstico de patologías cardiacas. En la imagen (B) se describen los pasos que indica el fabricante para su obtención. Las imágenes han sido obtenidas del manual de Philips de un escáner MRI.

visor.

- Existirán diferentes visores que podrán ser utilizados a la vez por el usuario.
- La imagen de referencia podrá ser sustituida o modificada en cualquier momento por el usuario. La modificación o sustitución de la imagen consistirá en:
  - El cambio del corte actual por otro corte de la misma adquisición.
  - El cambio de la imagen de referencia por otra de una adquisición distinta.
  - La modificación de su brillo, su contraste o su tamaño.
- La ROI podrá ser sustituida o modificada en cualquier momento por el usuario. La modificación o sustitución de la ROI consistirá en:
  - El cambio de la ROI actual por una anteriormente utilizada.
  - La variación de diferentes parámetros de la ROI, como su angulación, su posición, el grosor del corte, la separación entre cortes, número de cortes o el tamaño del corte.

La modificación de la angulación y la posición de la ROI deben ser realizadas de forma interactiva con el visor.

5. Con finalidad educativa, es interesante que el usuario pueda tener una visión gráfica de la planificación y cómo afecta su modificación en un entorno 3D. Por tanto, el sistema debe poder mostrar una combinación 3D de la ROI y los planos mostrados. Este visor contará con interactores básicos como el desplazamiento y el giro del entorno 3D.

Los NFR son:

1. Los cambios o modificaciones de la imagen de referencia deben ser visibles en cualquiera de los visores en tiempo real.
2. Los cambios o modificaciones de la ROI deben ser visibles en cualquiera de los visores en tiempo real.

## C.3 DISEÑO

Tanto el diseño del sistema como el diseño del programa para este panel es comentado brevemente en esta tesis en la Sección 5.3. Sin embargo, a continuación, y de forma más exhaustiva, se hace un repaso de su diseño mostrando el cumplimiento de los FR y NFR. Para ello que se reutiliza la Figura 5.6.

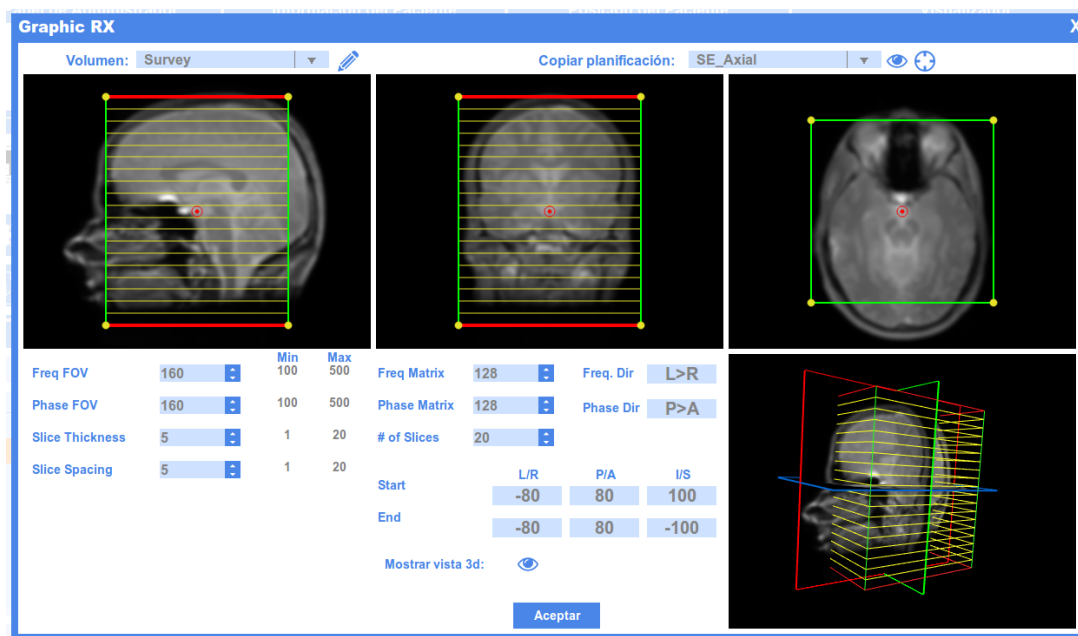


FIGURA C.2: Panel para la planificación geométrica mostrado en la Figura 5.6 de esta tesis.

En primer lugar, en la parte superior de la Figura C.2 se ven dos menús desplegables con las etiquetas de “Volumen” y de “Copiar planificación” respectivamente. Estos serían los interactores

creados para permitir tanto el cambio de la imagen de referencia como de la ROI satisfaciendo los FR [3b](#) y [4a](#). Los 3 recuadros negros situados en la parte superior de la figura serían los visores, que contienen una imagen 2D de referencia y el dibujo de la ROI sobre cada una de ellas (FR [1](#) y [2](#)). Las interacciones que permitiría el visor con la imagen de referencia serían (FR [3a](#) y [3c](#)):

- Mantener pulsado el clic izquierdo del ratón y realizar movimientos verticales u horizontales. Esto permite variar el brillo y el contraste de la imagen.
- Mantener pulsado el clic derecho del ratón y realizar movimientos verticales u horizontales. Esto permite alejar o acercar la imagen.
- Utilizar la ruleta provocaría el desplazamiento por los diferentes cortes adquiridos.

Las interacciones que permitiría el visor con la ROI:

- Mantener pulsado clic izquierdo sobre los puntos amarillos situados en los vértices de la ROI y desplazar el ratón. Esto debería provocar la rotación de la ROI usando como eje el vector normal al plano de la imagen de referencia.
- Mantener pulsado clic izquierdo sobre las líneas rojas de la ROI y desplazar el ratón en dirección perpendicular a estas. Esto aumentaría o reduciría el número de cortes de la ROI.
- Mantener pulsado clic izquierdo sobre el punto rojo de la ROI y desplazar el ratón. El punto rojo representa el centro y permite el desplazamiento de la ROI en el plano de la imagen de referencia.

En la Figura [C.2](#), abajo a la derecha, se muestra un visor 3D que por defecto está desactivado, y que puede activarse mediante el uso del botón “Mostrar vista 3d”. Este visor 3D cumple el FR [5](#) y contará con los siguientes interactores:

- Mantener pulsado el clic derecho del ratón y moverlo horizontal o verticalmente. O, también, utilizar la ruleta del ratón. Permite alejar o acercar la escena tridimensional.
- Mantener pulsado el clic izquierdo del ratón mientras se desplaza el cursor. Esto produce la rotación de la escena sobre el centro de coordenadas del volumen.
- Mantener la ruleta pulsada y desplazar el ratón hace que la escena se desplace.

Finalmente, el resto del panel está dedicado a los parámetros de la planificación. Los siguientes tendrán un efecto sobre la ROI: el FOV en la dirección de frecuencia y de fase “Freq FOV” y “Phase FOV”, el grosor del corte “Slice Thickness” y el espacio entre cortes “Slice Spacing”. El número de cortes “# of Slices” tendrá una relación bidireccional con la ROI, es decir, la ROI podrá modificarse desde este parámetro, pero este parámetro variará si es modificado desde alguno de los visores en la ROI. Además la dirección de frecuencia y de fase, “Freq Dir” y “Phase Dir”, así como la posición inicial “Start” y final “End” de la ROI serán modificadas únicamente interactuando con el visor.

## C.4 DETALLES DE IMPLEMENTACIÓN

---

La combinación de la ROI con las imágenes de referencia sólo es visible desde el panel de planificación y es una de las partes más identificativas y únicas de nuestro simulador. Esta sección tratará de detallar cómo ha sido su implementación usando como esquema la Figura C.3.

Pero, antes de comenzar con el detalle, debemos hablar de la librería XTK, con la cual es posible visionar imágenes con un formato médico, tanto en entornos 2D como 3D. Sin embargo, presenta algunas limitaciones:

- El visor 2D con el que cuenta sólo está pensado para mostrar las imágenes, pero no permite superponer nueva información a estas.
- El visor 3D sólo permite el dibujo de objetos usando un bajo nivel, es decir, es necesario describir cada uno de las líneas y puntos que van a ser dibujados así como su color.

En primer lugar, se trató de modificar el código de XTK para permitir dibujar sobre la imagen 2D de forma nativa, pero dada la gran cantidad de clases, su anidamiento, su complejidad y su poca documentación, resultaba muy complicado introducir esta función de forma nativa, por lo que se descartó. Sin embargo, durante el estudio de este código fue posible encontrar las funciones encargadas de realizar la transformación, por lo que fue posible modificarlas para permitir el acceso en todo momento a las matrices de transformación entre el espacio RAS y la pantalla. En consecuencia, ahora era posible obtener la imagen 2D y su información espacial.

Volviendo a la Figura C.3, esta cuenta con 3 elementos principales que describiremos a continuación:

- **ROI** (Figura C.3a): es un hexaedro que es definido como un único objeto, y que contiene los 8 vértices en coordenadas RAS que lo forman. Además contiene la relación de los vértices con las aristas y las caras, así como las aristas que forman cada cara. Las líneas amarillas representan los cortes que van a ser adquiridos y se calculan automáticamente. Para que esto pueda ser visible en el entorno 3D XTK es necesario la creación de un objeto nuevo que utilice esta información de forma que sean entendible por esta librería. Cuenta también otras variables para fines diversos, como la que permite su visibilidad o las que definen colores que van a tener cada una de las líneas. Por otro lado, la ROI tiene diversos métodos para su modificación y que permiten su expansión en cada una de las direcciones, su rotación y su desplazamiento. Cada vez que alguno de estos métodos son ejecutados, produce la modificación y/o recalculado de varios de los atributos nombrados anteriormente.
- **Imágenes de referencia** (Slice 1, 2 y 3 de la Figura C.3): estas imágenes 2D parten de coordenadas RAS, sin embargo, gracias a XTK es posible obtener dichas imágenes en coordenadas del canvas. Además, es posible obtener las matrices que permiten el cambio de coordenadas RAS a coordenadas x e y del mundo (siendo z igual a 0) y finalmente a coordenadas de la pantalla (canvas). Este proceso puede verse mejor en la Figura C.3b.

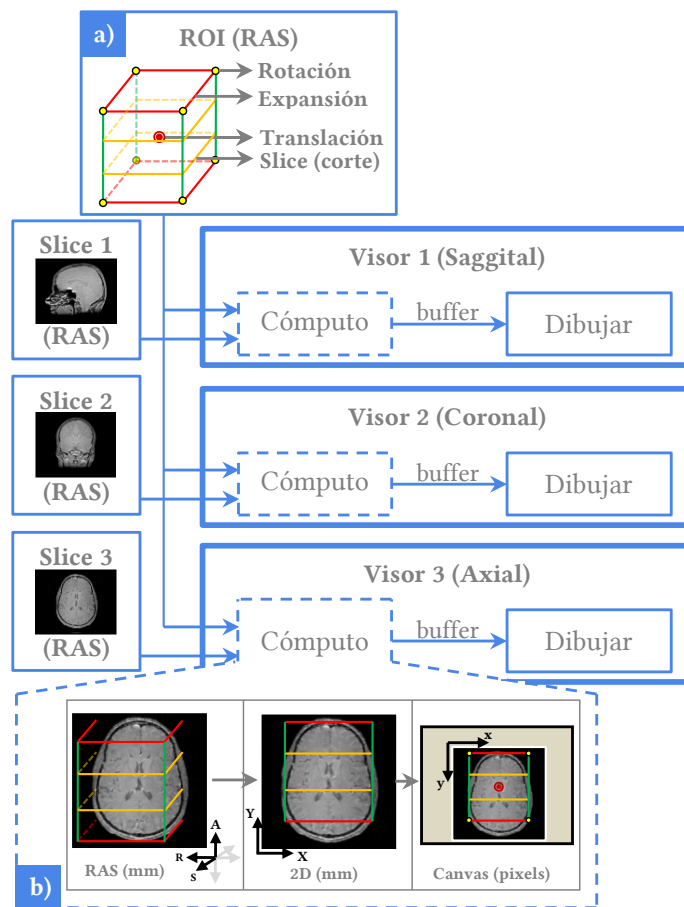


FIGURA C.3: Implementación de la ROI. El recuadro a) ilustra la ROI como objeto que contiene las posiciones de los cortes así como los métodos que permiten su modificación. Este objeto es compartido por los diferentes visores. Cada visor cuenta además con una imagen de un corte. Utilizando la ROI y la imagen como entradas, se realiza el cálculo de la proyección de la ROI sobre la imagen —mostrado en b)— y finalmente se dibuja en pantalla.

- **Visores** (Visor 1, 2 y 3 de la Figura C.3): Estos son los encargados de mostrar las imágenes por pantalla y siguen el siguiente proceso. En primer lugar se toman como entradas la ROI y la imagen de referencia, la cual es distinta para cada visor. A continuación se ejecuta el bloque de cómputo, que calcula los puntos de corte entre la imagen de referencia y la ROI y que se explica más detalladamente después. Además, una vez obtenidos, se realiza la transformación a coordenadas de la imagen. El resultado se almacena en un buffer, cuando todos los cálculos estén realizados, la imagen pasa a dibujarse por pantalla.

Esta es la idea general de la implementación, sin embargo, hay otros aspectos que deben ser tomados en cuenta:

- *Controlador de los visores*: Es un único objeto encargado de determinar qué adquisición va a ser la utilizada como referencia en cada visor, así como qué ROI va a ser usada.



Además también es el encargado de detener los cálculos cuando estos no sean necesarios, es decir, cuando no existan imágenes de referencia o ROIs, cuando la ROI no sea visible o cuando el panel esté cerrado.

- *Interfaz entre la ROI y el panel:* La ROI no cuenta con parámetros como el grosor de los cortes o su espacio, sin embargo esta varía cuando dichos parámetros son modificados. De la misma forma, cuando se varía el tamaño de la ROI en alguno de los visores, la información es actualizada en el panel. La realización de este tipo de comunicaciones sería el cometido de este objeto. Además estaría preparado para futuras formas de definir la ROI, como sucedería con la planificación radial, donde el hexágono ahora sería un cilindro y los cortes se distribuirían de forma radial.
- *Cálculo de los puntos de corte:* Los puntos de corte se calculan en 3 fases. En primer lugar se calculan los puntos de corte entre las caras de la ROI y el plano. Para ello se proyectan las líneas de las aristas hasta el plano en coordenadas RAS. Si el punto por el que corta el plano no está comprendido entre los extremos de la arista, este no es almacenado. Una vez obtenidos todos los puntos se realiza el cambio a coordenadas de la pantalla. Como se cuenta con la información de a qué cara pertenecía cada una de las aristas, las intersección de que pertenecen a la misma cara son unidas con una línea del color asignado a la cara (rojo o verde). Además, aquellos puntos que se correspondan con las aristas serán dibujados como círculos amarillos. En segundo lugar se calcula el punto de intersección del centro con el plano en las 3 posibles direcciones marcadas por las aristas del hexaedro, si alguna de ellas está dentro del área definido en el paso anterior, el centro es transformado a coordenadas de la imagen y dibujado. En tercer lugar se realiza el cálculo de los puntos de corte para cada uno de los posibles slices que definen la adquisición, el proceso es similar al del primer caso, y finaliza dibujando en color amarillo su resultado.
- *Interacciones con visor:* cada visor no se encarga sólo de dibujar, si no que también está preparado para detectar las diferentes acciones del ratón. Nativamente es posible conocer qué botón del ratón ha sido pulsado así como la posición x e y de la pantalla. Por otro lado, cuenta con un vector que contiene todos los objetos interactivos que han sido dibujados con el cálculo de los puntos de corte, a saber: el punto rojo del centro, los puntos amarillos de los vértices y las líneas rojas de las caras desplazables. Si se realiza un clic del ratón sobre uno de estos elementos se aplicarán los efectos descritos en el diseño. Por ejemplo, en el caso del desplazamiento, se almacenarán las x e y iniciales, y cuando el ratón se desplace, sus coordenadas actuales se compararán con las iniciales. La diferencia entre ambas indicará el desplazamiento, que deberá traducirse a coordenadas RAS para indicarle a la ROI cuánto debe desplazarse.



*Apéndice D*

## INSTRUMENTO DE MEDIDA

---



NOMBRE Y APELLIDOS:				
FECHA:		<b>POST-TEST</b>	ID:	GRUPO: <b>X</b>

- Instrucciones:**
- Lea atentamente cada una de las preguntas y señale la opción que considere correcta.
  - Cada pregunta ofrece 4 posibles respuestas. Cada pregunta tiene una única respuesta válida, aunque pueda haber respuestas parcialmente correctas. Se ruega que haga uso de la quinta opción, (letra e)), cuando desconozca la respuesta correcta (es decir, es preferible que evite respuestas aleatorias).
  - El tiempo aconsejado para la duración de este test es de aproximadamente 30 minutos.
  - Marque la opción que considere correcta rodeando con un círculo la letra. Si se ha confundido, tache el círculo y rodee con un círculo la nueva respuesta.

a) Respuesta válida

a) Respuesta tachada

1.- ¿Cuál de los siguientes elementos se puede introducir de manera segura en la sala de un escáner de resonancia magnética?

- a) Herramientas de cobre.
- b) Hemóstatos quirúrgicos de acero inoxidable.
- c) Tijeras quirúrgicas de acero inoxidable.
- d) Todas las anteriores.
- e) Lo desconozco

2.- ¿Por qué la señal de MRI que medimos es la procedente de la componente perpendicular al campo magnético principal?

- a) Porque la componente transversal varía en el tiempo mucho más deprisa que la longitudinal.
- b) Porque la componente longitudinal varía en el tiempo mucho más deprisa que la transversal.
- c) Ambas varían a la misma velocidad, pero la longitudinal es indetectable porque se solapa con el campo principal, el cual es mucho más intenso.
- d) Todas las anteriores.
- e) Lo desconozco

3.- ¿A qué se debe el desfase entre spines dentro de un vóxel?

- a) A que la absorción de radiofrecuencias por el tejido destruye la señal de resonancia magnética.
- b) A que el campo magnético no es homogéneo (constante) dentro del vóxel.
- c) A la aplicación temporal de un pulso de RF.
- d) Ninguna de las anteriores.
- e) Lo desconozco.

4.- ¿Por qué surgen las diferencias de señal en T1 entre los diferentes tejidos (en captaciones sin inyección de contraste)?

- a) Porque se producen diferentes cantidades de decaimiento de la señal antes de tomar una medición de la misma.
- b) Porque se producen diferentes cantidades de recuperación de la magnetización antes de iniciar el proceso de medir la señal de MRI.
- c) Porque se produce un cambio espontáneo del parámetro T1 de los tejidos.
- d) Todas las anteriores.
- e) Lo desconozco.

**5.- ¿Por qué surgen las diferencias de señal en T2 entre los diferentes tejidos?**

- a) Porque se producen diferentes cantidades de decaimiento de la señal antes de tomar una medición de la misma.
- b) Porque se producen diferentes cantidades de recuperación de la magnetización antes de iniciar el proceso de medir la señal de MRI.
- c) Porque se produce un cambio espontáneo del parámetro T2 de los tejidos.
- d) Todas las anteriores.
- e) Lo desconozco.

**6.- ¿Qué es un pulso de RF?**

- a) Un campo magnético de corta duración que oscila en el rango de las radiofrecuencias.
- b) Un gradiente de campo magnético.
- c) Una señal pulsada como la empleada en telegrafía.
- d) Todas las anteriores.
- e) Lo desconozco.

**7.- ¿Qué es el espacio k?**

- a) El espacio k es el lugar en el que almacenamos los datos de señal de MRI digitalizados.
- b) El espacio k es una gráfica infinita de sinusoides complejas.
- c) El espacio k es una matriz de coeficientes que ponderan las frecuencias espaciales.
- d) Todas las anteriores.
- e) Lo desconozco.

**8.- ¿Cuándo se aplica el gradiente de codificación de frecuencia?**

- a) Durante la lectura de la señal de resonancia magnética.
- b) Durante la grabación en disco de la imagen reconstruida.
- c) Durante la codificación de fase.
- d) Todas las anteriores.
- e) Lo desconozco.

**9.- Se utiliza un pulso de RF de 180° para \_\_\_\_\_ los spines que macroscópicamente dan lugar al vector de magnetización neta en el plano transversal.**

- a) Relajar.
- b) Reenfocar
- c) Atenuar.
- d) Retrasar.
- e) Lo desconozco.

**10.- En las secuencias de Spin Echo, TR y TE se refieren a:**

- a) TR: el espaciado entre pulsos sucesivos de 90°; TE: el tiempo entre el pulso de 90° y el eco.
- b) TR: el espaciado entre pulsos sucesivos de 90°; TE: la separación entre el pulso de 90° y el pulso de 180°.
- c) TR: tiempo entre el pulso de 90° y el eco; TE: el espaciado entre pulsos sucesivos de 90°.
- d) TR: la separación entre el pulso de 90° y el pulso de 180°; TE: el espaciado entre pulsos sucesivos de 90°.
- e) Lo desconozco.

**11.- En la Figura 1, la fila superior muestra tres posibles formas de rellenar un espacio k de resonancia cerebral; ¿cuál de las imágenes de MRI —fila inferior— se corresponde con el espacio k etiquetado como A?**

- a) D
- b) E
- c) F
- d) La pregunta está mal formulada: No se proporciona suficiente información.
- e) Lo desconozco.

**12.- De nuevo, en relación a la Figura 1 ¿cuál de las imágenes de MRI —fila inferior— se corresponde con el espacio k etiquetado como C?**

- a) D
- b) E
- c) F
- d) La pregunta está mal formulada: No se proporciona suficiente información.
- e) Lo desconozco.

**13.- En la RM, la variación únicamente del tamaño de la matriz determina:**

- a) El campo de visión (FOV) de la imagen.
- b) El solapamiento (Aliasing) en la imagen.
- c) La resolución de la imagen.
- d) Ancho de banda (bandwidth) del receptor.
- e) Lo desconozco.

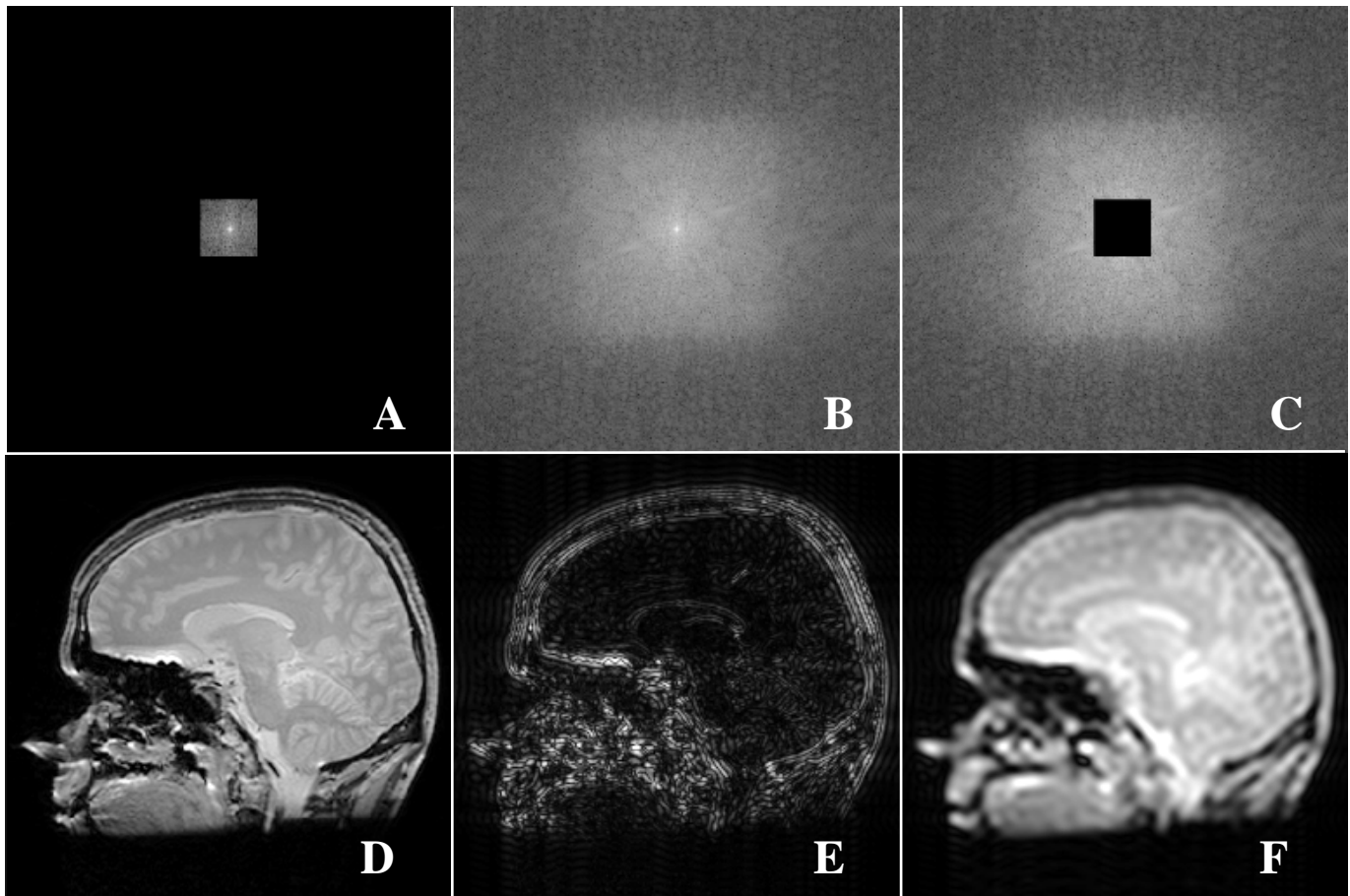


Figura 1

14.-En una imagen ponderada en T2, los tejidos con largos tiempos de relajación (T2) aparecen \_\_\_\_\_.

- a) Hiperintensos (brillantes)
- b) Hipointensos (oscuros)
- c) Isointensos (similar al resto)
- d) El tiempo T2 es siempre el mismo en todos los tejidos, luego no ha lugar a la pregunta
- e) Lo desconozco.

15.- En una imagen de Spin Echo, ¿cómo deben ser los tiempos TE y TR, según los valores orientativos de la tabla, para producir una imagen ponderada en T1?:

TR Largo	→	>1500 ms
TR Corto	→	250-700 ms
TE Largo	→	>60 ms
TE Corto	→	10-25 ms

- a) TE corto, TR corto
- b) TE corto, TR largo
- c) TE largo, TR corto
- d) TE largo, TR largo
- e) Lo desconozco.

16.- ¿Qué efecto en la imagen produce el artefacto llamado “spike”?

- a) Un efecto ondulado en los límites de alto contraste.
- b) Una banda ruidosa en el centro de la imagen.
- c) Rayas claras y oscuras a través de la imagen.
- d) Ninguna de las anteriores.
- e) Lo desconozco.

17.- ¿Cuáles de las siguientes opciones son soluciones para el aliasing?

- a) Aumentar el FOV para abarcar la dimensión anatómica completa del sujeto en la dirección del aliasing.
- b) Usar No Phase Wrap (NPW), que utiliza filtros digitales para cortar la señal que está fuera de los bordes del FOV
- c) Las dos anteriores
- d) Aumentar el parámetro NEX para robustecer la adquisición.
- e) Lo desconozco

18.- Supongamos que L>R significa “de izquierda a derecha”, A>P “de anterior a posterior” e I>S “de inferior a superior”. En un corte axial, ¿cuál sería la opción posible para las direcciones de fase y frecuencia?

- a) Fase L>R; frecuencia A>P
- b) Fase A>P; frecuencia I>S
- c) Fase I>S; frecuencia L>R
- d) Ninguna de las anteriores.
- e) Lo desconozco

19.- Manteniendo el resto de parámetros constantes, indique qué efecto tendría un aumento del parámetro NEX sobre la relación señal ruido (SNR):

- a) La SNR se incrementaría (mejoraría).
- b) La SNR permanecería igual.
- c) La SNR se reduciría (empejaría).
- d) La SNR no puede modificarse externamente.
- e) Lo desconozco.

20.- Manteniendo el resto de parámetros constantes, indique qué efecto tendría un intercambio de las direcciones de fase y frecuencia sobre la relación señal ruido (SNR):

- a) La SNR incrementaría (mejoraría).
- b) La SNR permanecería igual.
- c) La SNR reduciría (empejaría).
- d) La SNR no puede modificarse externamente.
- e) Lo desconozco.





*Apéndice E*

# MANUAL DEL SIMULADOR

---

# GUÍA DE USUARIO

# SimuMRI

Mayo, 2019



---

## Hoja de revisión

Revisión.	Fecha	Descripción
Rev. 0	20/04/19	Creación de la guía rápida
Rev. 1	24/04/19	Cambio en el formato y corrección de errores
Rev. 2	16/05/19	Ampliación de la guía, ampliación de primera adquisición, guía de cómo usar el tutor inteligente y guía de cómo crear un ejercicio con el tutor.

---

# Guía de Usuario

## CONTENIDO

	<u>Página #</u>
<b>1.0 INFORMACIÓN GENERAL.....</b>	<b>1-1</b>
<b>1.1 Descripción de SimuMRI.....</b>	<b>1-1</b>
<b>1.2 Acrónimos y abreviaciones utilizados en la.....</b>	<b>1-1</b>
<b>2.0 INFORMACIÓN GENERAL.....</b>	<b>2-1</b>
<b>2.1 Modelo cliente-servidor en SimuMRI.....</b>	<b>2-1</b>
<b>2.2 Elementos de la interfaz de consola de resonancia .....</b>	<b>2-2</b>
a) Barra lateral .....	2-3
b) Panel de Administrador .....	2-3
c) Información del Paciente .....	2-4
d) Posición del Paciente .....	2-4
e) Parámetros de la imagen.....	2-4
f) Visualizador.....	2-4
g) MR Manager.....	2-4
h) Tiempos del Escáner.....	2-5
i) Parámetros adicionales .....	2-5
j) Tiempos de adquisición .....	2-5
k) Rango del Escáner .....	2-5
l) Parámetros de pie de página .....	2-5
<b>3.0 USANDO SIMUMRI.....</b>	<b>3-1</b>
<b>3.1 Acceder a Imagine MRI e iniciar sesión.....</b>	<b>3-1</b>
<b>3.2 Realizar la primera adquisición. ....</b>	<b>3-2</b>
3.2.1 Datos del paciente.....	3-3
3.2.2 Colocación del paciente y bobina. ....	3-3
3.2.3 Selección del protocolo de adquisición.....	3-4
3.2.4 Protocolo y adquisición del survey. ....	3-5
3.2.5 Adquisición de una secuencia Spin Echo. ....	3-6
<b>3.3 Otros paneles de la consola .....</b>	<b>3-12</b>
3.3.1 Panel de configuración.....	3-13
3.3.2 Selector de patologías. ....	3-14
3.3.3 Editor y generador de protocolos y secuencias. ....	3-15
3.3.4 Otros paneles y características. ....	3-15
<b>4.0 USANDO el ITS.....</b>	<b>4-1</b>
<b>4.1 ¿Qué es sistema de tutoría inteligente?.....</b>	<b>4-1</b>
<b>4.2 Cómo acceder y cómo utilizarlo. ....</b>	<b>4-1</b>
<b>5.0 FAQ.....</b>	<b>5-1</b>
<b>5.1 ¿Qué hacer cuando no aparecen los protocolos? .....</b>	<b>5-1</b>
<b>5.2 ¿Por qué el protocolo que acabo de crear no aparece? .....</b>	<b>5-1</b>

## **1.0 INFORMACIÓN GENERAL**

## **1.0 INFORMACIÓN GENERAL**

### **1.1 Descripción de SimuMRI**

En una unidad de RM, uno de los elementos más importantes es su sistema informático que está constituido por el ordenador (a veces referido como CPU), el monitor de la sala de exploración y la consola de post procesado y visualización.

SimuMRI es actualmente una herramienta docente que se centra en emular este sistema de una forma amigable. Como veremos a lo largo de nuestra guía rápida, la principal aplicación de SimuMRI es una consola de adquisición de MRI que contiene una gran variedad de parámetros. Con ayuda de un teclado, un ratón y una conexión Web, podremos entender y familiarizarnos con el manejo de una máquina RM. Nuestra herramienta cuenta con una GUI preparada para ser ejecutada en cualquier navegador (aunque el navegador aconsejado es Firefox) y un potente servidor donde se realizarán todos los cálculos necesarios para la simulación de las adquisiciones.

### **1.2 Acrónimos y abreviaciones utilizados en la**

RM – Resonancia magnética.

CPU – Unidad central de procesamiento (del inglés Central Processing Unit).

MRI – Imagen por resonancia magnética (del inglés Magnetic Resonance Imaging).

GUI – Interfaz gráfica de usuario (del inglés Graphical User Interface).

FAQ – Preguntas frecuentes (del inglés Frequently Asked Questions).

SAR – Tasa de absorción específica (del inglés Specific Absorption Rate).

NEX – Número de excitaciones (del inglés Number of Excitations).

FOV – Campo de visión (del inglés Field Of View).

RIS – Sistema de información radiológica (del inglés Radiology Information System).

ARM – Angiografía de resonancia magnética.

CW / CCW – En el sentido del reloj (Clockwise) / en contra del sentido del reloj (Counterclockwise).

DICOM – Es un estándar para imagen médica (Digital Imaging and Communication On Medicine).

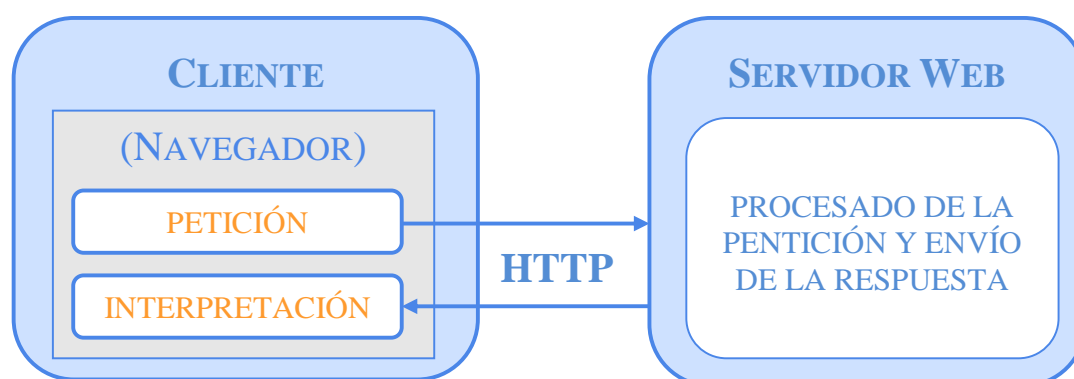
DWI – Imágenes ponderadas en difusión o Diffusion-weighted imaging en sus siglas en inglés.

## **2.0 DESCRIPCIÓN DEL SISTEMA**

## 2.0 INFORMACIÓN GENERAL

### 2.1 Modelo cliente-servidor en SimuMRI

SimuMRI es una aplicación web que sigue el modelo de cliente servidor, donde el servidor es el que se encarga de proveer cualquier tipo de recurso o servicio, mientras que el cliente se encarga de realizar estas peticiones.



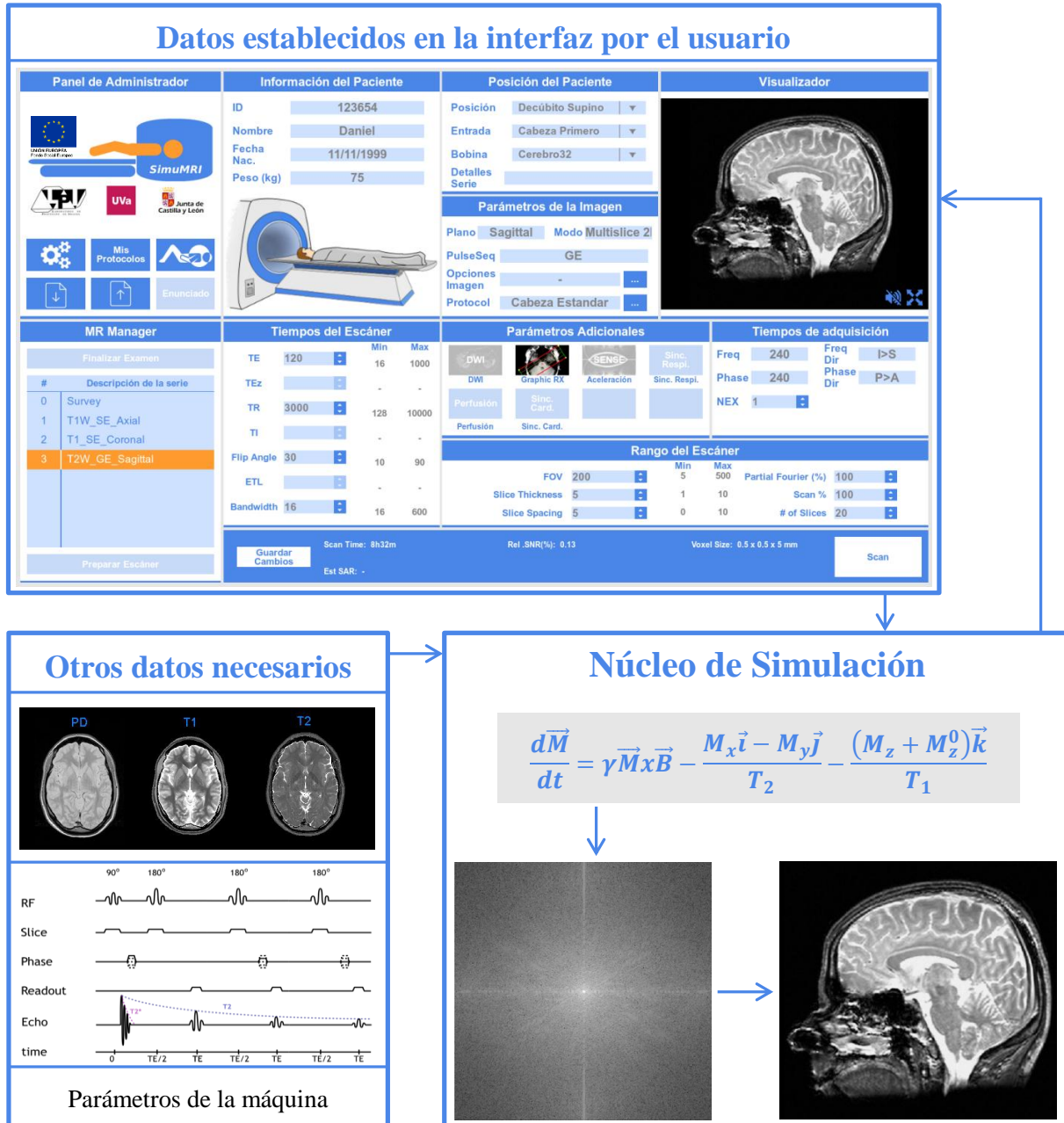
Por tanto, el cliente es ejecutado en el navegador web del PC del usuario, el cual, a través de la interfaz gráfica realizará peticiones al servidor. El navegador aconsejado para su uso es Firefox, pero en principio es soportado por la mayoría de los navegadores actuales. Más adelante veremos con qué elementos cuenta la interfaz gráfica, pero su característica principal es la emulación de una consola de resonancia magnética y la visualización de las imágenes generadas.

Por otro lado, el servidor atiende diversas peticiones hechas por el cliente, como la identificación del usuario, los protocolos disponibles, etc. Pero su principal tarea es la realización de la simulación. El siguiente esquema muestra brevemente cómo realiza la simulación.

Para ello es necesaria una serie de datos que contengan las propiedades magnéticas de los tejidos y que llamaremos modelo. Esencialmente se compone por: la densidad de protones (DP), el tiempo de recuperación de la componente longitudinal (T1) y el tiempo de pérdida de la componente trasversal (T2). Si a esto le añadimos los parámetros intrínsecos de la máquina (los que vienen definidos por sus características “físicas” o los que vienen prefijados) y los extrínsecos (que vendrían controlados por el usuario), podemos obtener una imagen de resonancia magnética con cierta similitud a las reales. Debido a su carácter educativo y a que es un simulador, existen otros parámetros que son interesantes forzar o anular, como la aparición de artefactos y que se activarían o desactivarían también desde la interfaz.

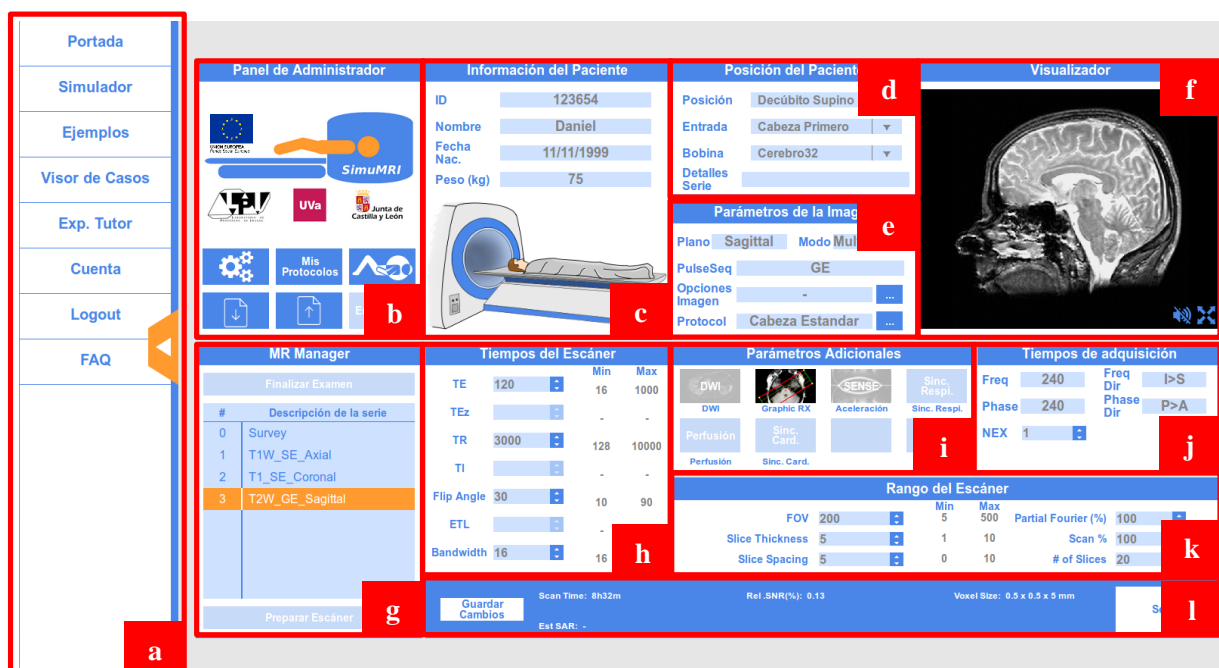
En resumen, al realizar la simulación tomamos todos los parámetros fijados en la interfaz así como la información necesaria que almacenamos en el servidor, todos estos datos los enviamos al núcleo de simulación, que haciendo uso de expresiones sacadas de las ecuaciones de Bloch, obtiene una aproximación al espacio-k de la imagen simulada y finalmente la imagen simulada que es retornada a la interfaz para su visualización. Esta división de tareas permite que nuestro sistema pueda correr en casi cualquier ordenador con un navegador web y acceso a internet.





## 2.2 Elementos de la interfaz de consola de resonancia

La figura muestra la interfaz de usuario, en los siguientes sub apartados pasarán a enumerarse cada uno de los elementos que la componen así como su finalidad.



### a) Barra lateral

La barra lateral permite navegar por las diferentes aplicaciones que compone SimuMRI y que veremos a lo largo de esta guía en más detalle:

- **Portada:** Es la presentación de la aplicación.
- **Simulador:** Se corresponde con la consola de adquisición de resonancia.
- **Ejemplos:** Son pequeños ejecutables web. Tratan conceptos físicos de forma simplificada con fines docentes.
- **Visor de Casos:** Es un panel que cuenta con diversos casos de estudios reales, permitiendo tanto su análisis, su visualización en un visor DICOM y su descarga en caso de ser necesaria.
- **Exp. Tutor:** Es un panel que cuenta con diversos ejercicios para la consola de adquisición de resonancia. Utiliza un sistema de tutoría inteligente destinado a guiar al alumno por el proceso y seguir su aprendizaje.
- **Cuenta:** Panel que da acceso a los datos del usuario.
- **Logout:** Cierra la sesión del usuario.
- **FAQ:** Preguntas frecuentes.

### b) Panel de Administrador

El panel de administrador está compuesto por diversos botones que permitirán la configuración de: las características de la simulación, la creación y edición de nuevos protocolos y secuencias, la elección del volumen que vamos a utilizar y otras opciones en función del modo en el que se encuentre.

### c) Información del Paciente

El panel de información del paciente está destinado a la introducción de datos importantes para el complejo sanitario. Aunque cuenta con varios parámetros, el más destacable es el Peso del paciente dado que será el que determine el SAR.

La parte inferior del panel muestra un esbozo donde aparece la máquina de resonancia magnética y la posición del paciente. Este esquema varía en función de las elecciones del panel *d*.

### d) Posición del Paciente

El panel de posición del paciente se utiliza para definir cómo es situado el paciente, es decir, el área principal de entrada del paciente en el túnel del imán para su correcto posicionamiento y orientación. Este, además, también es utilizado para la selección de la bobina que se le ha colocado al paciente.

### e) Parámetros de la imagen

El panel de parámetros de la imagen permite la elección del protocolo para el estudio así como las opciones de imagen que van a ser usadas en cada secuencia. A la derecha de tanto las opciones de imagen como del protocolo, aparece un botón azul con tres puntos en su interior, este es el interactor para abrir sus respectivos paneles.

Además, en él se muestra información relevante de la adquisición, como el plano, el modo o la secuencia que va a utilizarse.

### f) Visualizador

El visualizador es el encargado de mostrar la imagen una vez adquirida para realizar una rápida inspección. La inspección se lleva a cabo mediante la utilización del ratón en el interior del panel:

- **Clic izquierdo pulsado y desplazamiento horizontal:** Aumenta o reduce el brillo.
- **Clic izquierdo pulsado y desplazamiento vertical:** Aumenta o reduce los límites de intensidad de la imagen.
- **Uso de la ruleta:** Muestra los diferentes cortes adquiridos.
- **Clic derecho pulsado y desplazamiento vertical u horizontal:** Permite alejar o acercar la imagen.
- **Doble clic izquierdo encima de la imagen:** Retorna la imagen a sus características iniciales.

Además, en la esquina inferior derecha, dispone de dos iconos:

- **Altavoz:** Permite activar el ruido de la resonancia.
- **Maximizar:** Abre un panel en el que puede estudiarse la adquisición con mayor detalle.

### g) MR Manager

El panel MR Manager muestra todas las secuencias de pulso de las que dispone el protocolo cargado, permitiendo su selección y su configuración. Por otro lado, podremos ejecutar la calibración del estudio previa a su adquisición, obtener el survey y finalizar el examen. Hasta el momento, dicha calibración se

realiza automáticamente durante la adquisición del survey y la finalización del examen también es automática

### h) Tiempos del Escáner

El panel de tiempos del escáner permite la selección, por parte del operador de los parámetros que configuran el contraste de la imagen en RM. Por ejemplo, estos parámetros pueden variarse para la optimización de adquisiciones ponderadas en T1 o T2.

### i) Parámetros adicionales

El panel de parámetros adicionales está formado por diversos botones que dan acceso a sus respectivos paneles. Cada panel es utilizado para configurar opciones específicas de la secuencia de pulsos u otras características de la adquisición como la planificación de los cortes.

### j) Tiempos de adquisición

El panel de tiempos de adquisición contiene parámetros que influyen la duración del estudio, como la resolución espacial o el NEX.

### k) Rango del Escáner

El panel de rango del escáner da acceso a algunos de los parámetros que se modifican durante la planificación. Contiene el grosor y la separación de los cortes, el FOV en la dirección de los cortes, el número de cortes y otros parámetros más complejos que permiten la realización de adquisiciones parciales.

### l) Parámetros de pie de página

El panel de parámetros de pie de página contiene información relevante sobre la adquisición:

- **Scan Time** (Tiempo de escáner): Es el tiempo que durará la adquisición de la secuencia de pulsos actual en horas, minutos y segundos. Variará en función de los parámetros elegidos.
- **Relación Señal/Ruido**: Proporciona información sobre la calidad de la imagen con respecto a los parámetros iniciales de la secuencia.
- **Voxel size** (tamaño del vóxel): Indica el tamaño en milímetros de cada vóxel de la imagen.
- **Est. SAR** (estimación del SAR): Parámetro de seguridad del escáner (**En desarrollo**).

Además, cuenta con dos botones:

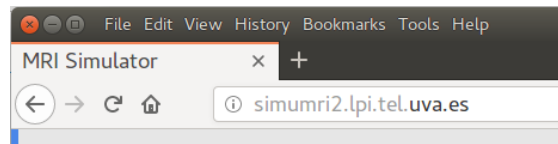
- **Scan**: Una vez seleccionados todos los parámetros de la secuencia de pulsos, el botón scan da comienzo la adquisición.
- **Guardar cambios**: Permite guardar los cambios realizados en las secuencias. Sólo se permite la modificación en las secuencias que cuentas con permisos pertinentes (generalmente las que has creado).

## **3.0 Usando SimuMRI**

## 3.0 USANDO SIMUMRI

### 3.1 Acceder a Imagen MRI e iniciar sesión

El primer paso será, en un navegador, introducir la dirección <http://simumri2.lpi.tel.uva.es> y acceder a la página principal.



Una vez finalizada la carga, la página web presentará el siguiente aspecto:



Para poder acceder a todas las características de SimuMRI es necesario registrarse. Es necesario o bien pulsar en la opción "Login" o bien pulsar encima de la imagen de SimuMRI. Esto le redireccionará al siguiente panel:

A screenshot of the SimuMRI login form. The form has a blue header with the text "Inicio de sesión". Below the header are two input fields: "Nombre de usuario" with the value "DanielT" and "Contraseña" with a masked password of ten dots. At the bottom of the form is a blue button labeled "Enviar".

Aquí debe introducirse el usuario y la clave proporcionada. Una vez registrado podrá acceder a todas las características vistas en el apartado 2.0.

## 3.2 Realizar la primera adquisición.

En este punto, contamos con el acceso a la consola de adquisiciones de una máquina de RM como la que se muestra a continuación.

Panel de Administrador	Información del Paciente	Posición del Paciente	Visualizador																								
 	ID: _____ Nombre: _____ Fecha Nac.: _____ Peso (kg): _____	Posición: _____ Entrada: _____ Bobina: _____ Detalles Serie: _____ <b>Parámetros de la Imagen</b> Plano: _____ Modo: _____ PulseSeq: _____ Opciones Imagen: _____ Protocol: _____																									
MR Manager	Tiempos del Escáner	Parámetros Adicionales	Tiempos de adquisición																								
Finalizar Examen # Descripción de la serie Preparar Escáner	<table border="1"> <thead> <tr> <th></th> <th>Min</th> <th>Max</th> </tr> </thead> <tbody> <tr><td>TE</td><td>-</td><td>-</td></tr> <tr><td>TEz</td><td>-</td><td>-</td></tr> <tr><td>TR</td><td>-</td><td>-</td></tr> <tr><td>TI</td><td>-</td><td>-</td></tr> <tr><td>Flip Angle</td><td>-</td><td>-</td></tr> <tr><td>ETL</td><td>-</td><td>-</td></tr> <tr><td>Bandwidth</td><td>-</td><td>-</td></tr> </tbody> </table>		Min	Max	TE	-	-	TEz	-	-	TR	-	-	TI	-	-	Flip Angle	-	-	ETL	-	-	Bandwidth	-	-	DWI, DWI, Perfusion, SENSE, Aceleración, Sinc. Resp., Sinc. Card., Sinc. Card. <b>Rango del Escáner</b> FOV: _____ Min: _____ Max: _____ Partial Fourier (%): _____ Slice Thickness: _____ Scan %: _____ Slice Spacing: _____ # of Slices: _____	Freq: None Freq Dir: _____ Phase: None Phase Dir: _____ NEX: _____
	Min	Max																									
TE	-	-																									
TEz	-	-																									
TR	-	-																									
TI	-	-																									
Flip Angle	-	-																									
ETL	-	-																									
Bandwidth	-	-																									
Guardar Cambios Scan Time: - Est SAR: -		Rel. SNR(%): - Voxel Size: -	Scan																								

### Breve Explicación

Algunos modelos de equipos de MRI, permiten elegir entre dos opciones de parámetros físicos:

- **dB/dt:** se refiere a las fluctuaciones bruscas del campo magnético B0 del imán del aparato, con respecto al tiempo. Estas fluctuaciones bruscas pueden provocar estimulación de los nervios periféricos en las extremidades de la persona que se encuentra en el túnel del imán. (ver parámetros de seguridad y normativas).
- **SAR:** Indica el calentamiento de los tejidos del paciente producida por la emisión de ondas de RF durante una exploración, que depende del peso. (Absorción calórica)

Existen niveles de seguridad relacionados con estas opciones:

- **Sistema Normal (máxima Seguridad):** Todas las exploraciones tienen una intensidad magnética y de ondas de RF tolerables. Estas imágenes podrán ser en algunos casos de peor calidad.
- **Sistema de Primer Nivel:** Utiliza parámetros de adquisición con la dosis mínima de RF e intensidad de campo magnético, adecuados para obtener resultados óptimos. No produce contratiempos, salvo en pacientes con fiebre o problemas cardíacos graves (ver parámetros de seguridad).

En nuestro simulador, se condiciona siempre un estado de seguridad máxima, respetando el SAR como medida de seguridad según los parámetros de seguridad. Siempre, en cualquier máquina, es importante seguir las recomendaciones del fabricante.

### 3.2.1 Datos del paciente

Información del Paciente	
ID	<input type="text"/>
Nombre	<input type="text"/>
Fecha Nac.	<input type="text"/>
Peso (kg)	<input type="text"/>

El primer paso para comenzar la adquisición de un nuevo estudio de resonancia es rellenar los datos del paciente. Muchos equipos de resonancia incorporan en esta ventana más datos y en general dependen del fabricante y de la institución de sus propios sistemas RIS o WorkList-Dicom de sus agendas de trabajo. Estos parámetros son comúnmente:

- Número de acceso: que normalmente están formado por el año en el que se realiza la exploración, el aparato y el número de orden del examen.
- **ID:** Identificador único del paciente.
- **Nombre:** Nombre del paciente.
- **Fecha de nacimiento:** Fecha de nacimiento del paciente.
- Edad: Edad del paciente.
- Sexo: Sexo del paciente.
- **Peso y Talla:** Peso del paciente en kilogramos.
- Radiólogo responsable: El radiólogo encargado.
- Operador a cargo del estudio: Operador que realiza la adquisición.
- Descripción del examen: Descripción del examen que va a realizarse.
- Historia clínica: Historia clínica del paciente.

En nuestro simulador, consideramos importante que se adquiriera el hábito de introducir estos datos, por ello hemos seleccionado algunos de los que consideramos más relevantes, de forma que introducir esta información sea una tarea rápida. Sería suficiente con introducir los datos marcados en azul. Por ejemplo, podría introducirse la ID *123456*, el nombre *Daniel*, la fecha *01/02/1999* y el peso *75*.

### 3.2.2 Colocación del paciente y bobina.

Posición del Paciente	
Posición	<input type="text"/> ▼
Entrada	<input type="text"/> ▼
Bobina	<input type="text"/> ▼
Detalles Serie	<input type="text"/>

Ahora toca determinar la posición en la que se ha situado al paciente sobre la mesa exploratoria, así como el sentido en el que se introduce al paciente dentro de la máquina. Ambos dependen del tipo de exploración.

A continuación se enumeran las diferentes opciones con las que cuenta este panel.

**Posición:** Existen cuatro posibilidades:

1. Decúbito supino (Supine): Es la más frecuente.
2. Decúbito prono (Prone): Para exploraciones de la mama.
3. Decúbito derecho (Right Decub): Poco utilizada.
4. Decúbito izquierdo (Left Decub): Poco utilizada.



**Entrada:** Indica la posición de entrada en el túnel del imán del aparato:

1. Primero la Cabeza (Head First): Cráneo-encefálicas y de la columna.
2. Primero los Pies (Feet First): Tórax, mama.

**Bobina (coil) o antena:** Hay que detallar el nombre de la antena de superficie que se ha colocado sobre la estructura anatómica que se va a explorar. (Todas las antenas disponibles en el aparato aparecen ordenadas en la ventana (Bobinas)).

**Detalles de Series (Series Description):** En esta casilla se describe detalles la secuencia de pulsos actual y se rellena al seleccionar una secuencia específica.

En conclusión, en este panel es necesario fijar la posición y la entrada del paciente. Por ejemplo, para una adquisición de cerebro, estas serían *Decúbito Supino* y *Cabeza Primero*.

El siguiente paso consiste en seleccionar la bobina. Debe coincidir con la que el equipo tenga en la sala de adquisición de resonancia, la misma que hemos determinado que se usará para el paciente en la exploración determinada, en este caso *Cerebro32*. Muchas casas comerciales tienen sus propios nombres de antena, nosotros determinamos la antena por la parte anatómica y el número de canales de la misma. Siguiendo con el ejemplo, el nombre de la bobina utilizada determina que es para cerebro y que tiene 32 canales.

### 3.2.3 Selección del protocolo de adquisición.



Para acceder al panel de protocolos, debe hacerse clic en el botón situado en la parte inferior derecha del panel “Parámetros de la imagen” (marcado con un recuadro rojo).

El panel de protocolos cuenta con un sistema de clasificación (un avatar en otros sistemas dividida en segmentos) por categorías. Cada una de ellas da acceso a los protocolos propios de su estructura anatómica.



Al seleccionar la parte anatómica, aparecerá un listado de protocolos referidos a esa localización. Las partes anatómicas son:

- Cabeza
- Cuello
- Tórax
- Cardio
- EESS
- Abdomen
- Columna
- Pelvis
- EEII
- Otros

Una vez seleccionado el protocolo (en este caso *TestSequences*), pulsaremos “Aceptar”.

**Nota:** SimuMRI cuenta con un espacio reservado para guardar los protocolos personalizados como veremos más adelante.

### Definición de Listados de Protocolos de Adquisición

Muchas casas comerciales tienen distribuidos en categorías los protocolos definidos por el fabricante. En ellos, si seleccionamos la parte anatómica, se abrirá una ventana de selección, y seleccionado una parte anatómica, aparecen los protocolos definidos, con las secuencias que cada protocolo tiene seleccionadas por el fabricante. Existe también la posibilidad de crear tus propios protocolos o protocolos de centros, ya optimizados con los parámetros de cada centro.

### 3.2.4 Protocolo y adquisición del survey.

Cada protocolo está compuesto por una serie de secuencias de pulsos. Todas son cargadas en el panel RM Manager al pulsar “Aceptar” en el panel “Protocolo”. Para seleccionar cualquiera de ellas, es necesario pulsar con clic izquierdo del ratón encima del nombre



MR Manager	
Finalizar Examen	
#	Descripción de la serie
0	Survey
1	SE_Axial
2	FSE_Axial
3	FLAIR_Axial
4	GE_Axial
5	SPGR_Axial
6	SSFP_FID_Axial
7	SSFP_ECHO_Axial

Preparar Escáner

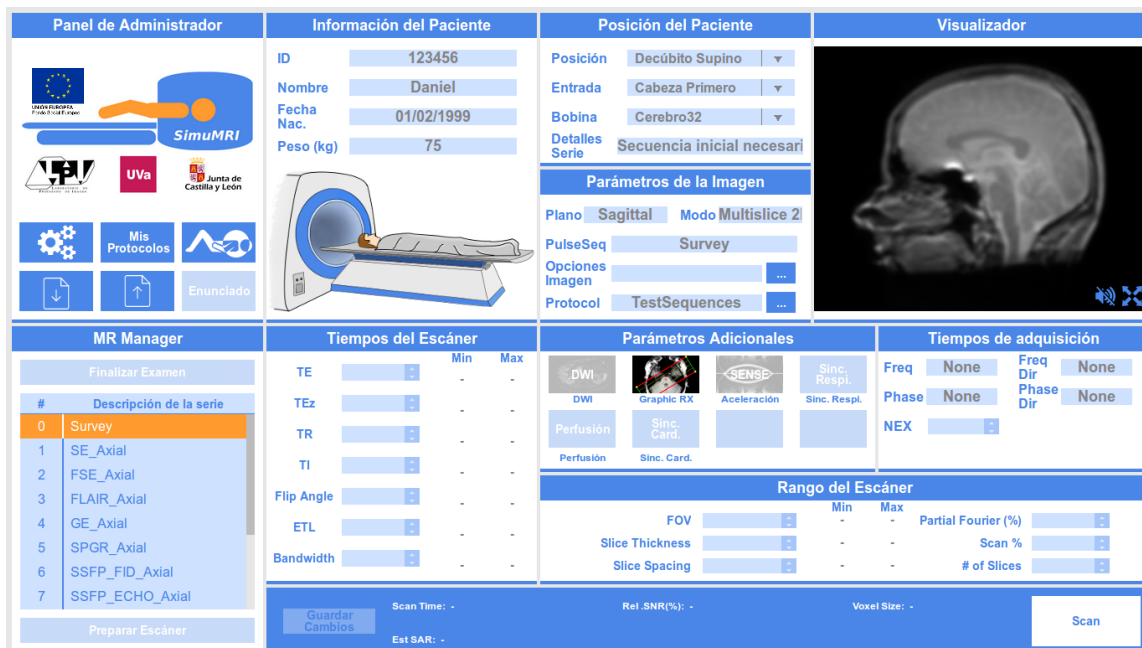
Las secuencias cargadas pueden visualizarse en esta imagen. En algunos casos, cuando la lista de secuencias es mayor que el espacio reservado para este panel, debe usarse la ruleta del ratón para acceder a las últimas.

#### Listado de secuencias:

- Secuencia de Espín Eco Clásica (ESTÁNDAR)
- Secuencias de Eco de Gradiente (GRADIENT ECHO)
- Secuencias de Espín Eco, con Tren de Ecos. (FSE)
- Métodos específicos para realizar exploraciones cardiovasculares (VASCULAR).
- Secuencias de Eco Plano. EPI. (ECHO PLANAR)

En este punto, lo primero será adquirir el “Survey”, para ello lo seleccionamos y pulsamos el botón “Scan” de la parte inferior derecha de la consola. La apariencia actual del simulador será

parecida a la siguiente imagen.



### PreScan

En las máquinas comerciales, antes de activar la secuencia hay que realizar el PRESCAN que es un procedimiento que sirve para calcular la frecuencia central de precesión de los protones y las ganancias analógicas y digitales que determinan la capacidad de la antena receptora.

Las opciones son las siguientes:

- **Auto Prescan:** Se realiza de manera automática. (en nuestro simulador entendemos que existe una calibración central adecuada de las frecuencias centrales de precesión).
- **Manual Prescan:** Se realiza manualmente por el Técnico sobre la ventana apropiada. Su desarrollo se llevará a cabo a largo del curso, definiendo la banda y la velocidad de precesión de forma manual.
- **Prep Scan:** Preparación para el disparo de la secuencia.
- **Scan:** Activación de la secuencia

### 3.2.5 Adquisición de una secuencia Spin Echo.

Ahora seleccionamos la siguiente secuencia llamada “SE\_Axial”. Inmediatamente cambiarán diversos paneles del simulador.



### PARÁMETROS DE LA IMAGEN

El panel cuenta con los siguientes parámetros

**Plano (Plane):** Orientación del Plano de corte que cuenta con las siguientes opciones:

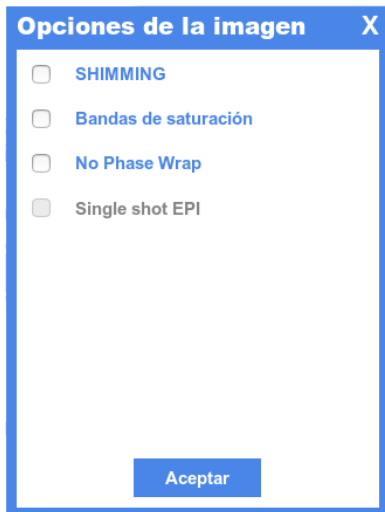
- Sagittal (Sagital)
- Axial
- Coronal
- Oblique (Oblicuo)

Para nuestro caso, sería oblicuo.

**Pulse Sequence:** Tipo de secuencia de pulsos elegida: de Espín Eco (SE), Echo de Gradiente (GRE), etc.

**Mode:** El modo de adquisición de las imágenes.

- 2D
- 3D
- Cine ARM
- 4D



### OPCIONES DE IMAGEN

A este panel se accede de forma similar que al de protocolos, sólo es necesario pulsar sobre el botón situado a la derecha de “Opciones de Imagen” en el panel “Parámetros de la Imagen”. El panel incluye todas las opciones de imagen que dispone el aparato.

En algunas casas comerciales, varían de nombre, pero las funciones principales y los principios físicos en las que se basan son semejantes en la mayor parte de los fabricantes

Tiempos del Escáner			
		Min	Max
TE	15	1.1	20
TEz		-	-
TR	500	15.6	600
TI		-	-
Flip Angle		-	-
ETL		-	-
Bandwidth	128	16	128

### PARÁMETROS FÍSICOS (SCAN TIMING)

En este panel aparecen los parámetros que se utilizan cuya influencia determina el contraste y el tiempo de exploración de las secuencias. Estos parámetros estarán disponibles en función de la secuencia de pulsos determinada. En nuestro ejemplo serían el TE o Tiempo de Eco, el TR o Tiempo de Repetición y el ancho de banda o Bandwidth.

Tiempos de adquisición			
Freq	160	Freq Dir	L>R
Phase	160	Phase Dir	P>A
NEX	1		

## PARÁMETROS GENERALES DE ADQUISICIÓN

**Freq (Frecuencia):** Indica el número de filas de la Matriz.

**Phase (Fase):** Número de columnas de la Matriz.

**Nex (Número de excitaciones):** Influye en la calidad y en la duración.

**Phase Dir:** Indica la dirección y sentido en que se produce la codificación de los datos según la fase.

**Freq DIR:** Indica la dirección y sentido en que se produce la codificación de los datos según la frecuencia.

### Notas y observaciones

En las máquinas comerciales pueden existir otros parámetros como o algunos pueden incluir otras características:

**Phase FOV:** Indica porcentualmente la reducción del FOV en el sentido de la codificación de fase, valores habituales serían el 100%, el 75% o el 50%.

**Freq DIR:** En los visores, esta dirección aparece en la zona inferior izquierda con la letra (V ^) y una cabeza de flecha que indica el sentido.

**Velocity Encoding:** Hay tres posibilidades

- A/P: Anteroposterior
- S/I: Superoinferior
- R/L: Derecha Izquierda

**Auto Center Freq:** Sincronización automática con el pico de la frecuencia central de precesión de los protones del agua (en el simulador lo centramos en H2O).

**Contrast amount:** La cantidad de contraste que se va a administrar.

**Agent:** elemento que compone la macromolécula de contraste.

## RANGO DE ADQUISICIÓN O RANGO DE EXPLORACIÓN

Rango del Escáner					
		Min	Max		
FOV	200	100	500	Partial Fourier (%)	100
Slice Thickness	5	5	5	Scan %	100
Slice Spacing	5	5	5	# of Slices	20

**FOV:** Indica el tamaño del FOV en la dirección de los cortes, es decir, en la imagen sería 200 que es el resultado de multiplicar el número de cortes por el grosor de los cortes mas el espacio entre cortes.

**Slice Thickness (Grosor de corte):** Detalla el grosor de cada corte.

**Spacing (Separación):** Se refiere a la separación que debe existir entre corte y corte para que no se produzca degradación de las imágenes.

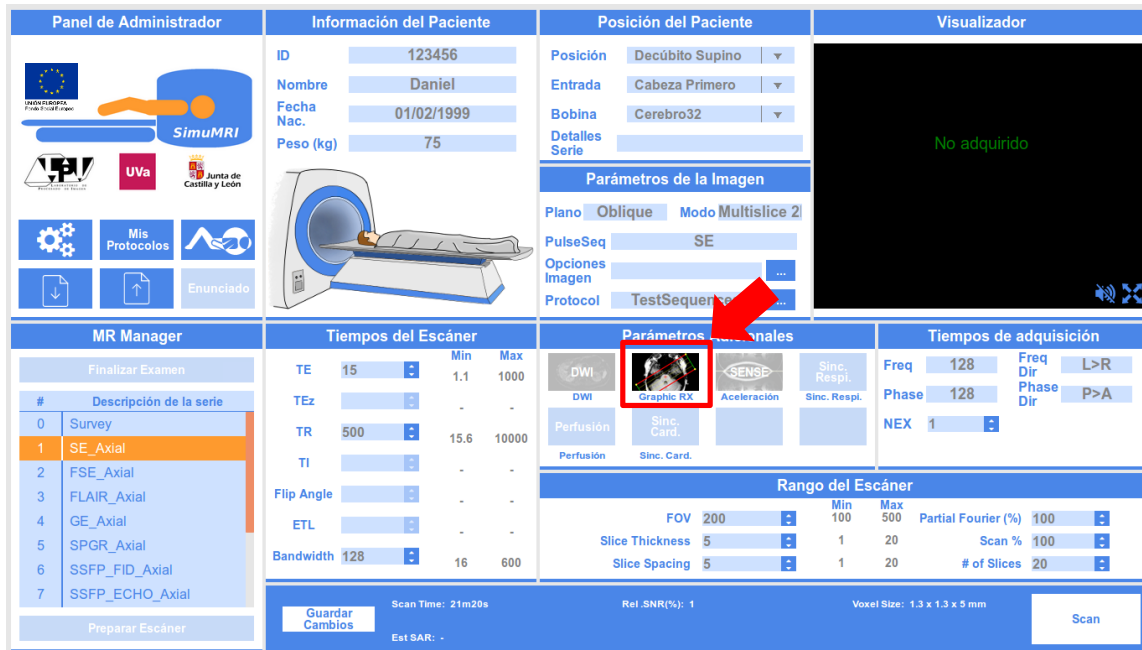
**Slices (Cortes):** Indica el número de cortes prescritos por el Técnico, en cada adquisición.

**Partial Fourier:** Permite una adquisición parcial del espacio k que posteriormente es rellenado. Para ello asume que la imagen es real y por tanto el espacio k es su complejo conjugado.

**SCAN%:** Adquiere sólo la parte central del espacio k.

## PLANIFICACIÓN 2D

En este punto, la consola de simulación debería presentar un aspecto similar a la imagen que se muestra a continuación. Hasta ahora hemos hecho un pequeño repaso de varios de los parámetros disponibles en una secuencia de RM. Manteniendo los valores que vienen por defecto es el momento de realizar la planificación o prescripción de la adquisición 2D.

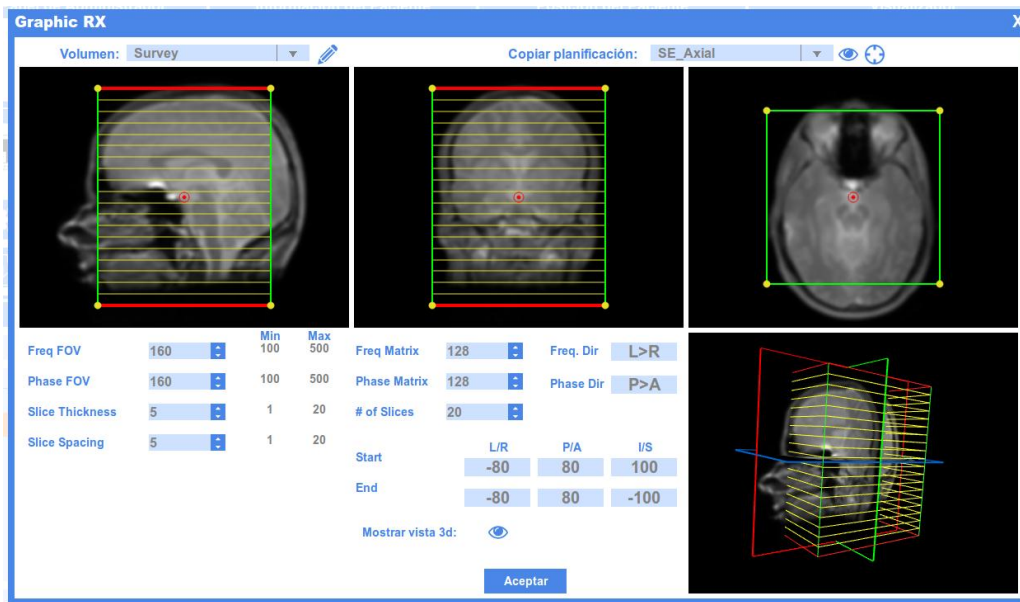


Para abrir la planificación sólo hay que pulsar en el botón “Graphic RX” situado en el panel de Parámetros Adicionales. A continuación se mostrará un panel como el siguiente. Este se compone de diversos elementos enumerados a continuación.

**Volumen:** Es un desplegable que permite elegir entre todos los distintos volúmenes (serie de cortes) adquiridos hasta ahora para llevar a cabo la planificación. A la derecha del desplegable, se sitúa un icono con forma de lápiz. Al hacer clic sobre ello permite cambiar el nombre del volumen.

**Copiar planificación:** Permite copiar cualquiera de las planificaciones utilizadas hasta el momento para la nueva adquisición. El nombre de la planificación viene dado por la secuencia donde ha sido utilizada. A su derecha nuevamente, se observan 2 iconos, uno con forma de ojo y otro con forma de mirilla. El primero sirve para ocultar o mostrar la planificación, mientras que el segundo centra automáticamente la

planificación en el centro del volumen. Este último botón es especialmente útil cuando no se ha realizado la planificación en una adquisición y no conocemos dónde está la zona de la planificación.



**Visores:** Se encargan de mostrar tanto las adquisiciones como el planificador y contiene las mismas interacciones que el visualizador normal. Además y dado a la existencia del planificador, es necesaria la aparición de nuevos interactores:

- **Rotar sobre los ejes del plano (puntos amarillos):** al situar el ratón sobre estos puntos el cursor del ratón cambia y al hacer clic permite la rotación del planificador. Esta rotación usa como eje la vertical al plano actual, es decir, si se rota 90 grados sobre la vista axial en la imagen, la dirección de fase y de frecuencia cambiarán. Si la rotación fuese ahora de 90 grados sobre el plano sagita, el corte pasaría a ser coronal en vez de axial.
- **Aumentar o reducir el número de cortes (líneas rojas):** Al situarse sobre una línea roja y pulsar clic, permite la reducción del número de cortes.
- **Desplazar el planificador (círculos rojos concéntricos):** Al pulsar sobre este interactor se permite el desplazamiento de la planificación sobre el plano.

**Mostrar vista 3d:** Este botón habilita la aparición de un nuevo visor tridimensional. Este no existe en las máquinas comerciales y tiene una finalidad meramente educativa. Su objetivo es ayudar a comprender qué sucede cuando desplazamos o rotamos el planificador. En el nuevo visor los interactores son algo distintos:

- **Zoom:** La ruleta del ratón permite acercar o alejar la escena. Mantener pulsado el clic derecho y moverlo horizontal o verticalmente tendrá el mismo efecto.
- **Rotar:** Mantener el clic izquierdo pulsado mientras el cursor se desplaza permite rotar la escena.
- **Desplazar:** Mantener la ruleta pulsada mientras se desplaza el cursor permite desplazar la escena.
- **Actualizar:** En algunas ocasiones alguno de los cortes parece desaparecer, actualizar la escena pulsando doble clic izquierdo debería resolverlo.

Por defecto esta opción aparece desactivada.



**Parámetros de la planificación:** Contiene todos los parámetros relevantes para establecer las dimensiones de la imagen y del campo de visión (FOV) que desea adquirirse.

**Notas y observaciones**

**Volumen:** En algunos sistemas es posible que cada uno de los visualizadores muestren cortes independiente y planificar así sobre ellos. Nuestro simulador permite trabajar sobre un conjunto de cortes a la vez. Además cada uno de los visores es utilizado únicamente para una vista, de izquierda a derecha serían, sagital, coronal y axial. En los cortes de menor resolución lo que se hace es una interpolación para rellenar los espacios.

**Tipos de adquisiciones:** Existen diferentes formas de realizar la prescripción de la adquisición que no están incluidas en la consola como podrían ser la radial 2D o la adquisición 3D. En la adquisición radial 2d la planificación pasa a tener forma cilíndrica y los cortes se corresponderían con los radios de las bases circulares. En este caso el sentido de adquisición sería la dirección radial y podría ser o CW o CCW (Clock Wise) o (Counter Clock Wise), además habría que añadir la separación parcial entre los radios. En la prescripción de una adquisición 3D lo que se pasa a manipularse son bloques o “slabs” de forma que ahora se establecerá el número de slab y el número de cortes en los que se divide cada slab así como su superposición.

Una vez establecidos todos estos parámetros, pulsaremos el botón “Aceptar” para guardar la planificación.

### ADQUISICIÓN DE LA SECUENCIA

En este momento sólo queda pulsar el botón “Scan” y ver el resultado de la adquisición en el visor. Tras la adquisición la consola debería mostrar un estado similar al siguiente:

The screenshot displays the SimuMRI console interface, organized into several panels:

- Panel de Administrador:** Includes logos for the European Union, SimuMRI, and the Junta de Castilla y León. It features navigation icons for 'Mis Protocolos' and 'Enunciado'.
- Información del Paciente:** Shows patient details: ID 123456, Name Daniel, Birth Date 01/02/1999, and Weight 75 kg. An illustration of a patient in the scanner is also present.
- Posición del Paciente:** Configuration for 'Decúbito Supino', 'Entrada Cabeza Primero', and 'Bobina Cerebro32'.
- Visualizador:** Displays a coronal MRI slice of a brain. A red arrow points to a small blue 'X' icon in the bottom right corner of the image area.
- Parámetros de la Imagen:** Shows 'Plano Oblique' and 'Modo Multislice 2'. Other options include 'PulseSeq SE', 'Opciones Imagen', and 'Protocol TestSequences'.
- MR Manager:** A list of sequences with 'SE\_Axial' selected as sequence 1. Other sequences include FSE\_Axial, FLAIR\_Axial, GE\_Axial, SPGR\_Axial, SSFP\_FID\_Axial, and SSFP\_ECHO\_Axial.
- Tiempos del Escáner:** A table of timing parameters:
 

TE	15	Min	Max
TEz		1.1	1000
TR	500		15.6 10000
TI			
Flip Angle			
ETL			
Bandwidth	128	16	600
- Parámetros Adicionales:** Includes options for 'DWI', 'Sinc. Resp.', 'Perfusión', and 'Sinc. Card.'.
- Tiempos de adquisición:** Shows 'Freq 128', 'Phase 128', and 'NEX 1'. It also includes 'Freq Dir L>R' and 'Phase Dir P>A'.
- Rango del Escáner:** A table of acquisition range parameters:
 

FOV	200	Min 100	Max 500	Partial Fourier (%)	100
Slice Thickness	5	1	20	Scan %	100
Slice Spacing	5	1	20	# of Slices	20
- Footer:** Displays 'Scan Time: 21m20s', 'Rel. SNR(%): 1', 'Voxel Size: 1.3 x 1.3 x 5 mm', and buttons for 'Guardar Cambios', 'Est SAR: -', and 'Scan'.



Como puede verse, el visualizador muestra la adquisición, en este caso con “Foldover” en la dirección de fase (Anterior-Posterior). Para ver los cortes más en detalle podemos abrir el panel de visualización pulsando en el botón señalado en la imagen anterior. Esto dará paso a un panel como el siguiente:



Permite algunas opciones similares al “Graphic RX” como es el cambio de nombre del volumen, la elección de elegir visualizar otras adquisiciones o la orientación de la visualización. Sin embargo, en la parte inferior derecha muestra un icono con forma de fecha, el cual permitiría la descarga del DICOM adquirido.

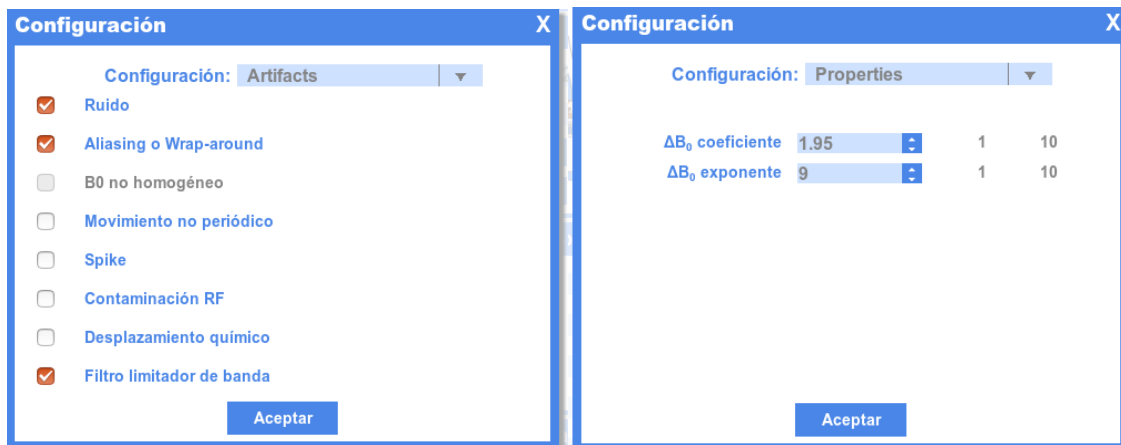
#### Notas y observaciones

**Autoview:** En los equipos de resonancia magnética este visualizador puede llamarse “Autoview”, en él, una vez pasado el tiempo de adquisición (TAD) que puede variar entre unos segundos a varios minutos, se mostrarán secuencialmente las imágenes adquiridas mientras que el técnico las observa. En nuestra consola esta observación debe hacerse manualmente.

### 3.3 Otros paneles de la consola

SimuMRI cuenta con una gran variedad de paneles que no hemos visto. En esta sección veremos algunos de ellos más detenidamente

### 3.3.1 Panel de configuración.



El panel de configuración es uno de los paneles necesarios en nuestra consola debido a su carácter de simulador. En él pueden habilitarse/deshabilitarse o modificar ciertos artefactos y/o propiedades de la simulación. Cuenta con un menú desplegable para navegar por las diferentes opciones. En su estado actual cuenta con dos opciones: artefactos, y propiedades. Su apertura se realiza pulsando el botón que contiene engranajes, situado dentro del panel del administrador.



Los artefactos pueden definirse como un efecto que aparece en la imagen, ya sea como intensidades o como estructuras que no se corresponden o no están presentes en la distribución espacial de los tejidos. Su aparición crea distorsiones en las imágenes, visibles por su mala calidad o por la aparición de elementos que puedan dificultar su interpretación. Aprender a distinguir estos elementos es clave para que estos artefactos no conduzcan a un diagnóstico erróneo.

El simulador cuenta con varios artefactos:

- **Ruido:** Habilita la aparición de ruido sobre las imágenes. Está habilitada por defecto.
- **Aliasing:** Habilita la aparición aliasing o foldover en la dirección de codificación de fase. Está habilitada por defecto.
- **B0 no homogéneo:** El campo B0 presenta inhomogeneidades, ya sea debido a las características físicas de la máquina o la interacción del campo magnético con el tejido. Habilitada por defecto.
- **Movimiento no periódico:** Simula el movimiento voluntario de un paciente en la máquina. Deshabilitada por defecto.
- **Spike:** Simula el artefacto spike que produce líneas en la imagen. Está causado por la escritura errónea de un punto del espacio k el cual contiene una gran señal de pico. Deshabilitada por defecto.
- **Contaminación RF:** Genera la aparición de una o varias líneas discontinuas y es debido a la aparición de ruido en una única frecuencia. Normalmente, su origen es debido a que la puerta de la sala ha sido mal cerrada. Deshabilitada por defecto.
- **Desplazamiento químico:** Es debido a que las partículas de agua y de grasa precesionan a velocidades distintas, esto origina que la codificación espacial de esas partículas sea errónea y genera que la grasa aparezca desplazada en la imagen final. Deshabilitada por defecto.
- **Filtro limitador de banda:** La mayoría de las máquinas cuenta con un filtro limitador de banda en el receptor, esto evita el efecto del foldover en la dirección de la codificación de frecuencia. Si se desactiva este filtro el artefacto aparecerá.

Por otro lado está el panel de propiedades que habilita la modificación de algunas de las propiedades de la máquina. Hasta el momento sólo se permite variar la el coeficiente que determina la inhomogeneidad de campo usado entre otras cosas para calcular los tiempos T2\*.

### Notas y observaciones

**Spike y Contaminación RF:** Estos defectos no son muy comunes y suelen darse en imágenes aisladas, no en todas ellas. Por tanto es importante revisar todas las imágenes adquiridas. En nuestra consola este efecto aparece en todas las imágenes y su finalidad es educativa.

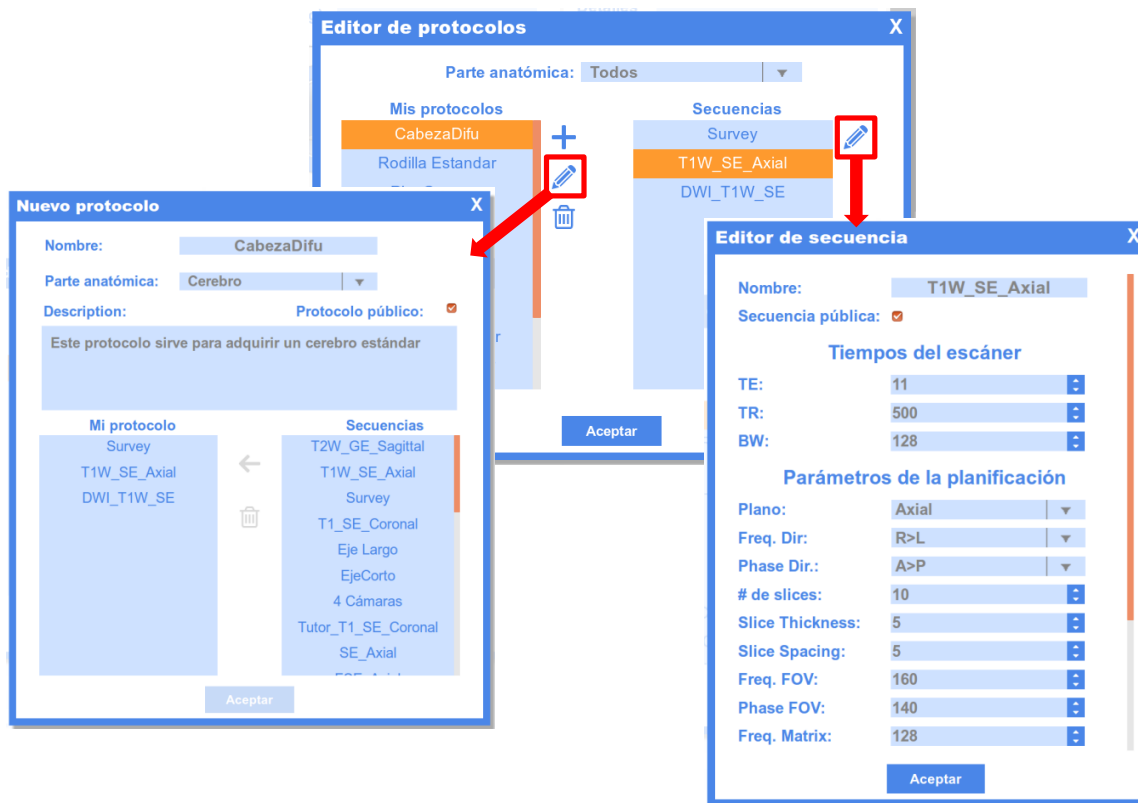
### 3.3.2 Selector de patologías.

Normalmente, nuestra consola elige automáticamente el paciente o la patología que va a mostrarse basándose en los datos de la interfaz, pero, dado que es un simulador, existe la posibilidad de tomar adquisiciones de diversos volúmenes de una misma parte anatómica. El panel selector de patologías se encarga de ello. Para abrir este panel se debe pulsar el botón con forma de paciente tumbado con un círculo.



La opción marcada es la del volumen por defecto, esto implica que los que escojan hacer por ejemplo una adquisición de cerebro con esta opción marcada verán el mismo paciente. Sin embargo, si seleccionamos la opción aleatoria, cada uno podría estar viendo un paciente distinto entre los disponibles para dicha zona anatómica. Existe una tercera opción en la que el paciente no es ni por defecto ni se elige de forma aleatoria y donde ambas opciones deben estar deseleccionadas. En ese momento se habilitan 3 desplegable donde se puede elegir la parte anatómica, la patología y el caso específico.

### 3.3.3 Editor y generador de protocolos y secuencias.



El menú para la edición de protocolos se despliega pulsando en el botón llamado “Mis protocolos” situado en el panel de administrador. La finalidad de este panel es la creación de nuevos protocolos o la edición de los ya existentes. El panel principal muestra un desplegable donde elegir la parte anatómica, y dos recuadros, uno dedicado a los protocolos y otro a las secuencias que lo componen.

Mis  
Protocolos

Si es la primera vez que crea un protocolo, el recuadro “Mis protocolos” aparecerá vacío y sólo podrá pulsar el botón con forma de más que se encuentra a su derecha. Este botón le permitirá añadir un nuevo protocolo. Al pulsarlo, se abrirá el panel de “Nuevo protocolo” y podrá comenzar con su creación. Podrá cambiar el nombre, seleccionar las secuencias, añadir una descripción, y si es profesor, podrá seleccionar ese protocolo como público, haciéndolo visible para el resto de usuarios. Los botones con forma de lápiz y de papelera valen para editar un nuevo protocolo y para eliminar uno ya existente.

Las secuencias también pueden ser editadas, una vez que selecciona un protocolo, podrá seleccionar una de las secuencias que lo componen y editarla. La secuencia contiene una serie de propiedades que podrán ser modificadas.

### 3.3.4 Otros paneles y características.

En nuestra consola de resonancia magnética existen otros paneles con diferentes funciones que mostraremos a lo largo de este punto.

## CARGA Y DESCARGA DEL ESTADO DE LA CONSOLA

Los botones situados en el panel de administrador con forma de hojas y una flecha dibujada en su interior habilitan la carga y la descarga del estado de la consola. Su finalidad es que un profesor pueda ver el estado del protocolo actual de un alumno y comprobar sus parámetros. También es interesante para el alumno guardar el estado de la consola tras conseguir una buena adquisición.



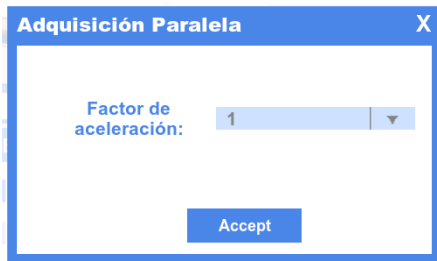
## PANEL DWI



Este panel permite seleccionar los parámetros necesarios para hacer una adquisición ponderada en difusión. Está disponible únicamente en las secuencias que lo permiten y se abre pulsando el botón situado en parámetros adicionales y que se llama DWI. Este panel permite la selección de los distintos valores b.



## PANEL DE ADQUISICIÓN PARALELA



Este panel permite habilitar la opción de adquirir la imagen paralelamente. El botón de apertura se encuentra en el panel de parámetros adicionales y se llama SENSE. Permite seleccionar el factor de aceleración con el que se va a realizar la adquisición.



## BANDAS DE SATURACIÓN

Las bandas de saturación se activan desde el panel de opciones de imagen, marcando la opción “Bandas de saturación”, esto habilitará las bandas de saturación en el panel de GraphicRX. Las bandas se situarán en la dirección de fase. La finalidad de estas bandas es anular la señal de todos los tejidos para que no produzcan artefactos en la imagen. En nuestra consola de simulación estas bandas tienen un color azul.



**Notas y observaciones**

**Bandas de saturación:** En algunos sistemas comerciales estas bandas suelen dibujarse en color rojo y permiten una colocación manual de las mismas. Suelen estar definidas por dos variables, la localización (Location) y el grosor (Thickness) y cuentan con un panel en el interior del planificador o localizador. En él pueden elegirse la dirección en la que aparecen Superior, Inferior, Anterior, Posterior, Derecha e Izquierda.

## **4.0 Usando el ITS**

## 4.0 USANDO EL ITS

### 4.1 ¿Qué es sistema de tutoría inteligente?

El concepto de sistema de tutoría inteligente (ITS en sus siglas en inglés) existe desde los años 80 y consiste en proporcionar un servicio de tutoría que apoye el aprendizaje. Se basa en la trinidad compuesta por:

- El área del conocimiento y los conceptos, reglas y estrategias que componen este dominio.
- El estudiante como componente central. Contendría información sobre los estados cognitivos del estudiante y su evolución a medida que avanza.
- El sistema de tutoría, que utiliza tanto la información proveniente el área de conocimiento como la proporcionada por estudiante durante la realización del ejercicio. Todos estos datos son usados para tomar decisiones sobre estrategias y acciones de la tutoría.

### 4.2 Cómo acceder y cómo utilizarlo.

Una vez dentro del simulador, el aspecto será similar al siguiente. La figura muestra la interfaz de usuario, compuesta por el panel principal y una barra lateral.



La barra lateral cuenta con una serie de opciones explicadas anteriormente. Seleccionamos la llamada “Exp. Tutor” que nos conducirá a un nuevo menú como el que se muestra a continuación.





Este, a su vez, se compone por varias secciones, en el área derecho aparecen 3 ejercicios, seleccionaremos cualquiera de ellos y pulsaremos al botón “Comenzar”. Al momento de hacer click, se abrirá una nueva consola similar a la de la siguiente imagen.

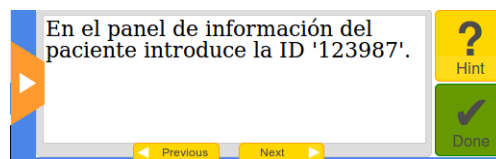


Esta consola integra el sistema de tutoría inteligente y cuenta con algunos cambios con respecto a la consola principal. El primero sería el enunciado, que es una breve descripción del ejercicio que espera que se realice. En cualquier momento que se necesite, es posible volver a abrirlo pulsando el botón “Enunciado” de la consola. El segundo elemento es el cuadro de diálogo con el tutor y permite la comunicación con el tutor. Cuenta con un cuadro de texto, donde el tutor puede realizar diferentes observaciones sobre el elemento activo. También cuenta con dos botones, el de pistas (“Hint”) y el de

terminado (“Done”). El botón amarillo “Hint” será la forma de indicar al tutor que necesitas ayuda para continuar, o también puede ser utilizado si se desea más información sobre alguno de los elementos que van a tratarse en el ejercicio. El botón verde “Done” se pulsará únicamente cuando el estudiante crea que ha finalizado el ejercicio.

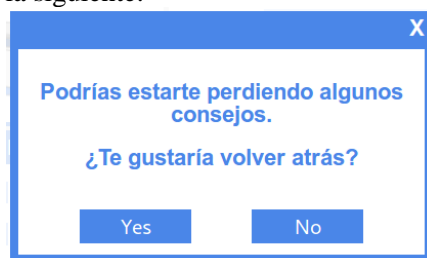
Su forma de indicar cuando alguno de los campos es correcto o no, es iluminando con verde o con rojo un determinado campo, en las imágenes de a continuación se muestra un ejemplo, donde la palabra “Hola” estaría mal introducida, mientras que el número “123987” sería correcto.

Si no es correcta la solución, podría ser un buen momento de pedir una pista al tutor. Para ello pulse en el elemento en rojo y pida una pista pulsando en el botón amarillo. En el cuadro de diálogo aparecerá algo como esto:



Las pistas pueden contar con diferentes “niveles”, aquí entran en juego los botones “Previous” y “Next” que sirven para ir consiguiendo pistas más concretas sobre la solución del error.

En ocasiones, la notificación de un error o un elemento correcto puede llegar al pulsar un botón, como podría ser el caso de pulsar el botón “Aceptar” en el panel de Graphic RX o al pulsar “Scan”. En este caso se muestra una pregunta similar a la siguiente:



Si contestas “No”, la consola realizará la acción que le has mandado. En caso de contestar “Yes” te dará la oportunidad de ver qué áreas están mal. Cuando se pulse el botón “Scan” en muchos casos puede ser interesante dar a “No” para ver cuál sería el resultado de la simulación.

**5.0 FAQ**

## **5.0 FAQ**

### **5.1 ¿Qué hacer cuando no aparecen los protocolos?**

Este problema, generalmente es debido a que la sesión del usuario ha caducado. Para solucionarlo debe pulsar al botón de “Logout” situado en la barra lateral y volver a iniciar sesión pulsando en el botón “Login”.

### **5.2 ¿Por qué el protocolo que acabo de crear no aparece?**

En ocasiones, el protocolo que acabas de crear no aparece en la selección de protocolos de la consola de adquisición. Este problema generalmente se resuelve actualizando la página; en la mayoría de los navegadores, el botón para esta acción es el F5.

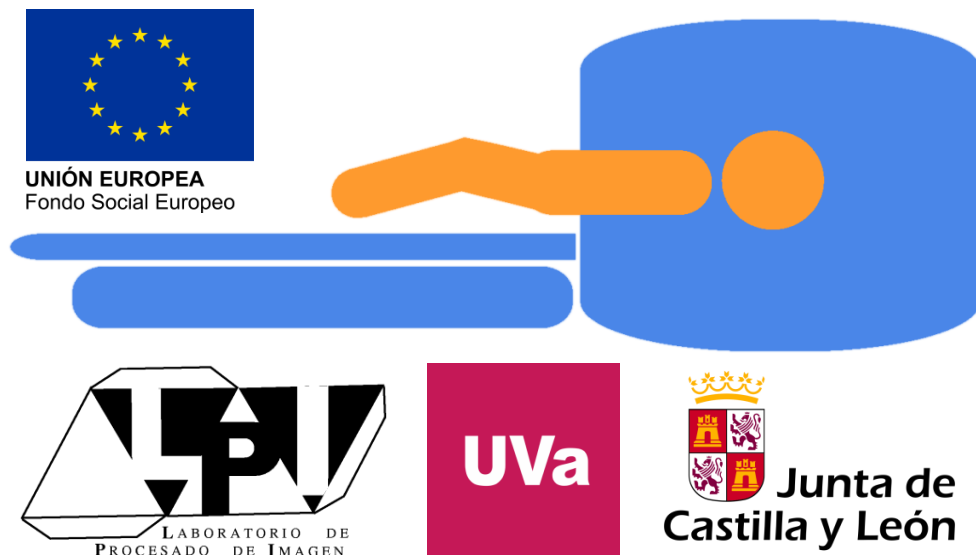
*Apéndice F*

## CREACIÓN DE UN EJERCICIO CON CTAT

# GUÍA PARA LA CREACIÓN DE UN ERJERCICIO CON CTAT

## SIMUMRI

Mayo, 2019



## 1.0 GUÍA RÁPIDA PARA LA CREACION DE UN EJERCICIO CON CTAT

### 1.1 Creación de un enunciado

El enunciado es lo primero que verá el alumno al entrar en el ejercicio del tutor inteligente. Hasta el momento no es posible añadir un ejercicio usando la base de datos y es necesario escribirlo directamente en el código de la aplicación. El archivo donde se especifica el enunciado del problema y el curso al que pertenece se encuentra en el directorio `/devClient/js/angular/other/services/other.courses.js`. Está compuesto por un objeto principal llamado `courses` que consiste en un vector de objetos donde cada objeto es un curso. A su vez, cada curso, cuenta con un vector llamado `lessons` que aloja los diferentes ejercicios o enunciadas que conforman dicho curso. Cada objeto en `lessons` alberga en el enunciado en un objeto llamado `msgText`, este está preparado para contener varios párrafos y puntos.

The image shows a composite of three elements related to the CTAT application:

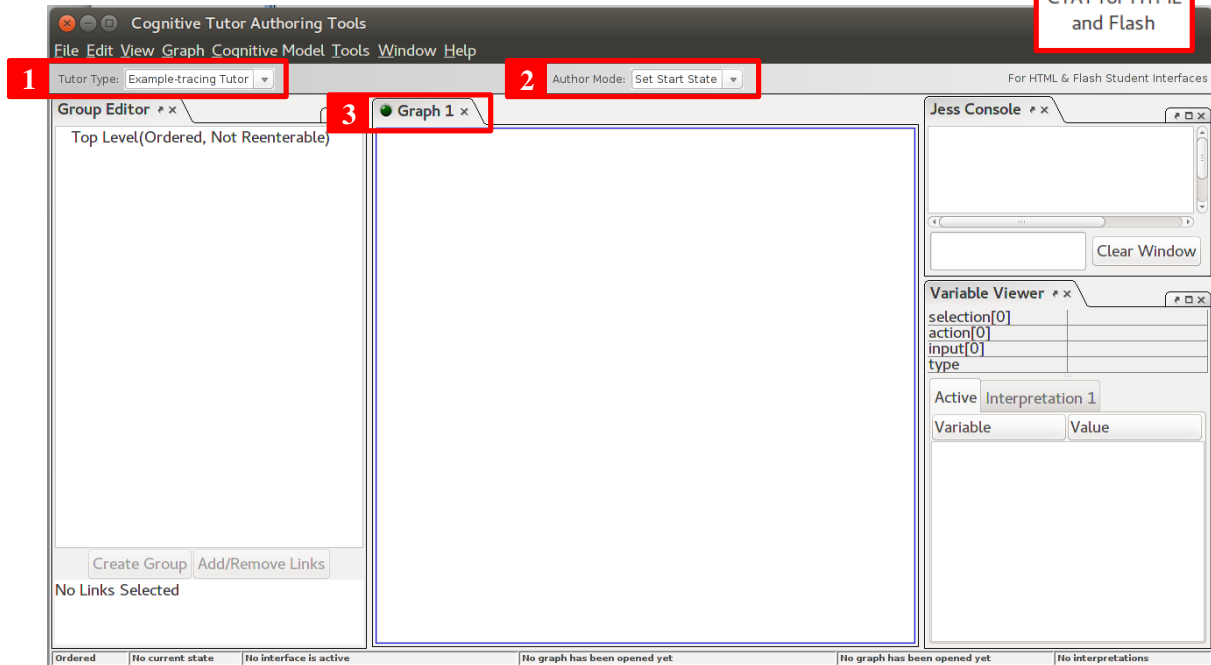
- Top Left:** A screenshot of the 'Exp. Tutor' interface. It features a table titled 'Ejercicios con tutor:' with columns '#', 'Ejercicios', and 'Descripción:'. The first row is highlighted in orange and contains '1', 'Tu primera adquisición', and 'En ese ejercicio se realizará una adquisición completa, incluyendo survey, planificación y un volumen ponderado en T1'. A red box labeled 'Comenzar' is positioned below the table.
- Top Right:** A code block showing the structure of the `courses` object. It includes fields for `selected`, `options` (with `id`, `name`, `date`, `modality`, `anatomicalPart`), and `lessons` (with `itemSelected` and `description`).
- Bottom Left:** A code block showing the structure of an individual exercise object. It includes fields for `id`, `name`, `description`, `brd`, `problemDef`, `msgText` (with `type` and `text`), `imgData`, and `acceptVisible`.
- Bottom Right:** A screenshot of the 'Enunciado' (Exercise Statement) screen. It displays the text from the first row of the table, a brain scan image with a red box, and a list of instructions: 'Utiliza las pistas 'hints' si no sabes cómo continuar...', 'Comienza con la información del paciente: ID: 123987; Nombre: Daniel; Fecha: 23/06/1999; Peso: 75. A continuación rellena la posición en la que debes introducirse al paciente para un escaner de cabeza.', 'Elige la bobina óptima de las posibilidades ofrecidas.', and 'Ejecuta el protocolo completo sin olvidar realizar la planificación para la secuencia'. There is an 'Accept' button at the bottom.

Una vez añadido el texto, el programa debe ser “compilado” siguiendo estos pasos. Primero debe irse a la carpeta `devSimuMRI`, una vez en ella, ejecutar el comando “`gulp rmConsole`”. Este comando sustituirá los archivos existentes en la carpeta `server` por los nuevos compilados.

### 1.2 Cómo crear un BRD.

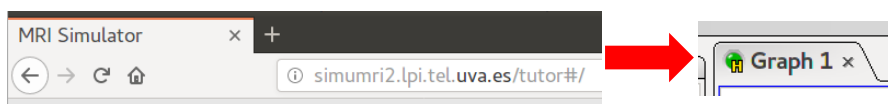
Los archivos `.brd` son propios de CTAT (Cognitive Tutor Authoring Tools), esta abreviación se corresponde con “Behavior Recorder graph” en sus siglas en inglés y su propósito durante su creación es almacenar los pasos que se van realizando. Una vez creado y editado, es utilizado por el seguidor o “Tracer” escrito en javascript para verificar el camino que toma el usuario.

El primer paso es descargar CTAT, en el este [link](#) puede encontrarse la versión 4.30 de CTAT. Una vez instalada es necesario lanzar “CTAT for HTML and Flash”.



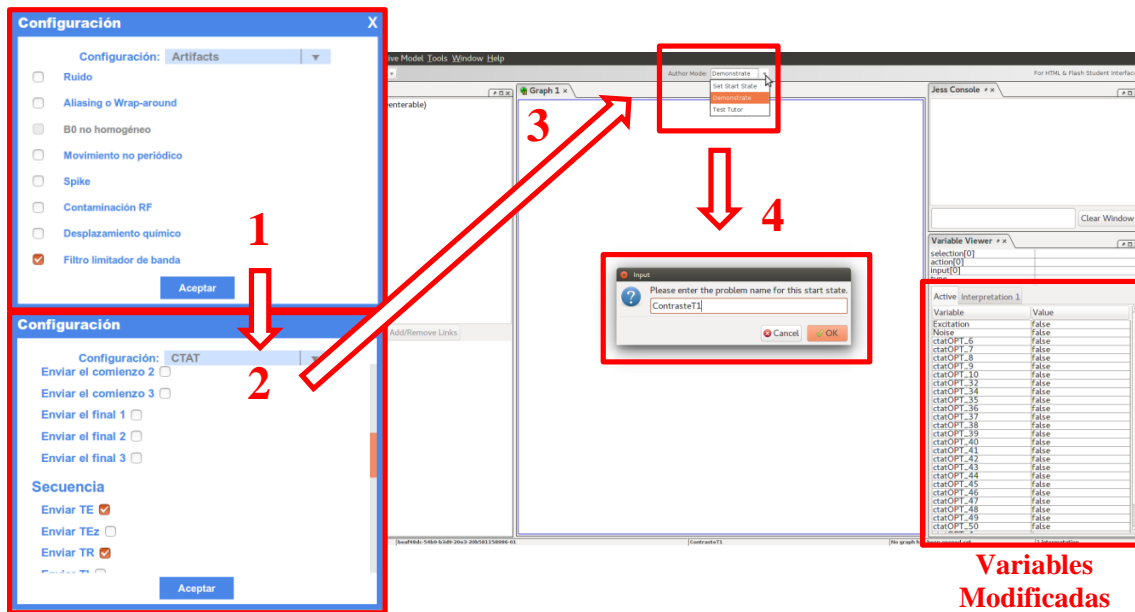
En la etiqueta 1 de la imagen podemos seleccionar el tipo de tutor (Tutor Type), en nuestro caso siempre trabajaremos con el “Example-tracing Tutor” que es el necesario para la creación de nuestro brd. En el modo autor (Author Mode), el número 2, existen 3 opciones diferentes: configurar el estado inicial (Set Start State), demostrar (Demostrate) y probar el tutor (Test Tutor). Finalmente el número 3 es la pestaña del brd que vamos a crear.

El siguiente paso es abrir SimuMRI, pero esta vez con una peculiaridad, debemos añadir la palabra “tutor” al final de la url.



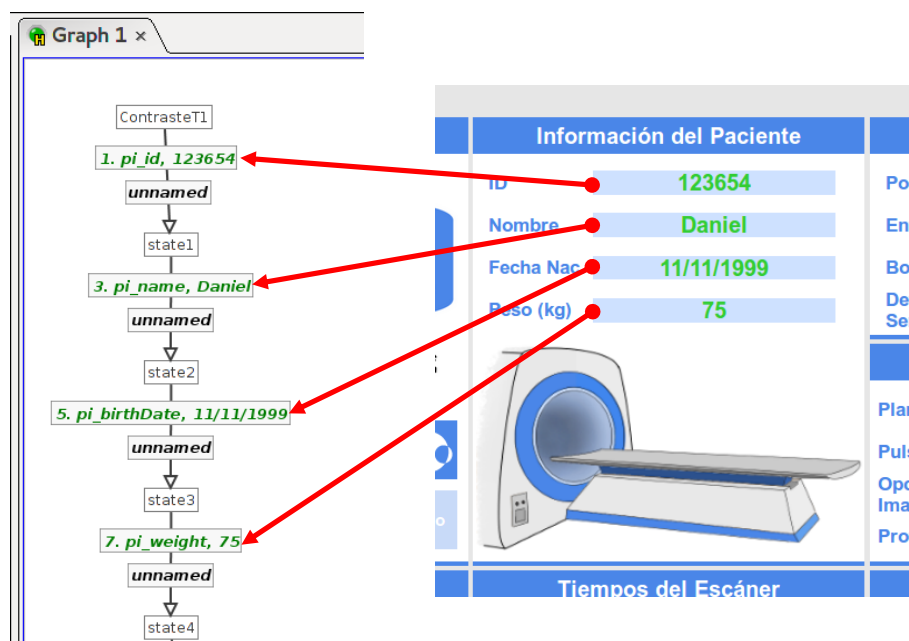
Esto nos dará acceso al SimuMRI que tiene activado CTAT. Si ahora nos fijamos en la pestaña del gráfico de CTAT, se mostrará de un tono verde más brillante y aparecerá una pequeña H amarilla indicando que ahora, tanto CTAT como la interfaz de SimuMRI está conectados. Podemos pasar a establecer el estado inicial de la consola. SimuMRI cuenta con una ventana de configuración que resulta de gran utilidad para la creación de ejercicios en CTAT. Supongamos un ejercicio en el que se desea mostrar adquisiciones ponderadas en T1, en este caso el ruido o el foldover no serían características importantes en la simulación y podrían interesar desactivarlas por defecto para este ejercicio. Además, cuenta con un menú llamado CTAT dentro del panel de configuración. En este menú podemos elegir qué elementos de SimuMRI queremos que se envíen para la generación del brd al pulsar el botón *Scan* y el botón *Aceptar* del panel *GraphicRX*. Existen unas opciones marcadas por defecto, pero siguiendo el ejemplo, podríamos modificarlas para enviar solamente el TE y TR al pulsar Scan y el FOV en fase y en frecuencia al pulsar Aceptar. Esto provoca que tengamos que dar muchas menos pistas o realimentaciones a los estudiantes y afrontar problemas más concretos.



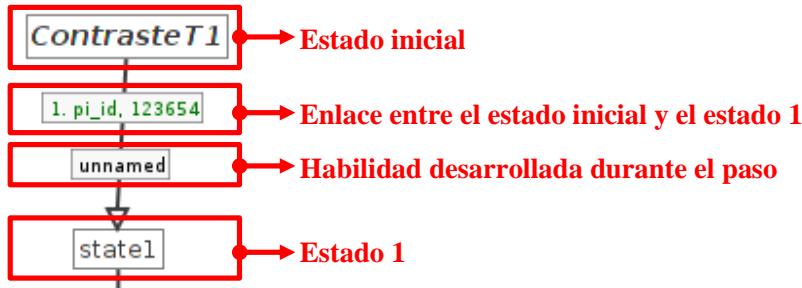


Brevemente, iríamos a la interfaz de SimuMRI, pulsaríamos en el botón de configuración, estableceríamos la configuración deseada tanto en *Artifacts* como en *CTAT*. Volveríamos al panel de *CTAT* y ahí seleccionaríamos el modo de tutor *Demonstrate* y estableceríamos el nombre del estado inicial en el árbol. Si nos fijamos en el visor de variables, aparecerían las primeras opciones como que la *Excitation* y *Noise* puestas al valor *false* y además varias opciones que empiezan por *ctatOPT* que establecerían los elementos que no van a ser registrados en el tutor.

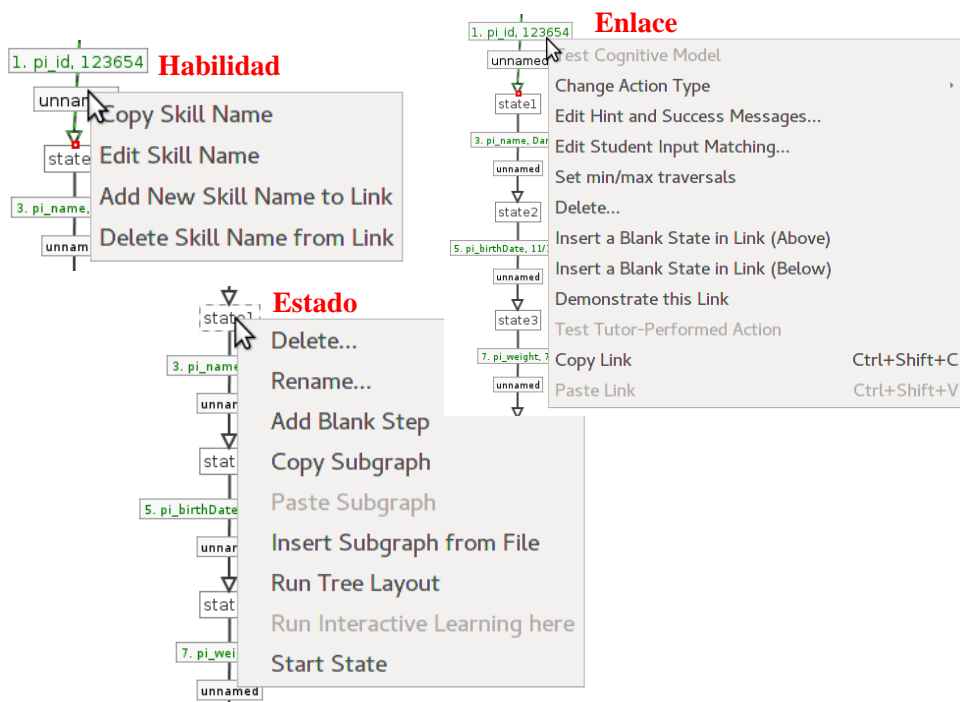
El tutor ahora estaría preparado para realizar la demostración. Esta tarea es sencilla, sólo deberemos establecer los pasos que el estudiante debería dar para realizar la tarea completa. Comenzamos introduciendo los datos del paciente:



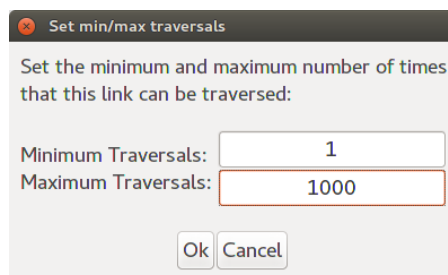
Esto irá dibujando el árbol de decisiones que hemos ido tomando. Al fijarse en él, cada una de las decisiones están formadas por 3 componentes, el estado, el enlace (entre estados) y la habilidad o habilidades que se desarrolla.



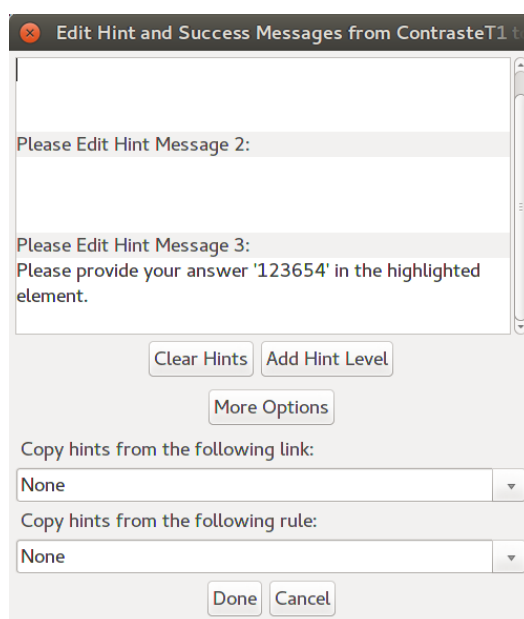
Y cada uno cuenta con menús distintos al pulsar clic derecho sobre ellos. A continuación explicamos algunas cosas relevantes. El estado permite que sea eliminado, dando la posibilidad de eliminar el resto de gráfico que cae de él. Además permite cosas como renombrarlo con una palabra más descriptiva o insertar subgráficos de otros brd. El de habilidad permite básicamente manejar los nombres de las habilidades en las que se progresa. El menú más importante es el que se desprende del enlace y las opciones más utilizadas son las de “Edit Hint and Success Messages...”, “Edit Student Input Matching...” y “Set min/max traversals” que veremos más detalladamente.



“Set min/max traversals” en este menú estableces el mínimo número y máximo de veces que puede atravesarse este enlace entre estados, en el simulador, es muy habitual que tengan que enviarse varias veces la misma información, así que lo aconsejado es poner estos valores para cada enlace entre 1 y 1000. El 0 significaría que no es necesario atravesar el enlace, en otras palabras, que el paso no sería necesario para completar el ejercicio.



“Edit Hint and Success Messages...” sirve para editar las pistas y los mensajes que envías al estudiante cuando avanza por cada una de los estados. Como puede verse ne la imagen tiene hasta 3 posibles pistas por defecto, además, permite copiar las pistas y los mensajes de otros pasos y si se pulsa en el botón de “More Options” permite dar un mensaje cuando el paso se ha realizado con éxito.



“Edit Student Input Matching...” es el menú más complejo dado que cuenta con una gran variedad de posibilidades para poder establecer la operación que debe realizar para pasar a través de este enlace satisfactoriamente. Cada enlace tiene cuatro atributos, la selección (Selection) que el ID de elemento que responsable, la acción (Action) que realiza dicho elemento, la entrada (Input) o valor que retorna esa acción y finalmente el actor (Actor) que la realiza. La selección como la acción son parámetros que no son necesarios tocar en general por lo que no se explicarán en mayor detalle. El Actor puede ser o bien el Tutor, es decir, la acción se desarrollaría automáticamente llegado el momento, o bien el estudiante, la acción sólo sucede si lo hace es estudiante, o cualquiera de los dos. En SimuMRI es recomendable dejar como actor al estudiante. Sin embargo, el valor que retorna el elemento tiene un gran número de posibilidades para su validación, por defecto está el “Exact Match” es decir, que el valor que retorna sea igual al valor especificado.

Sin embargo, permite el uso de introducir funciones para compararlas con el valor esperado, utilizar expresiones regulares o establecer un rango de valores como se ven en las siguientes imágenes.

**Edit Student Input Matching**

Link #1	Demonstrated Value	Matcher Settings	Matcher Type and Check
Selection: pi_id	pi_id	pi_id	Exact Match
Action: textModified	textModified	textModified	Exact Match
Input: 123654	123654	123654	Exact Match

Actor:  Tutor  Tutor (unevaluated)  Student  Any  
 Trigger:  On current state  On previous link

When correct, replace student input with: (don't replace)

OK Cancel

Input: 123987

action

ctatOPT\_32

ctatOPT\_34

ctatOPT\_35

ctatOPT\_36

ctatOPT\_37

ctatOPT\_38

ctatOPT\_39

boolean

regExMatch( "[0-9]{6}", input )

Formula Match

Check

Input: Daniel

action

ctatOPT\_32

ctatOPT\_34

ctatOPT\_35

ctatOPT\_36

^[a-zA-Z\D]+\$

Regular Expression Match

Check

Input: 75

action

ctatOPT\_32

ctatOPT\_34

ctatOPT\_35

ctatOPT\_36

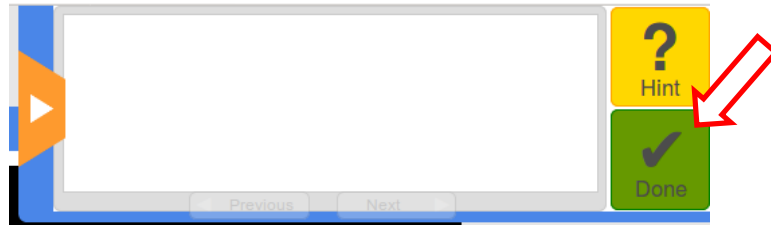
Min: 30

Max: 150

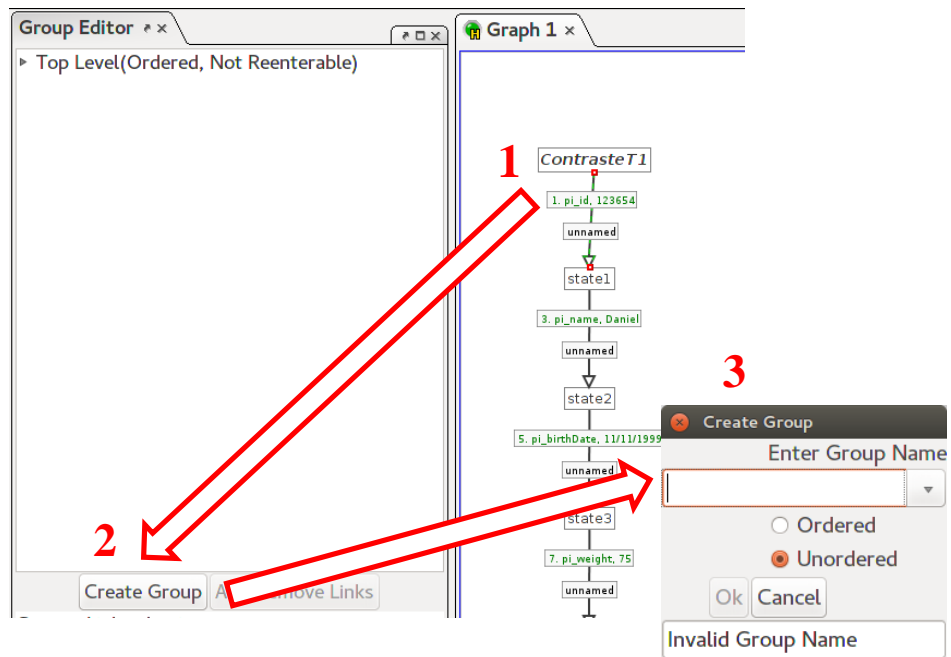
Range Match

Check

Después de realizar todos los pasos que deseemos es importante pulsar el botón “Done” situado en el cuadro de diálogo del CTAT dentro de la interfaz de SimuMRI.



Con esto el árbol se dará por finalizado y pasaremos al modo “Test tutor”. En este modo es interesante agrupar los diferentes pasos o enlaces en grupos. Estos grupos pueden permitir o no la reentrada de argumentos y pueden ser ejecutados en orden o no. Para la creación de un grupo lo primero que hay que hacer es selección un enlace haciendo clic izquierdo encima de él. Esto habilita el botón “Create Group” situado en el panel de “Group Editor”. Una vez creado el grupo sólo debemos arrastrar los enlaces al interior del mismo.



Los grupos se ordenan por nombre, por tanto es aconsejable comenzar cada grupo por un número para que presenten un aspecto más ordenado a la hora de seguir el árbol de decisiones.



# BIBLIOGRAFÍA

---

- Albert, W., Tullis, T., 2013. Measuring the user experience: collecting, analyzing, and presenting usability metrics. Newnes.
- Aleven, V., McLaren, B. M., Sewall, J., 2009a. Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning. *IEEE Transactions on Learning Technologies* 2 (2), 64–78.
- Aleven, V., McLaren, B. M., Sewall, J., Koedinger, K. R., 2009b. A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education* 19 (2), 105–154.
- Aleven, V., McLaren, B. M., Sewall, J., Van Velsen, M., Popescu, O., Demi, S., Ringenberg, M., Koedinger, K. R., 2016. Example-tracing tutors: Intelligent tutor development for non-programmers. *International Journal of Artificial Intelligence in Education* 26 (1), 224–269.
- Almohammadi, K., Hagra, H., Alghazzawi, D., Aldabbagh, G., 2016. Users-centric adaptive learning system based on interval type-2 fuzzy logic for massively crowded e-learning platforms. *Journal of Artificial Intelligence and Soft Computing Research* 6 (2), 81–101.
- Anderson, C., 2012. *The Model-View-ViewModel (MVVM) Design Pattern*. Apress, Berkeley, CA, pp. 461–499.
- Andrews, D. H., 1988. Relationships among simulators, training devices, and learning: A behavioral view. *Educational Technology* 28 (1), 48–54.
- Arroyo, I., Woolf, B. P., Burelson, W., Muldner, K., Rai, D., Tai, M., Dec 2014. A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *International Journal of Artificial Intelligence in Education* 24 (4), 387–426.
- Arsanjani, A., Booch, G., Boubez, T., Brown, P., Chappell, D., deVadoss, J., Erl, T., Josuttis, N., Krafzig, D., Little, M., et al., 2009. The soa manifesto. *SOA Manifesto* 35, 82–88.
- Ash, J. S., Berg, M., Coiera, E., 2004. Some unintended consequences of information technology in health care: the nature of patient care information system-related errors. *Journal of the American Medical Informatics Association* 11 (2), 104–112.
- Bangor, A., Kortum, P., Miller, J., 5 2009. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Studies* 4 (3), 114–123.

- Benoit-Cattin, H., Collewet, G., Belaroussi, B., Saint-Jalmes, H., Odet, C., 2005. The simri project: a versatile and interactive mri simulator. *Journal of Magnetic Resonance* 173 (1), 97–115.
- Bernstein, M. A., King, K. F., Zhou, X. J. (Eds.), 2004. *Handbook of MRI Pulse Sequences*. Academic Press, Burlington.
- Blessing, S. B., Gilbert, S. B., Ourada, S., Ritter, S., 2009. Authoring model-tracing cognitive tutors. *International Journal of Artificial Intelligence in Education* 19 (2), 189–210.
- Bloch, F., 10 1946. Nuclear induction. *Physical Review* 70, 460–474.
- Bloch, F., Purcell, E. M., 1952. Nobel prize for physics, 1952. *Nature* 170, 911–912.
- Boehm, B., Aug. 1986. A spiral model of software development and enhancement. *SIGSOFT Softw. Eng. Notes* 11 (4), 14–24.
- Brooke, J., 11 1995. Sus: A quick and dirty usability scale. *Usability Eval. Ind.* 189.
- Brooke, J., 2 2013. Sus: A retrospective. *J. Usability Studies* 8 (2), 29–40.
- Brown, R. W., Cheng, Y.-C. N., Haacke, E. M., Thompson, M. R., Venkatesan, R., 2014. *Magnetic resonance imaging: physical principles and sequence design*. John Wiley & Sons.
- Burch, C., 5 2010. Django: a web framework using python: Tutorial presentation. *Journal of Computing Sciences in Colleges* 25 (5), 154–155.
- Cai, Z., Graesser, A., Hu, X., 2015. Asat: Autotutor script authoring tool. *Design recommendations for intelligent tutoring systems: authoring tools* 3 (2015), 199–210.
- Cao, Z., Oh, S., Sica, C. T., McGarrity, J. M., Horan, T., Luo, W., Collins, C. M., 2014. Bloch-based mri system simulator considering realistic electromagnetic fields for calculation of signal, noise, and specific absorption rate. *Magnetic resonance in medicine* 72 (1), 237–247.
- Cao, Z., Sica, C., Oh, S., McGarrity, J., Horan, T., Park, B., Collins, C., 2010. An mri simulator for effects of realistic field distributions and pulse sequences, including sar and noise correlation for array coils. In: *Proceedings of ISMRM*. p. 1456.
- Carbonell, J. R., 01 1971. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *Man-Machine Systems, IEEE Transactions on* 11, 190 – 202.
- Chen, S., Wang, Y., Tsui, B. M., 2007. Development of simulation tools for small animal spect/mri reconstruction study. In: *2007 IEEE Nuclear Science Symposium Conference Record*. Vol. 5. IEEE, pp. 3250–3255.
- Chi, M. T. H., VanLehn, K. A., 1991. The content of physics self-explanations. *The Journal of the Learning Sciences* 1 (1), 69–105.
- Conati, C., Merten, C., 2007. Eye-tracking for user modeling in exploratory learning environments: An empirical evaluation. *Knowledge-Based Systems* 20 (6), 557–574.



- Cronbach, L. J., Sep 1951. Coefficient alpha and the internal structure of tests. *Psychometrika* 16 (3), 297–334.
- Crowley, R., Medvedeva, O., 2003. Slidetutor: A model-tracing intelligent tutoring system for teaching microscopic. *Artificial Intelligence in Education: Shaping the Future of Learning Through Intelligent Technologies* 97, 157.
- Damadian, R., 1971. Tumor detection by nuclear magnetic resonance. *Science* 171 (3976), 1151–1153.
- DIS, I., 2009. 9241-210: 2010. ergonomics of human system interaction-part 210: Human-centred design for interactive systems. International Standardization Organization (ISO). Switzerland.
- Drobnjak, I., Gavaghan, D., Süli, E., Pitt-Francis, J., Jenkinson, M., 08 2006. Development of a functional magnetic resonance imaging simulator for modeling realistic rigid-body motion artifacts. *Magnetic Resonance in Medicine* 56, 364–80.
- Drobnjak, I., S., P. G., Jenkinson, M., 2010. Simulating the effects of time-varying magnetic fields with a realistic simulated scanner. *Magnetic Resonance Imaging* 28 (7), 1014 – 1021.
- Edelman, R. R., 2014. The history of mr imaging as seen through the pages of radiology. *Radiology* 273 (2S), S181–S200.
- Edelstein, W., Glover, G., Hardy, C., Redington, R., 1986. The intrinsic signal-to-noise ratio in nmr imaging. *Magnetic resonance in medicine* 3 (4), 604–618.
- Elmaoglu, M., Çelik, A., 2012. *MRI Handbook*, 1st Edition. Springer-Verlag New York, Yeni Yüzyıl University, Cevizlibağ Topkapi, Turkey.
- Elsom-Cook, M., 1993. Student modelling in intelligent tutoring systems. *Artificial Intelligence Review* 7 (3-4), 227–240.
- Erasmus, L., Hurter, D., Naude, M., Kritzing, H., Acho, S., 2004. A short overview of mri artefacts. *South African Journal of Radiology* 8 (2).
- Eurostat, S., 2016. Number of magnetic resonance imaging (mri) scan examinations in spain from 2010 to 2016 (in 1,000).
- Fortin, A., Durand, E., Salmon, S., 2014. Extension of an mri simulator software for phase contrast angiography experiments. In: *International Symposium on Biomedical Simulation*. Springer, pp. 150–154.
- Fraenkel, J. R., Wallen, N. E., Hyun, H. H., 01 2011. *How to Design and Evaluate Research in Education*, 8th Edition. Vol. 60. McGraw-Hill Education.
- Frey, B. B., 2018. *The SAGE encyclopedia of educational research, measurement, and evaluation*. Vol. 1-4. Thousand Oaks, CA, USA: SAGE Publications, Inc.

- Gertner, A., Conati, C., Vanlehn, K., 1998. Procedural help in andes: Generating hints using a bayesian network student mode. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence. AAAI-98, pp. 106–111.
- Gilbert, S. B., Blessing, S., Kodavali, S. K., 2009. The extensible problem-specific tutor (xpst): Evaluation of an api for tutoring on existing interfaces. In: AIED. pp. 707–709.
- Gilbert, S. B., Blessing, S. B., Guo, E., 2015. Authoring effective embedded tutors: An overview of the extensible problem specific tutor (xpst) system. *International Journal of Artificial Intelligence in Education* 25 (3), 428–454.
- Google, 2019. Angularjs. <https://angularjs.org/>, accessed: 24-07-2019.
- Graesser, A. C., 2016. Conversations with autotutor help students learn. *International Journal of Artificial Intelligence in Education* 26 (1), 124–132.
- Graesser, A. C., D’Mello, S., 2012. Moment-to-moment emotions during reading. *The Reading Teacher* 66 (3), 238–242.
- Graesser, A. C., Hu, X., Nye, B. D., VanLehn, K., Kumar, R., Heffernan, C., Heffernan, N., Woolf, B., Olney, A. M., Rus, V., et al., 2018. Electronixtutor: an intelligent tutoring system with multiple learning resources for electronics. *International journal of STEM education* 5 (1), 15.
- Graesser, A. C., Li, H., Forsyth, C., 2014. Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science* 23 (5), 374–380.
- Graham, M. S., Drobnjak, I., Jenkinson, M., Zhang, H., 2017. Quantitative assessment of the susceptibility artefact and its interaction with motion in diffusion mri. *PloS one* 12 (10), e0185647.
- Graham, M. S., Drobnjak, I., Zhang, H., 2016. Realistic simulation of artefacts in diffusion mri for validating post-processing correction techniques. *NeuroImage* 125, 1079–1094.
- Griswold, M. A., Jakob, P. M., Heidemann, R. M., Nittka, M., Jellus, V., Wang, J., Kiefer, B., Haase, A., 2002. Generalized autocalibrating partially parallel acquisitions (grappa). *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 47 (6), 1202–1210.
- Hackländer, T., Mertens, H., 02 2005. Virtual MRI: A PC-based simulation of a clinical MR scanner. *Academic Radiology* 12, 85–96.
- Hahn, E. L., 1950. Spin echoes. *Physical review* 80 (4), 580.
- Hallowell, L. M., Stewart, S. E., e Silva, C. T. d. A., Ditchfield, M. R., 2008. Reviewing the process of preparing children for mri. *Pediatric radiology* 38 (3), 271.
- Hanson, L. G., 2007. A graphical simulator for teaching basic and advanced mr imaging techniques. *Radiographics* 27 (6), e27–e27.

- Hanson, L. G., 2008. Is quantum mechanics necessary for understanding magnetic resonance? *Concepts in Magnetic Resonance Part A: An Educational Journal* 32 (5), 329–340.
- Hashemi, R. H., Bradley, W. G., Lisanti, C. J., 2012. *MRI: The Basics: The Basics*. Lippincott Williams & Wilkins.
- Heffernan, N. T., Heffernan, C. L., 2014. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24 (4), 470–497.
- Hill Jr, R. W., Belanich, J., Lane, H. C., Core, M., Dixon, M., Forbell, E., Kim, J., Hart, J., 2006. Pedagogically structured game-based training: Development of the elect bilat simulation. In: *Proceedings of the 25th Army Science Conference*. Proceedings of the 25th Army Science Conference.
- Jeon, S. S., Lee, G., Lampotang, S., Su, S. Y., 2007. An integrated e-learning system for simulation-based instruction of anaesthesia machines. *International Journal of Knowledge and Learning* 3 (1), 106–120.
- Jeon, S. S., Su, S. Y., 2010. Adaptive e-learning using ecpaa rules, bayesian models, and group profile and performance data. *International Journal of Learning Technology* 5 (4), 415–434.
- Jobs, S., 4 2010. Thoughts on flash. <https://www.apple.com/hotnews/thoughts-on-flash/>, accessed: 19-09-2019.
- Jochimsen, T. H., Von Mengershausen, M., 2004. Odin—object-oriented development interface for nmr. *Journal of Magnetic Resonance* 170 (1), 67–78.
- Jorritsma, W., Cnossen, F., Dierckx, R. A., Oudkerk, M., Ooijen, P. M. V., 2016. Post-deployment usability evaluation of a radiology workstation. *International Journal of Medical Informatics* 85 (1), 28 – 35.
- Kabanza, F., Nkambou, R., Belghith, K., 2005. Path-planning for autonomous training on robot manipulators in space. In: *IJCAI*. pp. 1729–1731.
- Kantasis, G., Xanthis, C. G., Haris, K., Aletras, A. H., 2015. Massively parallel cuda simulations of cardiac and embryonic mri on a cloud-based cluster. In: *2015 Computing in Cardiology Conference (CinC)*. IEEE, pp. 469–472.
- Kazi, H., Haddawy, P., Suebnukarn, S., 2012. Employing umls for generating hints in a tutoring system for medical problem-based learning. *Journal of biomedical informatics* 45 (3), 557–565.
- Kenta, H., Kenichi, K., Syuhei, H., Toru, N., Masayuki, S., Toshihiro, O., Hiroi, N., Takayuki, O., Tetsuo, S., Tomohisa, O., Hitoshi, I., 01 2009. Evaluation of simulation accuracy in MR imaging simulation for education of radiological technology. *Transactions of Japanese Society for Medical and Biological Engineering* 47, 237–243.
- Kimberlin, C. L., Winterstein, A. G., 2008. Validity and reliability of measurement instruments used in research. *American journal of health-system pharmacy* 65 (23), 2276–2284.

- Kodavali, S. K., Gilbert, S., Blessing, S. B., 2010. Expansion of the xpst framework to enable non-programmers to create intelligent tutoring systems in 3d game environments. In: International Conference on Intelligent Tutoring Systems. Springer, pp. 365–367.
- Koedinger, K. R., Alevan, V., 2007. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review* 19 (3), 239–264.
- Kose, R., Kose, K., 2017. Blochsolver: a gpu-optimized fast 3d mri simulator for experimentally compatible pulse sequences. *Journal of Magnetic Resonance* 281, 51–65.
- Kuder, G. F., Richardson, M. W., 1937. The theory of the estimation of test reliability. *Psychometrika* 2 (3), 151–160.
- Kulik, J. A., Fletcher, J., 2016. Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of Educational Research* 86 (1), 42–78.
- Lane, H. C., Core, M. G., Hays, M. J., Auerbach, D., Rosenberg, M., 2015. Situated pedagogical authoring: Authoring intelligent tutors from a student’s perspective. In: International Conference on Artificial Intelligence in Education. Springer, pp. 195–204.
- Larman, C., Basili, V. R., 2003. Iterative and incremental developments. a brief history. *Computer* 36 (6), 47–56.
- Lauterbur, P. C., et al., 1973. Image formation by induced local interactions: examples employing nuclear magnetic resonance.
- Law, E., Roto, V., Vermeeren, A. P., Kort, J., Hassenzahl, M., 2008. Towards a shared definition of user experience. In: CHI’08 extended abstracts on Human factors in computing systems. ACM, pp. 2395–2398.
- Lefebvre, P. M., Van Reeth, E., Ratiney, H., Beuf, O., Brusseau, E., Lambert, S. A., Glaser, S. J., Sugny, D., Grenier, D., Koon, K. T. V., 2017. Active control of the spatial mri phase distribution with optimal control theory. *Journal of Magnetic Resonance* 281, 82–93.
- Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Roberts, L. G., Wolff, S., 2009. A brief history of the internet. *ACM SIGCOMM Computer Communication Review* 39 (5), 22–31.
- Lewis, J. R., Sauro, J., 2009. The factor structure of the system usability scale. In: Kurosu, M. (Ed.), *Human Centered Design*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 94–103.
- Listerud, J., Einstein, S., Outwater, E., Kressel, H., 1992. First principles of fast spin echo. *Magnetic resonance quarterly* 8 (4), 199–244.
- Liu, F., Kijowski, R., Block, W. F., 2013. MriLab: Performing fast 3d parallel mri numerical simulation on a simple pc. In: 2013 ISMRM. IEEE, p. 2373.

- Liu, F., Velikina, J. V., Block, W. F., Kijowski, R., Samsonov, A. A., 2017. Fast Realistic MRI Simulations Based on Generalized Multi-Pool Exchange Tissue Model. *IEEE Transactions on Medical Imaging* 36 (2), 527–537.
- Ma, W., Adesope, O. O., Nesbit, J. C., Liu, Q., 2014. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of educational psychology* 106 (4), 901.
- Mall, R., 2013. *Fundamentals of Software Engineering*. Phi Learning Private Limited, India.
- Matsuda, N., Cohen, W. W., Koedinger, K. R., 2015. Teaching the teacher: tutoring simstudent leads to more effective cognitive tutor authoring. *International Journal of Artificial Intelligence in Education* 25 (1), 1–34.
- McKagan, S., Perkins, K., Dubson, M., Malley, C., Reid, S., LeMaster, R., Wieman, C., 2008. Developing and researching phet simulations for teaching quantum mechanics. *American Journal of Physics* 76 (4), 406–417.
- Melis, E., Siekmann, J., 2004. Activemath: An intelligent tutoring system for mathematics. In: *International Conference on Artificial Intelligence and Soft Computing*. Springer, pp. 91–101.
- Mitrovic, A., Martin, B., Mayo, M., 2002. Using evaluation to shape its design: Results and experiences with sql-tutor. *User Modeling and User-Adapted Interaction* 12 (2-3), 243–279.
- Mitrovic, A., Martin, B., Suraweera, P., 2007. Intelligent tutors for all: The constraint-based approach. *IEEE Intelligent Systems* 4, 38–45.
- Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., McGuigan, N., 2009. Aspire: an authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education* 19 (2), 155–188.
- Moratal Pérez, D., Martí-Bonmatí, L., Vallés Lluch, A., 2006. *Resonancia magnética: principios físicos y aplicaciones*. Departamento de Ingeniería Electrónica, Universidad Politécnica de Valencia, 79–101.
- Mostow, J., et al., 2001. Evaluating tutors that listen: An overview of project listen.
- Mousavinasab, E., Zarifsanaiy, N., R. Niakan Kalhori, S., Rakhshan, M., Keikha, L., Ghazi Saeedi, M., 2018. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 1–22.
- Moya Sáez, E., Treceño Fernández, D., de Luis García, R., Simmross Wattenberg, F., Alberola López, C., 2017. Implementación paralela de secuencias de resonancia magnética mediante programación en gpu. In: *XXXV CASEIB*. CASEIB, pp. 29–32.
- MS, H., TS, S., 2012. Gadgetron: An open source framework for medical image reconstruction. *Magnetic Resonance in Medicine* 69 (6), 1768–1776.

- Mulgund, S., Asdigha, M., Zacharias, G., Krishnakumar, K., Dohme, J., 10 1998. Intelligent tutoring system for simulator-based helicopter flight training. In: Flight Simulation Technologies Conference.
- Murray, T., 2003. An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. In: Authoring tools for advanced technology learning environments. Springer, pp. 491–544.
- Nielsen, J., 1994. Usability engineering. Elsevier.
- Nkambou, R., Mizoguchi, R., Bourdeau, J., 2010. Advances in intelligent tutoring systems. Vol. 308. Springer Science & Business Media.
- Nwana, H. S., Dec 1990. Intelligent tutoring systems: an overview. *Artificial Intelligence Review* 4 (4), 251–277.
- Nye, B. D., Pavlik, P. I., Windsor, A., Olney, A. M., Hajeer, M., Hu, X., 2018. Skope-it (shareable knowledge objects as portable intelligent tutors): overlaying natural language tutoring on an adaptive learning system for mathematics. *International journal of STEM education* 5 (1), 12.
- OECD, S., 2017. Number of magnetic resonance imaging (mri) units in selected countries as of 2017 (per million population).
- Oh, S., Carluccio, G., Collins, C., 2011. Method and tool for improved, rapid n-gram average sar determination. In: *Proc Intl Soc Mag Reson Med*. Vol. 19. p. 3868.
- Ohlsson, S., 1992. Information-processing explanations of insight and related phenomena. *Advances in the psychology of thinking* 1, 1–44.
- Overall, W., Pauly, J., 2007. An extensible, graphical environment for pulse sequence design and simulation. *Int. Soc. Magn. Reson. Med.*, Berlin, Germany.
- Padgaonkar, A. J., Lawson, S. M., King, A. I., 1978. Anatomical coordinate systems for human body segments. In: *Proceedings of the Human Factors Society Annual Meeting*. Vol. 22. SAGE Publications Sage CA: Los Angeles, CA, pp. 676–679.
- Pane, J. F., Griffin, B. A., McCaffrey, D. F., Karam, R., 2014. Effectiveness of cognitive tutor algebra i at scale. *Educational Evaluation and Policy Analysis* 36 (2), 127–144.
- Papachristos, D., Alafodimos, N., 09 2011. The use of an educative model of asynchronous web based education (wbe) in the paramedical education.
- Paquette, L., Lebeau, J.-F., Beaulieu, G., Mayers, A., 2015. Designing a knowledge representation approach for the generation of pedagogical interventions by mts. *International Journal of Artificial Intelligence in Education* 25 (1), 118–156.
- Peredo, R., Canales, A., Menchaca, A., Peredo, I., 2011. Intelligent web-based education system for adaptive learning. Vol. 38. Elsevier, pp. 14690–14702.

- Polson, M. C., Richardson, J. J., 2013. Foundations of intelligent tutoring systems. Psychology Press.
- Pruessmann, K. P., Weiger, M., Scheidegger, M. B., Boesiger, P., 1999. Sense: sensitivity encoding for fast mri. *Magnetic resonance in medicine* 42 (5), 952–962.
- Rabi, I. I., Zacharias, J. R., Millman, S., Kusch, P., 1938. A new method of measuring nuclear magnetic moment. *Physical Review* 53 (4), 318.
- Ramos-Llordén, G., den Dekker, A. J., Steenkiste, G. V., Jeurissen, B., Vanhevel, F., Audekerke, J. V., Verhoye, M., Sijbers, J., 2017. A unified maximum likelihood framework for simultaneous motion and t1 estimation in quantitative MR t1 mapping. *IEEE Trans Med Imag* 36 (2), 433–466.
- Razzaq, L., Patvarczki, J., Almeida, S. F., Vartak, M., Feng, M., Heffernan, N. T., Koedinger, K. R., 2009. The assistent builder: Supporting the life cycle of tutoring system content creation. *IEEE Transactions on Learning Technologies* 2 (2), 157–166.
- Rosenberg, R., 1987. A critical analysis of research on intelligent tutoring systems. *Educational Technology* 27 (11), 7–13.
- Royston, J. P., 1982. Algorithm AS 181: The W Test for Normality. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 31 (2), 176–180.
- Rus, V., Arthur, C. G., 2009. The question generation shared task and evaluation challenge. In: The University of Memphis. National Science Foundation. Citeseer.
- Rus, V., Stefanescu, D., Niraula, N., Graesser, A. C., 2014. Deeptutor: Towards macro-and micro-adaptive conversational intelligent tutoring at scale. In: Proceedings of the first ACM conference on Learning@ scale conference. ACM, pp. 209–210.
- Sabourin, J., Shores, L., Mott, B., C. Lester, J., 06 2012. Understanding and predicting student self-regulated learning strategies in game-based learning environments. *International Journal of Artificial Intelligence in Education* 23.
- Sauro, J., Lewis, J. R., 2016. Quantifying the user experience: Practical statistics for user research. Morgan Kaufmann.
- Schoeffel, R., et al., 2003. The concept of product usability. *ISO bulletin* 34 (3), 6–7.
- Schulze, K. G., Shelby, R. N., Treacy, D. J., Wintersgill, M. C., VanLehn, K., Gertner, A., 09 2000. Andes: An intelligent tutor for classical physics. *Journal of Electronic Publishing* 6.
- Self, J., 1990. Theoretical foundations for intelligent tutoring systems. *Journal of Artificial Intelligence in Education* 1 (4), 3–14.
- Sleeman, D., Brown, R. J., 1982. Intelligent tutoring systems. Academic Press.

- Sottolare, R., et al., 2012. Considerations in the development of an ontology for a generalized intelligent framework for tutoring. In: *International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference*. pp. 19–25.
- Stöcker, T., Vahedipour, K., Pflugfelder, D., Shah, N. J., 2010. High-performance computing mri simulations. *Magnetic Resonance in Medicine* 64 (1), 186–193.
- Stottler, R. H., Vinkavich, M., 2000. Tactical action officer intelligent tutoring system (tao its). In: "Training, Simulation and Education Conference (I/ITSEC 2000)". *Proceedings of I/ITSEC*.
- Suetens, P., 2017. *Fundamentals of medical imaging*. Cambridge university press.
- Tavakol, M., Dennick, R., 2011. Making sense of Cronbach's alpha. *International Journal of Medical Education* 2, 53–55.
- Torchiano, M., 2018. Effsize: Efficient Effect Size Computation. R package version 0.7.4.
- Torheim, G., Rinck, P. A., Jones, R. A., Kvaerness, J., 1994. A simulator for teaching mr image contrast behavior. *Magnetic Resonance Materials in Physics, Biology and Medicine* 2 (4), 515–522.
- Treceño Fernández, D., del Campo, J. C., de Luis García, R., Alberola López, C., 2016. Una implementación eficiente no paralela de secuencias de resonancia magnética mediante matrices sparse. In: *XXXIV CASEIB*. CASEIB, pp. 177–180.
- Tullis, T. S., Stetson, J. N., 6 2004. A comparison of questionnaires for assessing website usability. In: *Usability Professionals Association (UPA) 2004 Conference*. pp. 189–194.
- Turner, R., Le Bihan, D., Maier, J., Vavrek, R., Hedges, L. K., Pekar, J., 1990. Echo-planar imaging of intravoxel incoherent motion. *Radiology* 177 (2), 407–414.
- Ur, S., Vanlehn, K., 1995. Steps: A simulated, tutable physics student. *Journal of Artificial Intelligence in Education* 6, 405–437.
- Van Reeth, E., Rafiney, H., Tesch, M., Glaser, S. J., Sugny, D., 2016. Optimizing mri contrast with b1 pulses using optimal control theory. In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE, pp. 310–313.
- Vanlehn, K., 2006. The behavior of tutoring systems. *International journal of artificial intelligence in education* 16 (3), 227–265.
- VanLehn, K., 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist* 46 (4), 197–221.
- Vanlehn, K., Jones, R., 12 1991. Learning physics via explanation-based learning of correctness and analogical search control. pp. 110–114.



- Vanlehn, K., Ohlsson, S., Nason, R., 1994. Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education* 5, 135–175.
- Weragama, D., Reye, J., 2013. The php intelligent tutoring system. In: *International Conference on Artificial Intelligence in Education*. Springer, pp. 583–586.
- Wolfe, C. R., Widmer, C. L., Reyna, V. F., Hu, X., Cedillos, E. M., Fisher, C. R., Brust-Renck, P. G., Williams, T. C., Vannucchi, I. D., Weil, A. M., 2013. The development and analysis of tutorial dialogues in autotutor lite. *Behavior research methods* 45 (3), 623–636.
- Woolf, B. P., 2010. *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.
- Xanthis, C. G., Venetis, I. E., Chalkias, A., Aletras, A. H., 2013. Mrisimul: a gpu-based parallel approach to mri simulations. *IEEE Transactions on Medical Imaging* 33 (3), 607–617.
- Zhi-Pei Liang, P. C. L., 2000. *Principles of magnetic resonance imaging: a signal processing perspective*. IEEE Press Series on Biomedical Engineering.