

PLATAFORMA DE SIMULACIÓN VIRTUAL EN 3D PARA LA FORMACIÓN DE PROFESIONALES EN TÉCNICAS ECOGRÁFICAS



VNiVERSiDAD
D SALAMANCA

TRABAJO DE FIN DE GRADO
INGENIERÍA INFORMÁTICA

Guillem Hernández León

Dirigido por
Juan Alberto García Esteban
y
Vidal Moreno Rodilla

Salamanca, septiembre 2022

Dr. Vidal Moreno Rodilla, Dr. Juan Alberto García Esteban Profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

CERTIFICAN:

Que el trabajo titulado “PLATAFORMA DE SIMULACIÓN VIRTUAL EN 3D PARA LA FORMACIÓN DE PROFESIONALES EN TÉCNICAS ECOGRÁFICAS” ha sido realizado por D. Guillem Hernández León, con DNI 41650756X y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Grado en Ingeniería Informática de esta Universidad.

Y para que conste a todos los efectos oportunos.

En Salamanca, 07 de septiembre 2022

Dr. Vidal Moreno Rodilla
Dpto. de Informática y Automática
Universidad de Salamanca

Dr. Juan Alberto García Esteban
Dpto. de Informática y Automática
Universidad de Salamanca

Agradecimientos

Agradezco este trabajo a:

Mi madre por obligarme a esforzarme en este trabajo.

Fernandito que siempre confió en mi persona.

Jose y Dacosta por aguantar mis quejas.

Jorge Pirruki por enseñarme que es una ecografía.

Resumen:

Con la llegada de nuevas tecnológicas, es necesario actualizar nuestros conocimientos y nuestras formas de entender estas. Es muy importante tener en mente que la tecnología esta para ayudarnos y poder mejorar nuestras vidas en muchos ámbitos.

La realidad virtual y los dispositivos que la procesa han sido desde su llegada una tecnología muy innovadora y fiel a lo que es sentir la realidad. Por eso en temas de simulación es un gran avance y se puede usar en gran variedad de campos.

Existen muchas formas distintas de instruir, algunas mas tradicionales que otras, pero siempre se una buena apuesta intentar enseñar con tecnológicas nuevas que te hagan sentir que lo que estas aprendiendo es los mas cercano a poner en practica esos conocimientos.

Nos referimos en este caso al ámbito sanitario, cuya enseñanza es a veces compleja y puede ser costosa. Una forma muy simple de poner solución a esto es crear simulaciones mediante realidad virtual de algún campo en especial que pueda resultar interesante, como por ejemplo algún tipo de cirugía, el diagnóstico de algunas patológicas, etc.

Uno de los campos donde se puede introducir simulaciones de una forma no muy compleja, es en las ecografías. Por lo que en este trabajo de fin de grado se explica el desarrollo de un sistema que simula y posibilita simulaciones de exploración del cuerpo humano y exploraciones ecográficas mediante varios modelos del cuerpo humano.

Como resultado tenemos un sistema completo, donde se nos permite el uso de varias herramientas, donde es posible simular ecografías, exploraciones, activar y desactivar partes de modelos totalmente completos y mucho más en un entorno de realidad virtual.

Palabras clave: Realidad Virtual, cuerpo humano, ecografía, exploración ecográfica.

Summary:

With the arrival of new technologies, it is necessary to update our knowledge and our ways of understanding them. It is very important to keep in mind that technology is here to help us and to improve our lives in many areas.

Virtual reality and the devices that process it have been, since its arrival, a very innovative technology and faithful to what it is to feel reality. That is why in simulation terms it is a great advance and can be used in a wide variety of fields.

There are many different ways to instruct, some more traditional than others, but it is always a good bet to try to teach with new technologies that make you feel that what you are learning is the closest to putting that knowledge into practice.

We refer in this case to the health field, whose teaching is sometimes complex and can be expensive. A very simple way to solve this is to create virtual reality simulations of a particular field that may be interesting, such as some type of surgery, the diagnosis of some pathologies, etc.

One of the fields where simulations can be introduced in a not very complex way is in ultrasound. Therefore, in this end-of-degree project, the development of a system that simulates and enables simulations of the human body and ultrasound examinations through various models of the human body is explained.

As a result we have a complete system, where we are allowed to use several tools, where it is possible to simulate ultrasounds, explorations, activate and deactivate parts of complete models and much more in a virtual reality environment.

Keywords: Virtual reality, human body, ultrasound, ultrasound examination.

Tabla de contenidos

1.	Introducción	1
1.1	Motivación	1
2.	Objetivos del proyecto	3
2.1	Objetivos del Software	3
2.1.1	Funciones básicas en realidad virtual	3
2.1.2	Simulación exploración ecográfica.....	3
2.1.3	Exploración del modelo del cuerpo humano	3
2.1.4	Modificación del modelo del cuerpo humano	3
2.1.5	Interfaces intuitivas	4
2.1.6	Generar modelos según fichero XML	4
2.2	Objetivos técnicos del desarrollador	4
2.2.1	Conocimientos de programación en realidad virtual.....	4
2.2.2	Conocimientos en Unity	5
2.2.3	Conocimientos para herramientas en el ámbito sanitario.....	5
3.	Conceptos teóricos	6
3.1	Realidad Virtual	6
3.2	Robot Da Vinci.....	6
3.3	Ecografía	7
4.	Técnicas y herramientas.....	8
4.2	C#.....	8
4.3	WebXR	8
4.4	Modelo-Vista-Controlador	9
4.5	Proceso Unificado Ágil.....	9
4.6	Ficheros XML	10
4.8	Visual Paradigm	10
4.9	EZ Estimate	10
5.	Aspectos relevantes del desarrollo del proyecto	11
5.1	Gestión temporal y estimación del esfuerzo.....	11

5.2	Especificación de requisitos del software.....	14
5.2.1	Actores del sistema	15
5.2.2	Objetivos del sistema.....	15
5.2.3	Requisitos de información.....	15
5.2.4	Requisitos no funcionales.....	15
5.2.5	Requisitos funcionales.....	16
5.3	Especificación de Diseño.....	18
5.4	Desarrollo del proyecto.....	21
5.4.1	Aprendizaje del dispositivo de realidad virtual y WebXR.....	21
5.4.2	Procesamiento de ficheros XML	22
5.4.3	Visor de ecografías.....	23
5.4.4	Simulación de minicámara.....	24
5.5	Funcionamiento general del sistema	26
5.5.1	Interfaz de opciones	26
5.5.2	Botón Tutorial	28
5.5.3	Botón Modelo	28
5.5.4	Menú Activar/Desactivar Partes.....	28
5.5.5	Menú de Herramientas	29
5.5.6	Teletransporte	32
5.5.7	Centrar Cámara.....	33
6.	Conclusiones	34
7.	Líneas de trabajo futuras	35
7.1	Mejora del rendimiento	35
7.2	Incluir más XML.....	35
7.3	Mas herramientas.....	35
7.4	Simulaciones ecográficas más fieles	35
7.5	Entornos más estéticos	35
8.	Bibliografía.....	36

Lista de figuras

Figura 1. Robot Da Vinci.....	6
Figura 2. Ecografía real	7
Figura 3. Ejemplo fase inicio proceso unificado ágil	11
Figura 4. Diagrama de Gantt 1.....	12
Figura 5. Diagrama de Gantt 2.....	12
Figura 6. Diagrama de Gantt 3.....	12
Figura 7. Factores de complejidad técnica.....	13
Figura 8. Factores de complejidad del entorno	13
Figura 9. Complejidad de actores y casos de uso.....	14
Figura 10. Paquetes del sistema.....	16
Figura 11. Paquete gestión usuario	16
Figura 12. Paquete gestión modelo	17
Figura 13. Paquete gestión simulación exploración	17
Figura 14. Paquetes de diseño del sistema	18
Figura 15. Paquete Modelo.....	19
Figura 16. Paquete Controladores	19
Figura 17. Paquete View.....	20
Figura 18. Ejemplo diagrama de secuencia UC-0002	20
Figura 19. Ejemplo disposición cámara del ecógrafo	24
Figura 20. Ejemplo disposición minicámara	25
Figura 21. Interfaz principal.....	26
Figura 22. Interfaz de opciones.....	27
Figura 23. Ejemplo "ScrollBar" Altura Modelo	27
Figura 24. Botón Tutorial	28
Figura 25. Interfaz botón Modelo	28
Figura 26. "Toggles" por defecto.....	29
Figura 27. Interfaz de herramientas	30
Figura 28. Icono Ecógrafo.....	30
Figura 29. Ejemplo simulación exploración ecográfica.....	30
Figura 30. Icono minicámara.....	31
Figura 31. Ejemplo minicámara y visor	31
Figura 32. Icono Visor.....	31
Figura 33. Texto de "Zoom"	32
Figura 34. Icono aguja	32
Figura 35. Ejemplo "LineRenderer" teletransportación.....	32

Lista de tablas

Este documento no contiene tablas.

1. Introducción

En la actualidad existen muchas formas, técnicas y mecanismos distintos dedicados a la enseñanza de futuros trabajadores en el campo sanitario. Siempre hemos visto las técnicas más tradicionales, pero a lo largo de los últimos años con las mejoras tecnológicas han surgido nuevas herramientas para instruir en este campo.

Al ser un campo donde se tienen que tomar muchos riesgos y puede tener una dificultad elevada, una de las mejores opciones es crear simulaciones lo más fieles posibles para dar un modelo y poner en práctica los conocimientos requeridos.

Este trabajo de fin de grado consiste en una simulación de exploraciones ecográficas y exploración del cuerpo humano en un entorno de realidad virtual. Pese a no ser una disciplina que destaque por el riesgo, puede ser que no se cuente con las herramientas necesarias para este tipo de exploraciones. Estas exploraciones son posibles gracias a un modelo del cuerpo humano muy fiel proporcionado por la Universidad de Salamanca.

Al sumergirnos en un entorno de realidad virtual se puede instruir de una manera más simple y lo más fiel posible, ya que crea una experiencia totalmente inmersiva para el usuario. Esto será posible mediante un dispositivo de realidad virtual, como serían unas *Oculus*.

El sistema desarrollado está inspirado en los robots *Da Vinci* de una forma más simplificada e intuitiva. Este cuenta con ecógrafos, microcámaras, visores y elementos meramente estéticos para una experiencia más detallada.

1.1 Motivación

El desarrollo de este proyecto se ha basado en el interés de aprender a programar sistemas con funcionalidades para dispositivos de realidad virtual. Al ser una tecnología altamente extendida en los últimos años y con el gran abanico de posibilidades que ofrece, es una buena opción tener una base de conocimientos sobre este ámbito.

El desarrollo en Unity del proyecto también ha sido una motivación durante todo el proceso. Tanto como para aprender en un nivel más alto este entorno de desarrollo, como para aplicar los conocimientos ya aprendidos anteriormente.

Aprender las posibilidades que puede tener la creación de herramientas para la enseñanza de gente en el ámbito sanitario, ha sido una aportación que enriquece mucho la forma de entender el desarrollo para nuevas tecnologías y técnicas que en un futuro pueden ser muy útiles.

2. Objetivos del proyecto

2.1 Objetivos del Software

El sistema se basa en una aplicación que simula en tiempo real exploraciones ecográficas y exploraciones del mismo cuerpo humano en un entorno de realidad virtual. Teniendo en cuenta esto, se presentan los objetivos que debe cumplir nuestro sistema.

2.1.1 Funciones básicas en realidad virtual

Al estar inmersos en un entorno de realidad virtual, el sistema debe cumplir los requisitos mínimos para poder interactuar con este.

Funciones básicas hace referencia a movimiento por el entorno, fácil acceso a los menús y visualización correcta de los elementos.

2.1.2 Simulación exploración ecográfica

Este es uno de los objetivos más importantes del sistema y que le da la razón de ser. Una vez nosotros tengamos elegido el ecógrafo, al entrar en contacto con la piel, se nos mostrará una simulación de una ecografía en tiempo real del modelo del cuerpo humano.

No es una ecografía real, pero es lo más fiel posible a ella con las características que ofrece Unity.

2.1.3 Exploración del modelo del cuerpo humano

Al tener un modelo del cuerpo humano con todos sus elementos, es normal que el sistema permita explorar todo lo que este tiene. Para esto hay una herramienta de una microcámara que grabará en tiempo real lo que se ve desde la punta de esta herramienta. Para visualizar el contenido, deberemos activar los visores correspondientes

2.1.4 Modificación del modelo del cuerpo humano

Como es posible que el usuario no esté interesado en visualizar o explorar todas las partes del cuerpo humano, el sistema permitirá al usuario desactivar o reactivar ciertas

partes. Esto se hace para que el usuario se centre en el contenido que más le interese.

También se permitirá hacer ciertos ajustes del cuerpo como escalado, modificación de altura, etc.

2.1.5 Interfaces intuitivas

El sistema cuenta con varias opciones y distintos menús accesibles tanto desde el mismo ordenador como desde el dispositivo de realidad virtual.

Esta desarrollado de una forma que llegar hasta los menús e interactuar con ellos sea fácil e intuitivo, incluso aunque haya menús que se acceden desde distintos puntos.

2.1.6 Generar modelos según fichero XML

El sistema deberá ser capaz de instanciar ciertas partes de este según la información de los ficheros XML proporcionados. De esta forma se puede automatizar algunos procesos.

Deberá ser capaz de instanciar los modelos, y desactivar las partes no importantes para ese modelo en concreto. Esto se logrará leyendo los distintos nodos que tienen los XML y procesando esa información.

También deberá crear los menús de “*Toggles*”, leyendo también la información de los ficheros XML.

2.2 Objetivos técnicos del desarrollador

El desarrollador, como ya se ha comentado previamente, buscaba aprender sobre ciertos temas más en profundidad, como sería el entorno de desarrollo usado, técnicas de simulación, etc.

2.2.1 Conocimientos de programación en realidad virtual

Al ser una tecnología muy innovadora con muchas oportunidades, entender su uso y las bases de programación para estos dispositivos puede ser muy útil para futuras aplicaciones que consistan en estos entornos.

2.2.2 Conocimientos en Unity

Unity es conocido por ser un entorno de desarrollo muy usado. Este ofrece muchas posibilidades a la vez que resulta interesante tener un buen uso, debido a que es posible que muchos proyectos futuros lo usen.

2.2.3 Conocimientos para herramientas en el ámbito sanitario

Para el desarrollo del proyecto ha sido necesario la búsqueda de mucha información relacionado con aspectos sanitarios, por lo que han dado al autor un buen entendimiento de estos y una visión más clara sobre todas las posibilidades que puede ofrecer desarrollar herramientas para la ayuda en este ámbito.

3. Conceptos teóricos

En este apartado se presentarán los distintos puntos de información que han sido necesarios para el desarrollo del sistema.

3.1 Realidad Virtual

Se entiende como realidad virtual un entorno donde se muestran ciertas escenas y elementos, donde el usuario está completamente inmerso. Nos permite crear todo tipo de escenarios, por lo que existen posibilidades casi ilimitadas e interacciones muy interesantes.

Se puede crear cualquier tipo de sistema, donde el usuario podrá ser parte del mismo. Este es un concepto muy innovador que ofrece experiencias nuevas y de cierta forma “reales”.

Hay muchos ámbitos donde esta tecnología puede resultar muy útil, como por ejemplo en cualquier tipo de simulación donde el mismo usuario podrá experimentar una alta fidelidad.

3.2 Robot Da Vinci

Es un dispositivo robótico articulado, que controla el cirujano mediante una consola y ciertos controles. El cirujano lo que hace es operar mediante estas articulaciones. Ofrece una precisión muy alta para operaciones.

Existe cierto grado de inmersión en este dispositivo, debido a la torre de visión desde donde se observan las articulaciones y el sistema de visión estereoscópica que ofrece.



Figura 1. Robot Da Vinci

El sistema desarrollado se ha basado en un funcionamiento simplificado de estos dispositivos. Donde las herramientas elegidas serán como las articulaciones mecanizadas que el cirujano controla, pero en este caso se controlarán mediante los mandos del dispositivo de realidad virtual.

3.3 Ecografía

Una ecografía es un método de diagnóstico por imagen, en la cual se usan ultrasonidos como media para ver los cortes de la parte que deseemos y diagnosticar o descartar patologías. Estas ondas sonoras rebotan y al volver es cuando se generan las imágenes. Se usan en varios campos como por ejemplo visualizar un embarazo, detectar tumores, etc.

En el sistema desarrollado se ha intentado implementar y simular de la forma más fiel posible una ecografía, pese a las limitaciones que tiene el entorno de desarrollo.



Figura 2. Ecografía real

Aunque en el sistema no podamos ver una ecografía como tal, podremos observar los “cortes” de lo que se vería si usásemos un ecógrafo real sobre un paciente.

4. Técnicas y herramientas

En este apartado se explica cuáles son las distintas herramientas y mecanismos usados para el desarrollo del sistema y su implementación.

4.1 Unity

Unity es un entorno y motor de desarrollo con una interfaz gráfica bastante intuitiva, pensado para la creación de videojuegos. Cuenta con una gran variedad de bibliotecas que permiten un desarrollo muy extenso. Se puede operar sobre interfaces 3D y también en 2D, en este caso se ha usado Unity 3D.

La interfaz gráfica hace que este entorno sea el más adecuado para el desarrollo de este sistema, además de soportar bibliotecas para programar interacciones con el dispositivo de realidad virtual.

Unity usa por defecto el lenguaje de programación C#, por lo que el sistema entero está programado con este lenguaje.

4.2 C#

Como se ha comentado en el subapartado anterior, C# viene por defecto en Unity.

Este es un lenguaje de programación orientado a objetos, que permite a los desarrolladores crear una gran variedad de proyectos seguros y robustos.

Es un lenguaje multiparadigma, cuya procedencia deriva de una evolución de los lenguajes de programación C y C++.

4.3 WebXR

WebXR es una biblioteca que brinda a Unity acceso a los dispositivos realidad virtual y realidad aumentada. Este permite desarrollar y alojar experiencias de realidad virtual y realidad aumentada en la web.

En este caso el sistema se desarrolló pensando en poder tener una opción de ejecutable en la web. Todas las funciones que trae WebXR también se pueden usar en un programa ejecutable sin necesidad de tener que usarse dentro de la web.

Se ha usado para hacer las interacciones básicas con los mandos de dispositivo de realidad virtual, como el movimiento por el entorno, accionar los menús, las funcionalidades de las distintas herramientas, etc.

4.4 Modelo-Vista-Controlador

Modelo-Vista-Controlador es un patrón de desarrollo de software que se usa para organizar el proyecto en 3 apartados distintos.

- 1- Modelo: Contiene los datos que maneja sistema.
- 2- Controlador: Conecta con el modelo y la vista y es el encargado de actuar sobre el flujo de información del sistema.
- 3- Vista: Se encarga de la presentación y es la parte que interactúa con el usuario.

Es usado para tener bien diferenciadas todas las partes que componen el sistema y poder automatizar ciertos procesos. También hará que el sistema se entienda de una forma muchos más fácil para otros desarrolladores que puedan querer usar el sistema.

4.5 Proceso Unificado Ágil

El proceso unificado ágil es un marco de trabajo, altamente usado en el desarrollo de proyectos software. Como características que lo diferencia del proceso unificado, usa las técnicas ágiles conocidas como “Test Driven Development”, “Model Driven Development” y la gestión del software.

Dividido en 4 fases: Inicial, Elaboración, Construcción y Transición Fases divididas a su vez en Disciplinas: Modelado de Negocio, Requisitos, Análisis, Diseño, Implementación, Pruebas

Cuenta con las siguientes características:

- 1- Iterativo e incremental: la división de este marco de trabajo hace que después de cada una de las iteraciones, el producto “aumenta” y este mas cerca del resultado final.
- 2- Dirigido por casos de uso: Estos casos de uso son usado para obtener los requisitos que se desarrollan en cada una de las iteraciones a través de las distintas disciplinas.

- 3- Centrado en la arquitectura: Se dice que un único modelo no puede satisfacer todas las necesidades de un sistema software, por lo que se utilizan varios para este.

4.6 Ficheros XML

Es un lenguaje de marcado, que sirve para crear etiquetas y almacenar datos. Unity, que usa C# tiene funciones para poder leer estos ficheros XML sin problema.

En el sistema se usan estos ficheros para almacenar el nombre de los modelos a instanciar, las partes que se tienen que eliminar y también las partes que deben ser activables o desactivables con los “*Toggles*” mediante la creación de un menú.

4.7 Microsoft Project

Este es un programa destinado a la gestión de proyecto. Se encarga de gestiones temporales, administración de carga de trabajo, de recursos y de presupuestos.

Este programa se ha usado para la administración temporal hecha en el anexo 1, para crear los diagramas de Gantt.

4.8 Visual Paradigm

Este es una herramienta software para el modelado de UML.

Se ha usado durante todo el desarrollo del proyecto para generar todos los diagramas necesarios para explicar el proyecto. Diagramas de casos de uso, diagramas de paquetes, diagramas de clases de diseño y diagramas de secuencia.

4.9 EZEstimate

Este es una herramienta utilizada para la estimación de costes y temporal. Hace uso de los factores de complejidad del entorno y técnica y también la complejidad de los casos de uso y de los actores.

Se usa en el primer anexo para hacer estimaciones sobre la entrega del proyecto.

5. Aspectos relevantes del desarrollo del proyecto

5.1 Gestión temporal y estimación del esfuerzo

Para una correcta planificación y distribución del tiempo ha sido necesario organizar las cargas de trabajo del sistema según varios parámetros.

Lo ideal es hacer una planificación temporal, según un calendario acorde a nuestro tiempo que resulta en un diagrama de Gantt y también según el esfuerzo o dificultad que conlleva cada acción que tiene que desarrollar el sistema, como serían los casos de uso.

La primera parte fue organizar el desarrollo del sistema bajo un marco de trabajo. En este caso, como ya se ha comentado se hace uso del proceso unificado ágil, muy parecido al proceso unificado normal. Por consiguiente, cada uno de los pasos a realizar se enlazó con cada una de las fases que tienen este proceso unificado (inicio, elaboración y construcción) y a su vez con la disciplina que le corresponde.

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1		Inicio	26 días	lun 14/02/22	vie 22/04/22		
2		Modelado de Negocio	6 días	lun 14/02/22	mar 01/03/22		
3		Programación para Oculus	2 días	lun 14/02/22	jue 17/02/22		Guillem Hernández León
4		Repaso Unity	2 días	vie 18/02/22	mié 23/02/22		Guillem Hernández León
5		Aprendizaje XML	2 días	jue 24/02/22	mar 01/03/22		Guillem Hernández León
6		Requisitos	5 días	mié 02/03/22	mar 15/03/22		
7		Definición de los Objetivos	1 día	mié 02/03/22	jue 03/03/22		Guillem Hernández León
8		Definición de Requisitos No funcionales	2 días	vie 04/03/22	mié 09/03/22	7	Guillem Hernández León
9		Definición de Requisitos Funcionales	2 días	jue 10/03/22	mar 15/03/22	8	Guillem Hernández León
10		Análisis	5 días	mié 16/03/22	mar 29/03/22		
11		Análisis de los Objetivos	1 día	mié 16/03/22	jue 17/03/22		Guillem Hernández León
12		Análisis de Requisitos No Funcionales	2 días	vie 18/03/22	mié 23/03/22	11	Guillem Hernández León
13		Análisis de Requisitos Funcionales	2 días	jue 24/03/22	mar 29/03/22	12	Guillem Hernández León
14		Diseño	9 días	mié 30/03/22	vie 22/04/22		
15		Crear Documentos Técnicos de Requisitos	3 días	mié 30/03/22	mié 06/04/22		Guillem Hernández León
16		Diseño de Subsistemas	2 días	jue 07/04/22	mar 12/04/22		Guillem Hernández León
17		Diseño de las Clases de la Arquitectura	4 días	mié 13/04/22	vie 22/04/22	16	Guillem Hernández León

Figura 3. Ejemplo fase inicio proceso unificado ágil

Estableciendo un calendario estándar, explicado en el anexo 1, y establecidos también unos tiempos para cada uno de los pasos a hacer, como por ejemplo 2 días para aprender a programar para los dispositivos *Oculus*, Microsoft Project no da la opción de crear un Diagrama de Gantt que sirve para organizar en proyecto en tareas pequeñas y gestionables.

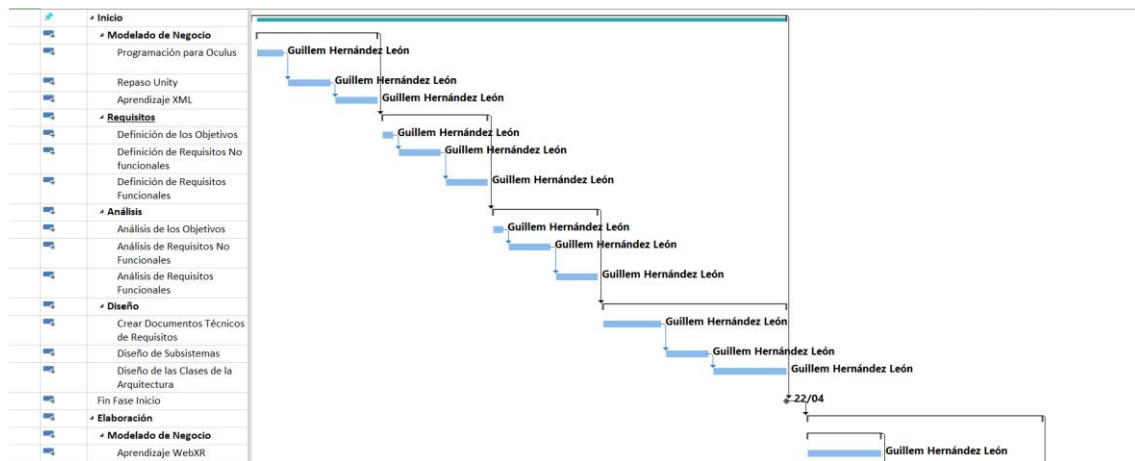


Figura 4. Diagrama de Gantt 1

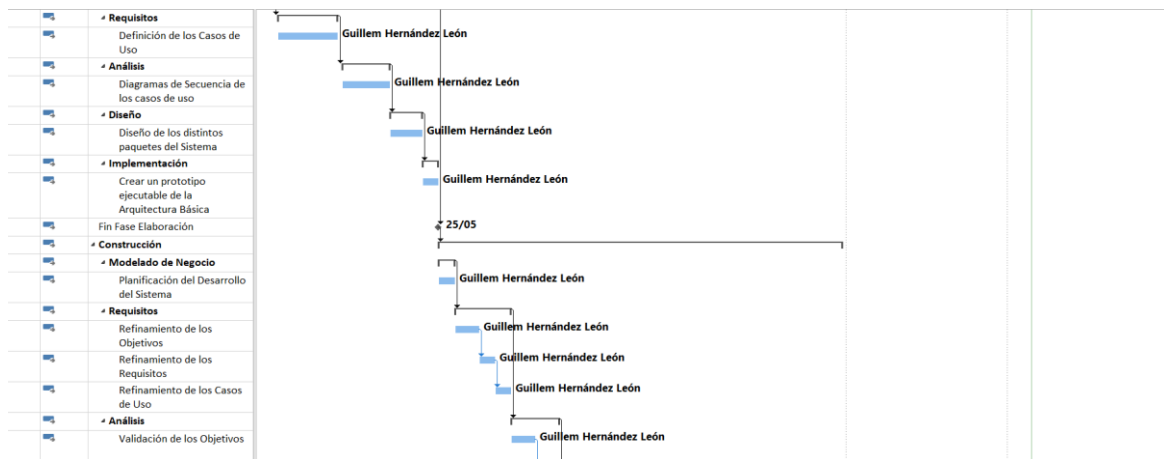


Figura 5. Diagrama de Gantt 2

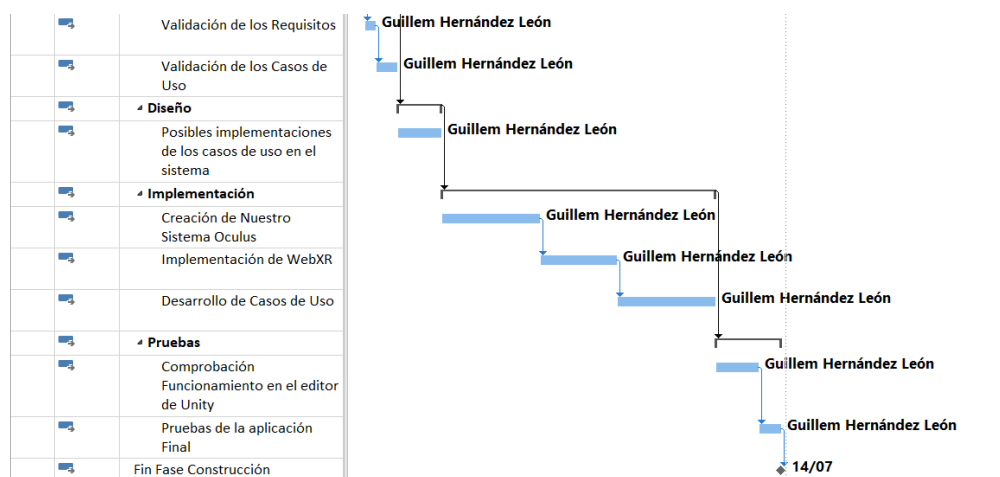
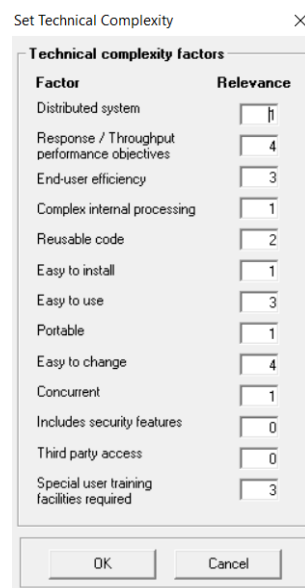


Figura 6. Diagrama de Gantt 3

En segundo lugar, corresponde la estimación del esfuerzo. Para este apartado ha sido necesario 3 puntos principales:

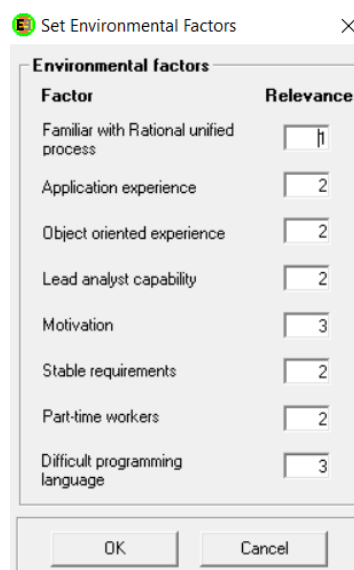
- 1- Factores de complejidad del entorno
- 2- Factores de complejidad técnica
- 3- Dificultad de los casos de uso y de los actores

Los factores de complejidad se explican en el anexo I y el encargado de los cálculos de los pesos de estos los hace el programa EZEstimate.



Factor	Relevance
Distributed system	1
Response / Throughput performance objectives	4
End-user efficiency	3
Complex internal processing	1
Reusable code	2
Easy to install	1
Easy to use	3
Portable	1
Easy to change	4
Concurrent	1
Includes security features	0
Third party access	0
Special user training facilities required	3

Figura 7. Factores de complejidad técnica



Factor	Relevance
Familiar with Rational unified process	1
Application experience	2
Object oriented experience	2
Lead analyst capability	2
Motivation	3
Stable requirements	2
Part-time workers	2
Difficult programming language	3

Figura 8. Factores de complejidad del entorno

EZEstimate - C:\Users\guill\Desktop\Visuals TFG\Estimacion de Costes.ezp

File Settings Help

Module
Default
Add Module Delete

Summary
Total Modules 1 **Excel Report** Generate Report
Use cases Simple 12 Average 5 Complex 0
Actors Simple 1 Average 0 Complex 1

Add Actor / Use case
Actor / Use case Name Select Type Complexity
Usecase Simple Add

Tech / Env Factors
Set Tech Factor
Set Env Factors

Estimation Summary
UAW 4
UUCW 110
UUPC = UAW + UUCW 114
TFactor 28
EFactor 8
TCF = 0.6 + (.01*TFactor) 0.88
EF = 1.4 + (-0.03*EFactor) 1.16
UCP = UUPC*TCT*EF 116,3712
Total Effort@ 10 Hrs/UCP 1163,712

Use case / Actor List (Double click to delete)

Id	Module	Type	Name	complexity
11	Default	Usecase	Simular explora...	Simple
12	Default	Usecase	Teletransportar ...	Simple
13	Default	Usecase	Centrar Cámara	Simple
14	Default	Usecase	Simular Microcá...	Average
15	Default	Usecase	Acercar/Alejar ...	Simple
16	Default	Usecase	Modificar Altura...	Simple
17	Default	Usecase	Modificar Dista...	Simple
18	Default	Usecase	Modificar Tama...	Simple
19	Default	Usecase	Modificar Modelo	Simple
2	Default	Actor	System	Simple
3	Default	Usecase	Leer Datos XML	Average
4	Default	Usecase	Cargar Modelo	Simple
5	Default	Usecase	Elegir Herramie...	Simple
6	Default	Usecase	Moverse Por El ...	Average
7	Default	Usecase	Activar/Desacti...	Average
8	Default	Usecase	Visualizar Tutorial	Simple
9	Default	Usecase	Crear Menú	Average

Figura 9. Complejidad de actores y casos de uso

Como se puede observar en la figura 9, la estimación del esfuerzo nos da un total de 1163, 712 horas.

5.2 Especificación de requisitos del software

Para poder empezar con el desarrollo del proyecto hay que recoger todos los requisitos que debe tener el sistema software. Para esto analizaremos: objetivos, requisitos de información, requisitos funcionales, requisitos no funcionales y actores del sistema.

A todo esto, también se le añadirá los casos de uso, que forman parte de los requisitos funcionales y los distintos paquetes que tendrá el sistema para dividir la realización de casos de uso.

5.2.1 Actores del sistema

Se identifican los distintos actores que tendrá el sistema. Al ser un sistema simple sin conexión a la red, solo intervendrán el usuario y el sistema.

El usuario será la persona que interactuará con el sistema y el mismo sistema será el encargado de administrar todos los datos.

5.2.2 Objetivos del sistema

En este apartado se identifica los distintos objetivos de que deberá cumplir el sistema.

Como ya se ha comentado algunos de estos objetivos, son simulaciones y exploraciones del cuerpo humano sobre un entorno de realidad virtual creado por el desarrollador.

5.2.3 Requisitos de información

Estos contienen la información que deberá alojar el sistema para su correcto funcionamiento.

Estos requisitos tienen los datos básicos de nuestro sistema, como por ejemplo los datos de las simulaciones o los datos de los ficheros XML.

5.2.4 Requisitos no funcionales

Estos requisitos determinan la calidad del sistema y también sus operaciones.

Estos requisitos se tienen que hacer en base a que características queremos que nuestro sistema cumpla a sabiendas de lo que podemos proporcionar.

En el caso de este sistema, se ha tenido en cuenta 2 requisitos no funcionales muy importantes. El primero sería que el sistema sea intuitivo, la tener que usar dispositivos de realidad virtual es posible que pueda llegar a ser un poco confuso por lo que se ha intentado que tanto los controles como los destinos acceso a menús sean lo mas simples posibles. El segundo sería el rendimiento, al tener modelos con un gran peso es normal que el sistema a veces funcione un poco lento. Por lo que la eficiencia en el sistema es un requisito que se debe ver a fondo.

5.2.5 Requisitos funcionales

Estos requisitos, como su propio nombre indica son los encargados de especificar las funcionalidades que tendrá el sistema software. Estos requisitos se determinan mediante casos de uso, explicados a fondo en el anexo II

Antes de definir estos casos de uso hay que tener bien definidos los actores y los paquetes del sistema para agrupar estos casos de uso.

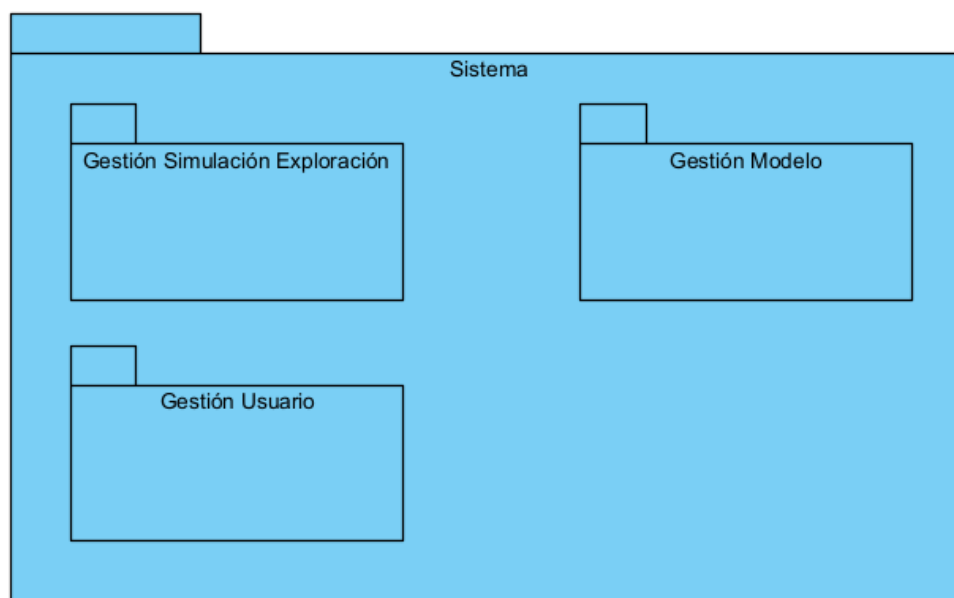


Figura 10. Paquetes del sistema

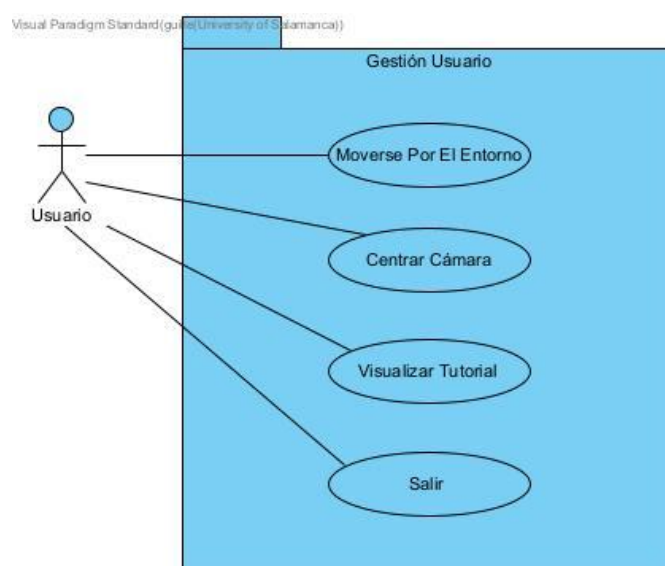


Figura 11. Paquete gestión usuario

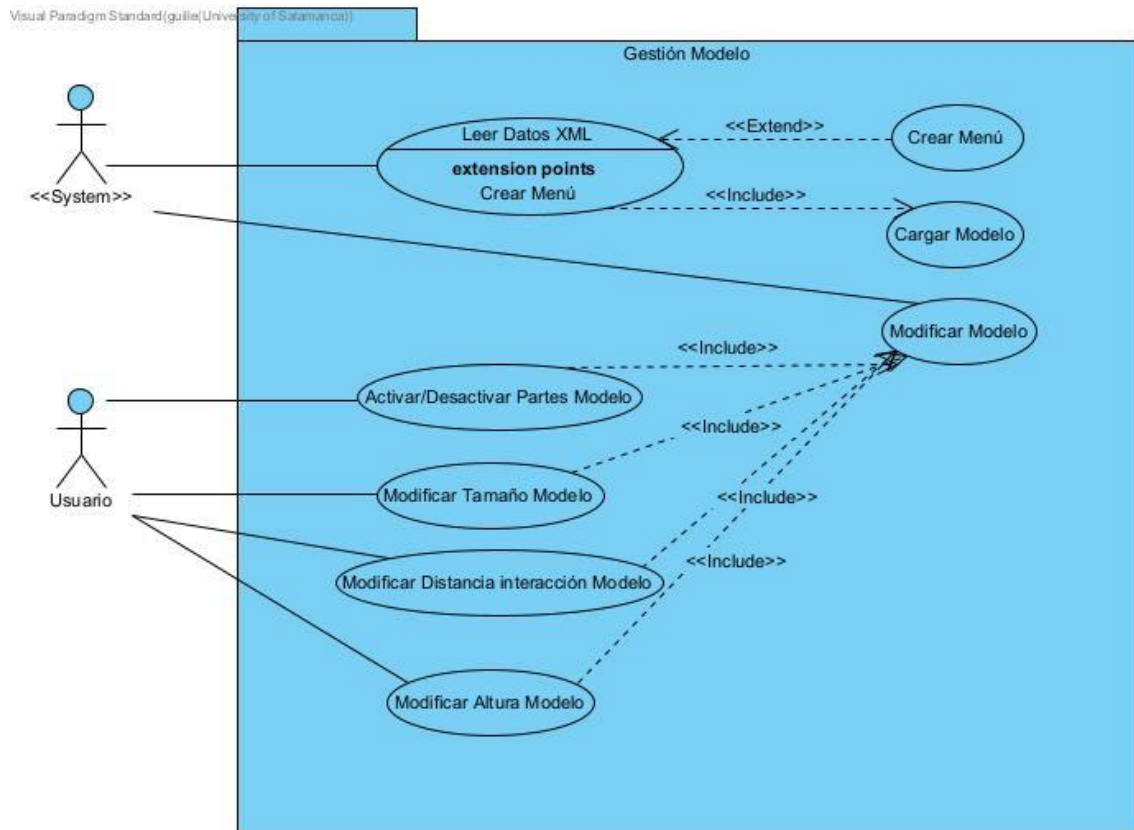


Figura 12. Paquete gestión modelo

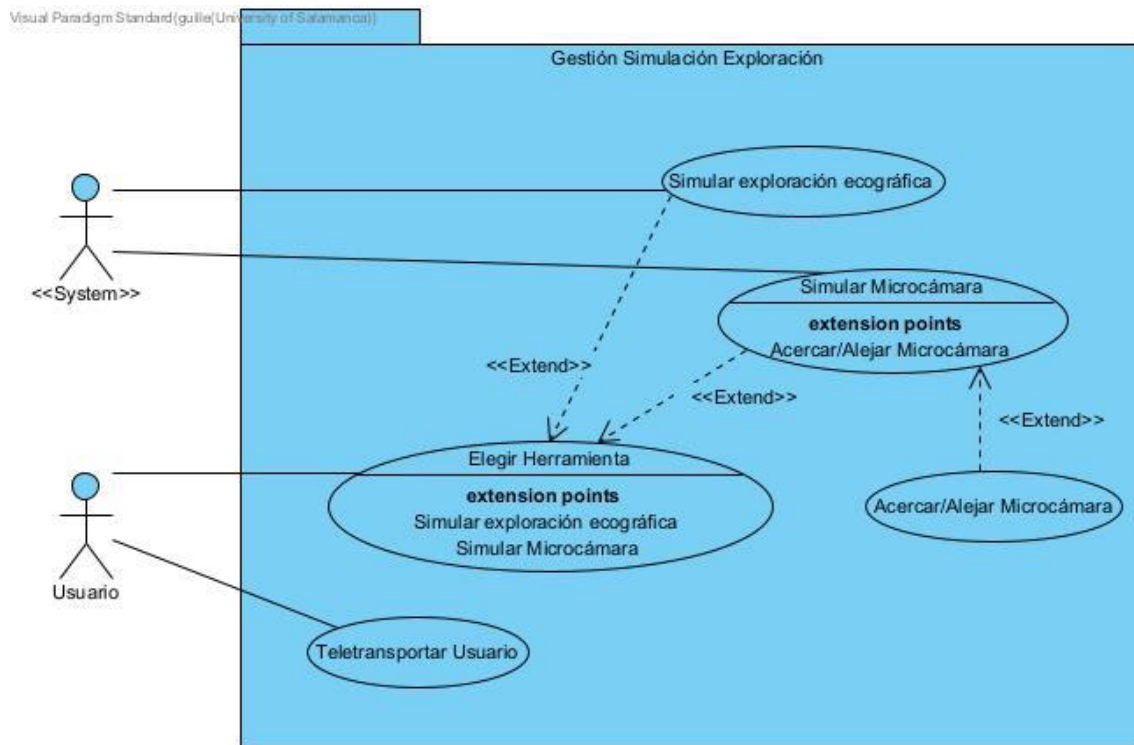


Figura 13. Paquete gestión simulación exploración

5.3 Especificación de Diseño

En este apartado se especifica un modelo de análisis y la realización de los casos de uso mediante los casos de uso definidos anteriormente.

Definiremos los distintos paquetes que tiene el sistema mediante diagramas de clases y también diagramas de secuencias para los casos de uso más importantes y destacados.

Al usar el patrón Modelo-Vista-Controlador, nuestros paquetes de diseño constaran de estos tres apartados mas los objetos 3D que se encuentran en la escena con la cual el usuario interactúa.

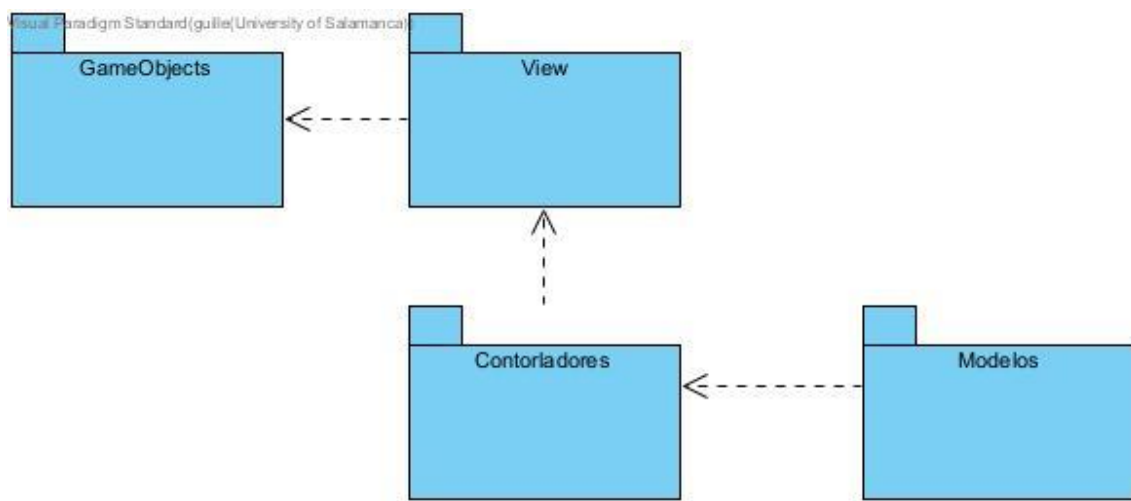


Figura 14. Paquetes de diseño del sistema

Una vez definidos los paquetes de nuestro patrón de diseño software, haremos hincapié en cada uno de estos diseñando unos diagramas de clases de diseño según los datos y métodos que contiene nuestro sistema. Todos estos diagramas se encuentran explicados en el anexo III.

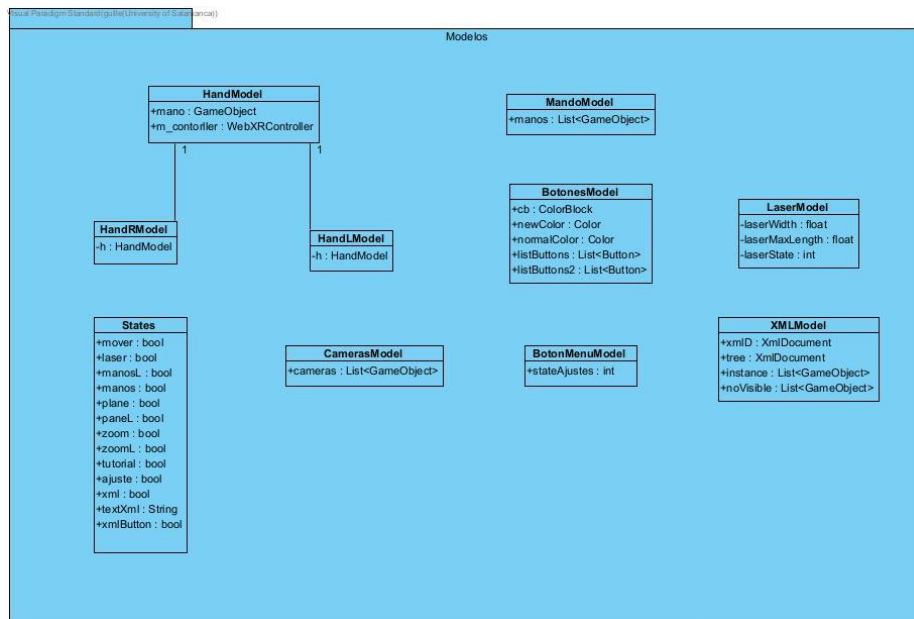


Figura 15. Paquete Modelo

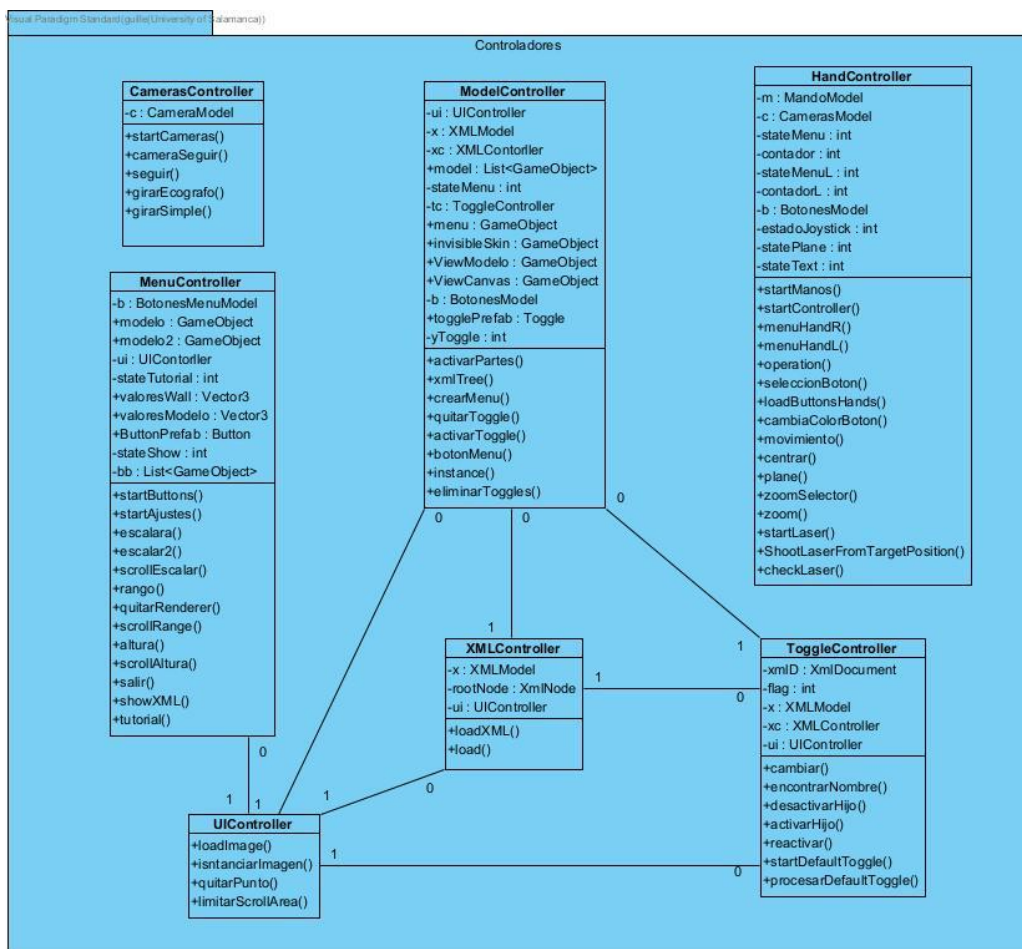


Figura 16. Paquete Controladores

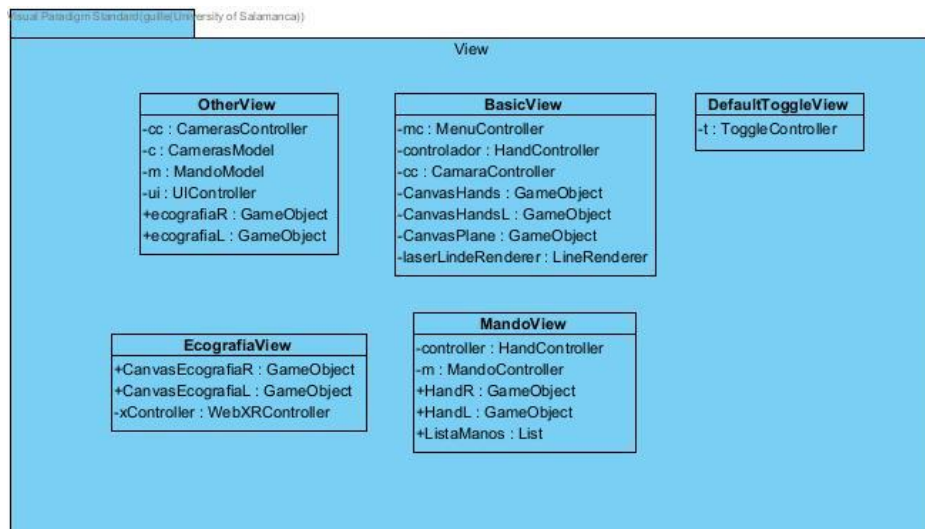


Figura 17. Paquete View

En estos diagramas se puede observar lo que representan todos los scripts que confirman la totalidad del programa, con sus variables, parámetros y métodos.

Una vez ya definidas todas las clases y paquetes de diseño, hay que centrarse en la realización de casos de uso mediante unos diagramas de secuencia haciendo referencia a estos paquetes implementados anteriormente.

Existe uno para cada caso de uso, pero solo se han identificado y desarrollado los más notables.

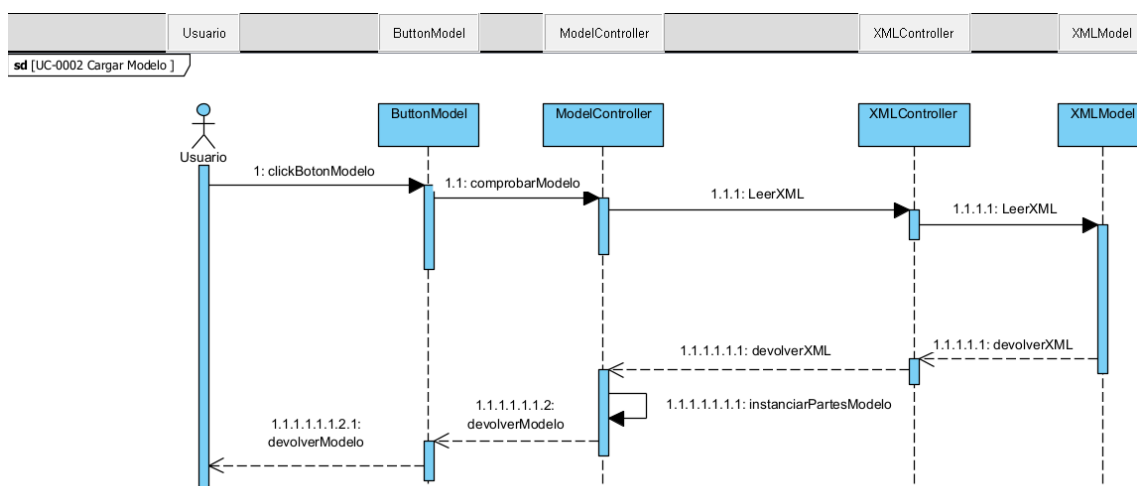


Figura 18. Ejemplo diagrama de secuencia UC-0002

5.4 Desarrollo del proyecto

En este apartado se explicará en orden las partes más importantes del desarrollo del proyecto junto con las distintas dificultades, complicaciones y riesgos que se han presentado a lo largo de esta etapa.

5.4.1 Aprendizaje del dispositivo de realidad virtual y WebXR

Desde un comienzo el sistema había estado pensado para desarrollarse con un dispositivo de realidad virtual, para dar una experiencia única e innovadora en el campo de las simulaciones. Por lo que en primera instancia para poder comenzar el proyecto fue necesario aprender y tener un buen manejo del dispositivo de realidad virtual. Para poder hacer un sistema con unas buenas especificaciones y un buen uso de los controles definidos es necesario entender la totalidad del dispositivo que se usará en el mismo.

Durante toda la implementación del sistema se ha hecho uso de unas *Oculus Rift*, cuya dificultad de manejo no es muy elevada por lo que no es una tarea complicada tener unas bases de su funcionamiento.

Una vez se entiende el manejo de estos dispositivos y las posibilidades que ofrece, hay que aprender a usar las bibliotecas y funciones para programar sus funcionalidades. Existen bibliotecas como *SteamVR* que son altamente usadas y con muchas opciones, pero al querer hacer un sistema multiplataforma con la posibilidad de ejecutarlo desde la web, *SteamVR* lo hace mas complicado ya que tiene muchas incompatibilidades para esto. La alternativa que se ha usado ha sido *WebXR*.

Al final las funcionalidades que ofrece los mandos de estos dispositivos de realidad virtual son las acciones o eventos que ocurren cuando accionamos los distintos botones o joysticks que tienen. Por lo que esta biblioteca se usa para registrar cuando el usuario hace uso de cualquiera de sus botones.

Estos son algunos de los ejemplos:

```
Vector2 joystickR =  
HandRModel.h.m_controller.GetAxis2D(WebXRController.Axis2DTypes.Thumbstick);
```

Es importante decir que `m_controller` hace referencia al controlador de *WebXR*. Una vez dicho esto, esta orden lo que hace es registrar en un vector las coordenadas del joystick. Estas van de -1.0 a 1.0 y es usado en varios eventos como para moverse o navegar a través de los distintos menús.

```
if (HandRModel.h.m_controller.GetButtonDown(WebXRController.ButtonTypes.Trigger))
```

Lo que hace esta orden es comprobar si desde el mando derecho se ha pulsado el gatillo superior. A partir de ahí, en caso de que haya ocurrido y cumpliéndose ciertas circunstancias, procede un evento.

5.4.1.1 *Movimiento*

Para realizar el movimiento (en los ejes x y z) de las cámaras usando los mandos del dispositivo de realidad virtual, se tuvo que hacer lo siguiente:

Primero de todo recoger los valores de los joysticks del mando izquierdo:

```
Vector2 joystickL =  
HandLModel.h.m_controller.GetAxis2D(WebXRController.Axis2DTypes.Thumbstick);
```

Una vez tengamos estos valores en tiempo real, procedemos a usar estos valores para modificar las posiciones.

```
if (!States.laser && !States.manos && !States.manosL && !States.zoom &&  
    !States.zoomL)  
{  
    GameObject.Find("CamaraSeguir").transform.Translate(Vector3.forward *  
        (joystickL.y / 10));  
  
    GameObject.Find("CamaraSeguir").transform.Translate(Vector3.right * (joystickL.x  
        / 10));  
}
```

5.4.2 Procesamiento de ficheros XML

Como la información de los modelos con los que trabajar, tanto las partes que interesan para ese modelo como las que hay que invisibilizar, está organizada en ficheros XML. Ha sido necesario aprender desde Unity como procesar dichos ficheros.

Unity y C# cuentan con facilidades para esto usando la directiva: `using System.Xml;`

Principalmente hay que usar 2 funciones que son `LoadXml(string)` y `selectSingleNode(string)`. La primera función lo que hace es cargar en una variable `XmlDocument` el fichero XML organizado en nodos, en este caso pasaremos la ubicación del fichero en el proyecto. El segundo método lo que hace es cargar un nodo del XML que nosotros decidamos, con nodo nos referiremos a cualquier etiqueta:

`<practica name="">` en este caso el nombre del nodo sería "practica" y todo lo que viene después son atributos. En los atributos es donde se encuentra la información que se

quiere procesar. Para esto crearemos un bucle e iremos nodo por nodo recogiendo los atributos que se necesiten.

```
foreach (XmlNode tag in rootNode)
{
    if (tag.LocalName == "triDimModel"){
        Debug.Log("tag.Attributes.GetNamedItem("name").Value");
    }
}
```

En este bucle lo que se hace es iniciar mostrar en la consola todos los valores del atributo name del nodo "triDimModel" que este contenido en el nodo inicial. Explicado de una forma más simple sería como acceder a los atributos de los nodos hijos. De esta forma podemos obtener todos los valores que queramos siempre con las condiciones apropiadas.

5.4.3 Visor de ecografías

Esta parte fue una de las más complicadas de hacer, ya que el hecho de simular una exploración ecográfica no es fácil en ningún aspecto.

En un principio la idea fue anidar una cámara a la herramienta del ecógrafo y hacer una foto que se cargaría en una imagen que se instanciaría sobre esta herramienta. Esta opción era complicada de hacer por el hecho del script de realizar fotos (que luego también se tenían que cargar) y por el rendimiento, cada foto realizada hacía que el programa se parase un momento.

Después de usar la documentación de Unity, se decidió usar un "*Render Texture*" que lo que hace es posibilitar la creación de materiales los cuales contienen en tiempo real lo que esta "grabando la cámara". Entonces la imagen instanciada cuando queremos ver la simulación ecográfica estará en tiempo real.

El siguiente problema fue la disposición de la cámara. Como un corte no es visto exactamente desde arriba del ecógrafo, había que mover la cámara de su posición original. Primero fue necesario desplazarla hacia abajo y rotarla 90 grados en el eje x. De esta forma si da la sensación de estar viendo los cortes de forma correcta.

Finalmente, para que esta simulación fuera más fiel, se activó la opción de que las cámaras rendericen en escala de grises.

Aunque de forma lógica no sea lo mismo que una ecografía, debido a que no se pueden usar ultrasonidos, es lo más fiel que se permite hacer en Unity.



Figura 19. Ejemplo disposición cámara del ecógrafo

5.4.4 Simulación de minicámara

Este apartado está pensado para la exploración del modelo del cuerpo humano y tuvo una solución bastante más simple que la simulación ecográfica.

La idea principal de este es parecida a la del ecógrafo, una cámara anidada a una herramienta y que muestra en tiempo real lo que esta cámara renderiza.

La cámara anidada se encuentra en la punta de la herramienta, que es una aguja. En un principio la idea fue pasar la visión completa del visor de realidad virtual a esta cámara en la punta de la herramienta y que se controlaba mediante el mando que contiene esta herramienta. El problema que esto conlleva es que se pierde la noción de la posición y no se puede ver que se hace con las otras herramientas.

Entonces la solución a esto es implementar visores que muestren lo que se ve desde la cámara sin tener que cambiar la cámara principal que esta anidada al visor de realidad virtual.

Los visores son o imágenes o planos (objetos 3D) que contienen “*Render Textures*” de la cámara en cuestión. Existen 2 tipos de estos visores asociados a la simulación de la microcámara. El primero es una herramienta como tal, que solo funciona cuando tenemos la microcámara activa en la otra mano y el segundo es una que podemos accionar pulsando B/Y en el mando en el que tengamos la microcámara. Este ultimo aparecerá en la parte central superior del programa.



Figura 20. Ejemplo disposición minicámara

5.4.4.1 Zoom

Esta opción fue incluida al final como contenido extra para poder hacer una simulación más realista. Esta opción es activable cuando en ambas manos se tiene la combinación de la herramienta visor y de la herramienta de la microcámara. Un pequeño texto aparecerá en la parte superior izquierda indicando como accionar este zoom.

Consiste en cambiar de forma gradual el parámetro *FieldOfView* de la cámara según los valores recogidos en el joystick del mando en cuestión.

```
public void zoom()
{

    Vector2 joystickR =
    HandRModel.h.m_controller.GetAxis2D(WebXRController.Axis2DTypes.Thumbstick);

    Vector2 joystickL =
    HandLModel.h.m_controller.GetAxis2D(WebXRController.Axis2DTypes.Thumbstick);

    if (States.zoom)
    {
        c.cameras[3].GetComponent<Camera>().fieldOfView =
        Mathf.MoveTowards(c.cameras[3].GetComponent<Camera>().fieldOfView, 60,
        -joystickR.y / 8); // plano derecha mini cámara izquierda

        c.cameras[4].GetComponent<Camera>().fieldOfView =
        Mathf.MoveTowards(c.cameras[4].GetComponent<Camera>().fieldOfView, 60,
        -joystickR.y / 8); // plano derecha mini cámara izquierda
    }

    else if (States.zoomL)
    {
        c.cameras[1].GetComponent<Camera>().fieldOfView =
        Mathf.MoveTowards(c.cameras[1].GetComponent<Camera>().fieldOfView, 60,
        -joystickL.y / 8); // plano izquierda mini cámara derecha

        c.cameras[2].GetComponent<Camera>().fieldOfView =
        Mathf.MoveTowards(c.cameras[2].GetComponent<Camera>().fieldOfView, 60,
        -joystickL.y / 8); // plano izquierda mini cámara derecha
    }
}
```

5.5 Funcionamiento general del sistema

Como se ha comentado anteriormente el sistema funciona en su totalidad con un dispositivo de realidad virtual, de otra forma solo se pueden ver algunas interfaces con las cuales no se podrá interactuar de la forma correcta.

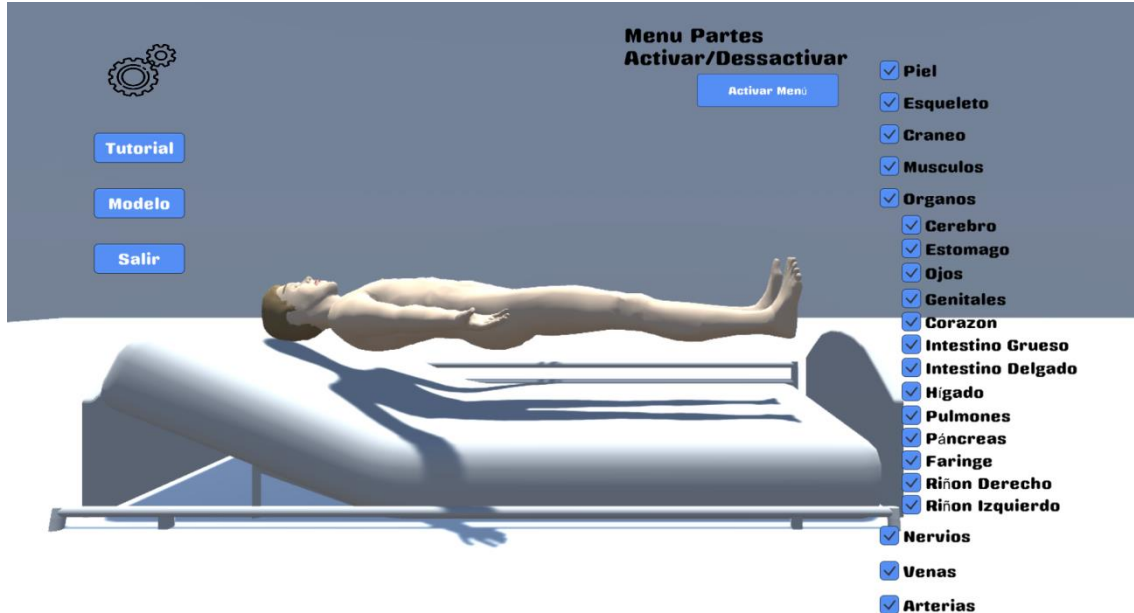


Figura 21. Interfaz principal

Es importante remarcar que, aunque el programa esté pensado para ejecutarse con este tipo de dispositivos, existen interfaces que solamente son accesibles desde el mismo ordenador. Concretamente hablamos de las opciones que se observan en la figura 21. Esto está hecho de esta forma debido a algunas dificultades que tiene crear menús y su navegación mediante realidad virtual.

5.5.1 Interfaz de opciones

Haciendo uso del botón que tenemos arriba a la izquierda accedemos a un menú de opciones. Donde permite al usuario modificar ciertos parámetros del modelo para personalizar la experiencia que que pueda ser más fácil para este hacer uso del sistema y de sus funcionalidades.

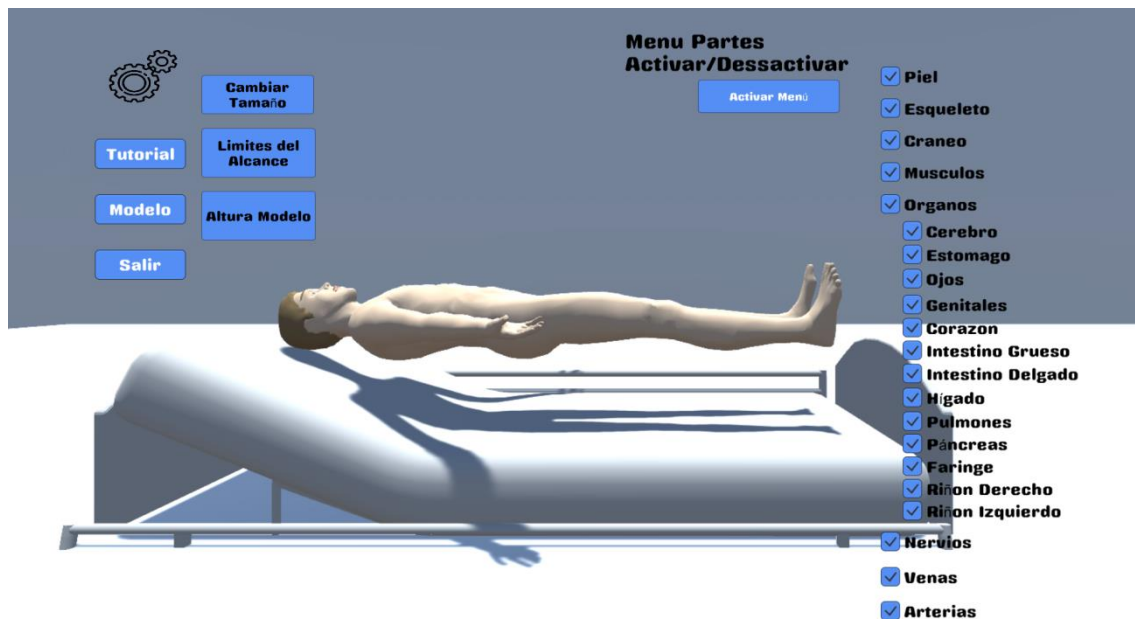


Figura 22. Interfaz de opciones

Concretamente tenemos 3 opciones que son: “Cambiar Tamaño”, “Limites del Alcance” y “Altura Modelo”

Cada una de las opciones activa una “ScrollBar” que deja modificar el parámetro en cuestión. “Cambiar Tamaño” permite escalar el modelo, “Limites de Alcance” permite modificar un rectángulo asociado al modelo para que el usuario no se meta dentro del mismo. De esta forma se puede hacer más pequeño para acercarse más al modelo o más grande. Finalmente “Altura Modelo” cambiar el valor del eje Y del modelo.

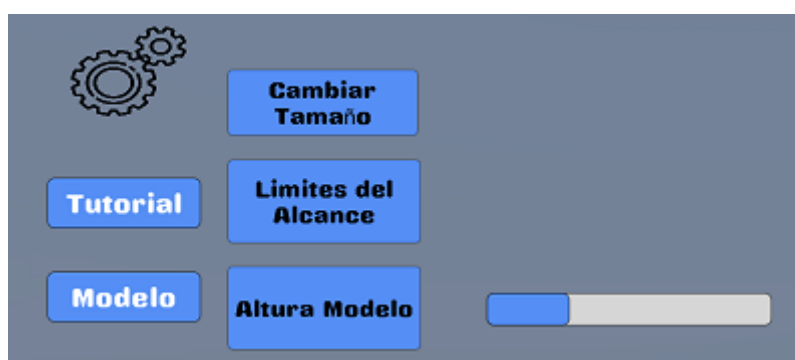


Figura 23. Ejemplo "ScrollBar" Altura Modelo

5.5.2 Botón Tutorial

Botón que muestra un pequeño tutorial de los controles básicos.

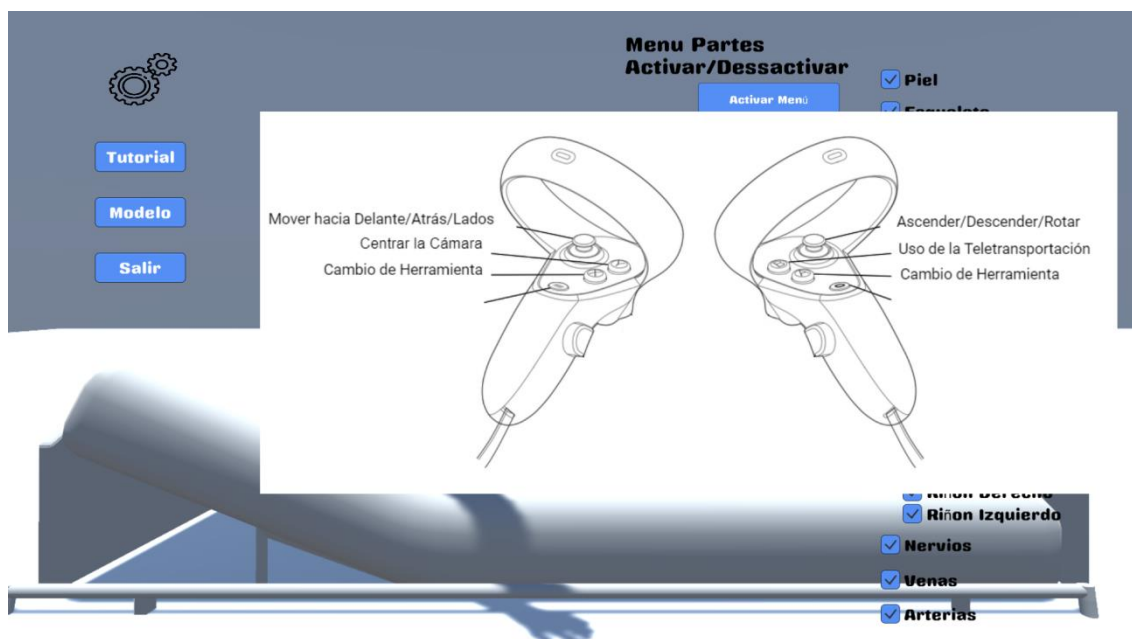


Figura 24. Botón Tutorial

5.5.3 Botón Modelo

Botón que instancia otros botones con las distintas opciones de modelos del cuerpo humano que encontramos almacenados como ficheros XML dentro del sistema.

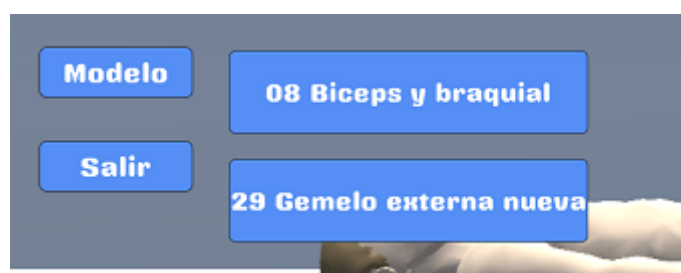


Figura 25. Interfaz botón Modelo

5.5.4 Menú Activar/Desactivar Partes

Este menú está compuesto por distintas partes dependiendo de que modelo estemos viendo.

En un principio el sistema cuenta con un modelo completo y con unos “*Toggles*” ya instanciados que siempre están cuando se ve ese modelo en cuestión.



Figura 26. "Toggles" por defecto

Estos "Toggles" siempre se encuentran y están por el único motivo de que existe la posibilidad que con tantas partes el sistema tenga un bajo rendimiento. Por lo que se recomienda desactivar algunas de estas partes no importantes para el usuario.

La segunda parte de este menú solo se puede activar cuando se carga un modelo desde el botón "Modelo" visto anteriormente y también pulsando el botón "Activar Menú". Esto se hace debido a que contiene muchas partes que se deben tener en cuenta para activar o desactivar y por consiguiente se instancian muchos de estos "Toggles", lo cual puede llegar a ser molesto visualmente para el usuario.

5.5.5 Menú de Herramientas

Este menú ya es solamente accesible desde los mandos del dispositivo de realidad virtual, pulsando el botón A/X dependiendo de en qué mando queramos la herramienta.

Para navegar por este menú usaremos los joysticks del mando con el que hayamos accionado el menú. Si queremos volver a la herramienta estándar (que es una mano simple) volveremos a accionar los botones comentados anteriormente.

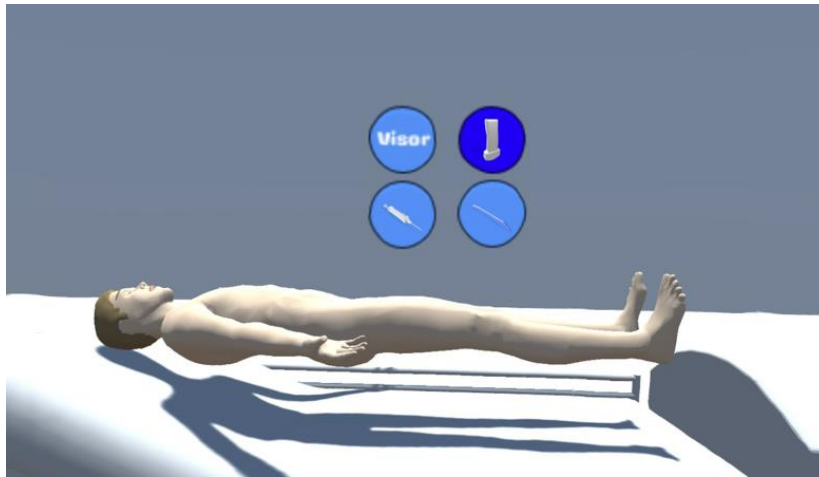


Figura 27. Interfaz de herramientas

5.5.5.1 Ecógrafo

Esta herramienta nos va a permitir hacer simulaciones de exploraciones ecográficas cuando accionemos el gatillo superior del mando con el que tengamos esta herramienta y la herramienta esté en contacto con la piel.



Figura 28. Icono Ecógrafo

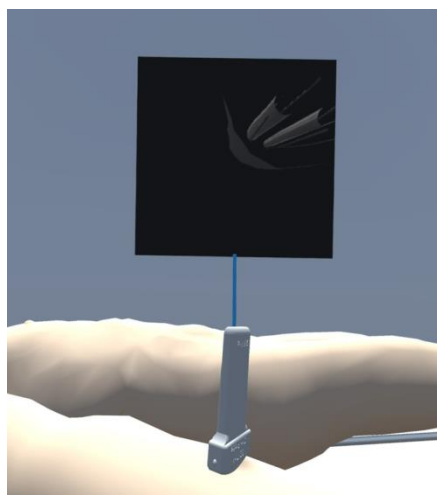


Figura 29. Ejemplo simulación exploración ecográfica

5.5.5.2 Minicámara

Esta herramienta nos permitirá ver mediante varios visores lo que la cámara renderiza. Esta cámara está anidada a la punta de la herramienta.



Figura 30. Icono minicámara

Para poder visualizar lo que renderiza la cámara, tenemos 2 opciones: la primera será poner en la otra mano la herramienta “Visor” que veremos a continuación y la segunda será accionar el botón B/Y dependiendo de que mando estemos usando y aparecerá un pequeño visor en la parte central superior.

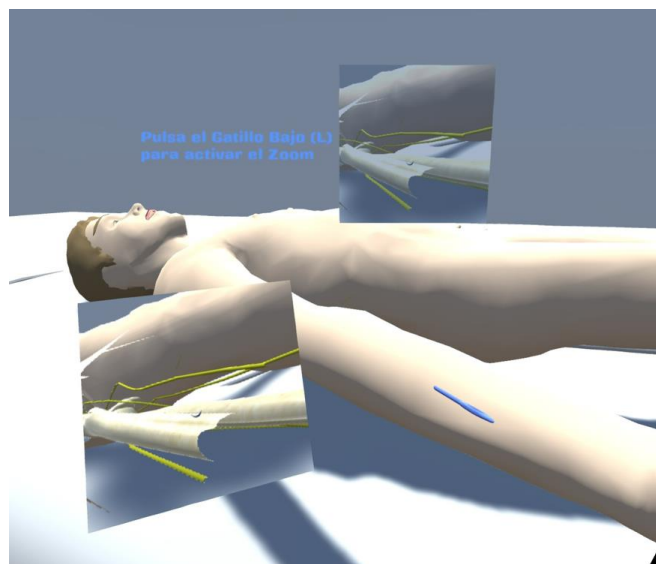


Figura 31. Ejemplo minicámara y visor

5.5.5.3 Visor

Esta herramienta únicamente servirá para que el usuario pueda ver lo que renderiza la Cámara de la herramienta “Minicámara”. Si se usa sola no tiene utilidad.

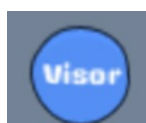


Figura 32. Icono Visor

Cuando están activas a la vez las herramientas de “Visor” y “Microcámara” aparecerá la opción de usar el “Zoom”. Este se acciona pulsando el gatillo bajo del mando donde tengamos el visor. Para acercar o alejar se usara el joystick del mismo mando con el que se accionado el “Zoom”.



Figura 33. Texto de "Zoom"

5.5.5.4 Aguja

Esta herramienta es meramente decorativa y sirve para poder hacer las simulaciones más fieles y tener más herramientas con las que experimentar.

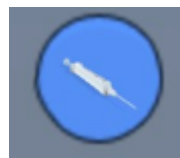


Figura 34. Icono aguja

5.5.6 Teletransporte

El sistema da opción de teletransportarse en caso de que el usuario lo necesite aunque con los joysticks se puede mover de una forma simple e intuitiva. Para activar esta opción simplemente el usuario deberá pulsar el botón B del mando derecho. Una vez hecho esto aparecerá un “LineRenderer” que apuntará hacia donde este apuntando el mando y en caso de que se enfoque al suelo aparecerá una pequeña base circular que indica donde irá el usuario. Si este pulsa el gatillo superior del mando se trasladará a la posición apuntada.

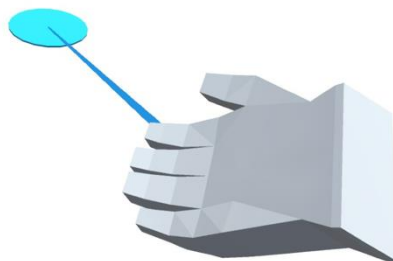


Figura 35. Ejemplo "LineRenderer" teletransportación

5.5.7 Centrar Cámara

Existe la posibilidad que el visor de realidad virtual no este acorde al eje X del objeto que se mueve por el entorno. Esto hace que los movimientos sean complicados y no se dirijan hacia donde el usuario desea.

Para esto, si se acciona el botón Y del mando izquierdo, se alinean estos ejes y los movimientos volverán a ser normales.

6. Conclusiones

Este proyecto se inició con la finalidad de entender y observar todas las posibilidades, ventajas y limitaciones que podía ofrecer el mundo de realidad virtual en el ámbito de las simulaciones.

El proyecto empezó con pocos conocimientos sobre todas las ventajas que puede llevar esta tecnología, pero a medida que avanzaba esté fue muy fácil ver que contiene posibilidades casi ilimitadas y es una inversión muy potente para el futuro.

Como es comprensible una tecnología tan innovadora tiene mucho que ofrecer al mundo en muchos ámbitos, pero esta vez nos hemos centrado en el ámbito sanitario. Los médicos u otro personal sanitario no siempre podrán tener a mano ciertas herramientas que les permita poderse instruir en algunos campos, por lo que es una buena opción contar con un dispositivo que pueda simular cualquier tipo de operación o técnicas de diagnóstico.

En un principio fue complicado marcar el alcance de este proyecto debido a que en este tipo de simulaciones se pueden incluir muchas cosas y no suele ser simple aun más sin conocimientos en este ámbito. Pero gracias a compañeros y documentación las ideas se fueron aclarando y esto se ha refinado en el sistema que se puede observar.

Simular exploraciones ecográficas tampoco ha sido una tarea fácil, existen muchas maneras de hacerlo y al no poder hacerlo con exactitud debido a la falta de recursos complicaba las cosas. Al fin y al cabo, como bien indica su nombre, es una simulación por lo que aunque no parezca exactamente igual hay que intentar ser los más fiel posible a la realidad, cosa que se ha conseguido después de numerosas pruebas.

En resumidas cuentas, este es un proyecto ambicioso que ha conseguido cumplir con una gran parte de los objetivos marcados desde un inicio y resolviéndolos de la forma más original posible que haga que el usuario tenga experiencias nuevas. Ha sido una gran oportunidad para aprender sobre tecnológicas que se puedan usar ampliamente en un futuro no muy lejano y para entender la importancia de simulaciones en ciertos ámbitos.

7. Líneas de trabajo futuras

El sistema está completo con los objetivos que se establecieron desde un inicio, pero como en cualquier producto software es posible hacer ciertas mejoras.

En este apartado se presentan algunas de las posibles acciones que se podrían realizar para mejorar el sistema.

7.1 Mejora del rendimiento

El sistema cuenta con modelos muy pesados, con lo que es posible que a veces tenga un rendimiento muy bajo. Buscar algún método para que el modelo pueda estar instanciado de forma completa, pero no afecte tanto a la eficiencia de este.

7.2 Incluir más XML

El sistema tiene un proceso automatizado para cargar modelos desde ficheros XML y crear sus menús. Siempre que se incluyan en el proyecto de Unity podrán existir opciones ilimitadas.

7.3 Mas herramientas

Incluir más herramientas podría mejorar de forma exponencial el sistema y la experiencia para el usuario. Podrían coexistir distintos tipos de simulaciones dentro del sistema.

7.4 Simulaciones ecográficas más fieles

Aunque las simulaciones hechas en el sistema son altamente fieles, es posible mejorarlas y hacer que se parezcan más a una real.

7.5 Entornos más estéticos

El sistema cuenta con un entorno muy básico, es posible que para mejorar la experiencia del usuario se puedan añadir elementos estéticos o con laguna utilidad.

8. Bibliografía

- Documentación de Unity:
<https://docs.unity3d.com/es/2019.4/Manual/ManualVersions.html>
- Documentación de WebXR: <https://github.com/De-Panther/unity-webxr-export>
- Ingeniería del Software I y II
- Animación Digital
- Gestión de Proyectos
- Tutorial Programación Orientada a Objetos C#:
https://www.youtube.com/watch?v=C95ca4nJHGs&ab_channel=Programacion101.net