



ESCUELA DE INGENIERÍA DE FUENLABRADA

GRADO EN INGENIERIA EN SISTEMAS  
AUDIOVISUALES Y MULTIMEDIA

**TRABAJO FIN DE GRADO/MÁSTER**

ENTORNO INMERSIVO 3D CON TECNOLOGÍAS WEB

Autor : Jorge Luis Grande González

Tutor : Dr. Gregorio Robles

Curso académico 2023/2024



# **Trabajo Fin de Grado/Máster**

Entorno Inmersivo 3D con Tecnologías Web

**Autor :** Jorge Luis Grande González

**Tutor :** Dr. Gregorio Robles

La defensa del presente Proyecto Fin de Carrera se realizó el día                      de  
de 202X, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

Fuenlabrada, a                      de                      de 202X



*Dedicado a  
mi madre*



# Agradecimientos

Aquí vienen los agradecimientos... Aunque está bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.





# Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.



# Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Sección . . . . .	1
1.1.1. Estilo . . . . .	2
1.2. Estructura de la memoria . . . . .	4
<b>2. Objetivos</b>	<b>5</b>
2.1. Objetivo general . . . . .	5
2.2. Objetivos específicos . . . . .	5
2.3. Planificación temporal . . . . .	5
<b>3. Estado del arte</b>	<b>7</b>
3.1. Lenguajes de Marcado y Programación . . . . .	7
3.2. Importaciones de Bibliotecas JavaScript . . . . .	8
3.3. Herramientas de Desarrollo . . . . .	8
<b>4. Diseño e implementación</b>	<b>9</b>
4.1. Arquitectura general . . . . .	9
4.2. Escena . . . . .	12
4.3. Controles de Realidad Virtual . . . . .	14
4.3.1. Configuración de los Controladores VR . . . . .	14
4.3.2. Interacción Mediante Controladores VR . . . . .	15
4.3.3. Implementación Técnica . . . . .	15
<b>5. Experimentos y validación</b>	<b>17</b>

<b>6. Resultados</b>	<b>19</b>
6.1. Descripción de las Interacciones . . . . .	19
6.1.1. Interacción con UA1 . . . . .	19
6.1.2. Interacción con el Proxy . . . . .	20
6.1.3. Interacción con UA2 . . . . .	24
6.1.4. Interacción con paquetes . . . . .	25
<b>7. Conclusiones</b>	<b>29</b>
7.1. Consecución de objetivos . . . . .	29
7.2. Aplicación de lo aprendido . . . . .	29
7.3. Lecciones aprendidas . . . . .	30
7.4. Trabajos futuros . . . . .	30
<b>A. Manual de usuario</b>	<b>31</b>

# Índice de figuras

1.1. Página con enlaces a hilos . . . . .	3
1.2. Estructura del parser básico . . . . .	4
3.1. Porcentaje de lenguajes usados. . . . .	7
4.1. Estructura del proyecto . . . . .	12
4.2. Escena del proyecto . . . . .	14
4.3. Controlador VR . . . . .	15
4.4. Botones controlador VR . . . . .	16
6.1. Estado inicial del UA1 . . . . .	21
6.2. Atributos UA1 con valores específicos . . . . .	22
6.3. Estado inicial del Proxy . . . . .	23
6.4. Atributos Proxy con valores específicos . . . . .	24
6.5. Estado inicial del UA2 . . . . .	26
6.6. Atributos UA2 con valores específicos . . . . .	27
6.7. Secuencia del intercambio de paquetes . . . . .	28





# Capítulo 1

## Introducción

En este capítulo se introduce el proyecto. Debería tener información general sobre el mismo, dando la información sobre el contexto en el que se ha desarrollado.

No te olvides de echarle un ojo a la página con los cinco errores de escritura más frecuentes<sup>1</sup>.

Aconsejo a todo el mundo que mire y se inspire en memorias pasadas. Las memorias de los proyectos que he llevado yo están (casi) todas almacenadas en mi web del GSyC<sup>2</sup>.

En mayo de 2023 me apunté a un curso de innovación docente donde nos pidieron hacer un podcast con temática docente. Aproveché entonces para hacer un podcast de unos 30 minutos donde en los primeros quince minutos introducía LaTeX y la memoria, y en los segundos hacía hincapién en aquellas cosas que más os cuestan utilizar en la memoria: las figuras, las tablas y las citas. Podéis escuchar el podcast en Internet<sup>3</sup>.

### 1.1. Sección

Esto es una sección, que es una estructura menor que un capítulo.

Por cierto, a veces me comentáis que no os compila por las tildes. Eso es un problema de codificación. Al guardar el archivo, guardad la codificación de “ISO-Latin-1” a “UTF-8” (o viceversa) y funcionará.

---

<sup>1</sup><http://www.tallerdeescritores.com/errores-de-escritura-frecuentes>

<sup>2</sup><https://gsyc.urjc.es/~grex/pfcs/>

<sup>3</sup><https://podcasters.spotify.com/pod/show/gregorio-robles9/episodes/>

Tu-memoria-de-Trabajo-Fin-de-Grado-o-de-Mster-en-LaTeX-e23hucr/a-a58kp2

### 1.1.1. Estilo

Recomiendo leer los consejos prácticos sobre escribir documentos científicos en L<sup>A</sup>T<sub>E</sub>X de Diomidis Spinellis<sup>4</sup>.

Lee sobre el uso de las comas<sup>5</sup>. Las comas en español no se ponen al tuntún. Y nunca, nunca entre el sujeto y el predicado (p.ej. en “Yo, hago el TFG” sobre la coma). La coma no debe separar el sujeto del predicado en una oración, pues se cortaría la secuencia natural del discurso. No se considera apropiado el uso de la llamada coma respiratoria o *coma criminal*. Solamente se suele escribir una coma para marcar el lugar que queda cuando omitimos el verbo de una oración, pero es un caso que se da de manera muy infrecuente al escribir un texto científico (p.ej. “El Real Madrid, campeón de Europa”).

A continuación, viene una figura, la Figura 1.1. Observarás que el texto dentro de la referencia es el identificador de la figura (que se corresponden con el “label” dentro de la misma). También habrás tomado nota de cómo se ponen las “comillas dobles” para que se muestren correctamente. Nota que hay unas comillas de inicio (“) y otras de cierre (”), y que son diferentes. Volviendo a las referencias, nota que al compilar, la primera vez se crea un diccionario con las referencias, y en la segunda compilación se “rellenan” estas referencias. Por eso hay que compilar dos veces tu memoria. Si no, no se crearán las referencias.

A continuación un bloque “verbatim”, que se utiliza para mostrar texto tal cual. Se puede utilizar para ofrecer el contenido de correos electrónicos, código, entre otras cosas.

```
From gaurav at gold-solutions.co.uk  Fri Jan 14 14:51:11 2005
From: gaurav at gold-solutions.co.uk  (gaurav_gold)
Date: Fri Jan 14 19:25:51 2005
Subject: [Mailman-Users] mailman issues
Message-ID: <003c01c4fa40$1d99b4c0$94592252@gaurav7klgnyif>
```

Dear Sir/Madam,

How can people reply to the mailing list? How do i turn off  
this feature? How can i also enable a feature where if someone  
replies the newsletter the email gets deleted?

Thanks

---

<sup>4</sup><https://github.com/dspinellis/latex-advice>

<sup>5</sup><http://narrativabreve.com/2015/02/opiniones-de-un-corrector-de-estilo-11-recetas-par>  
html

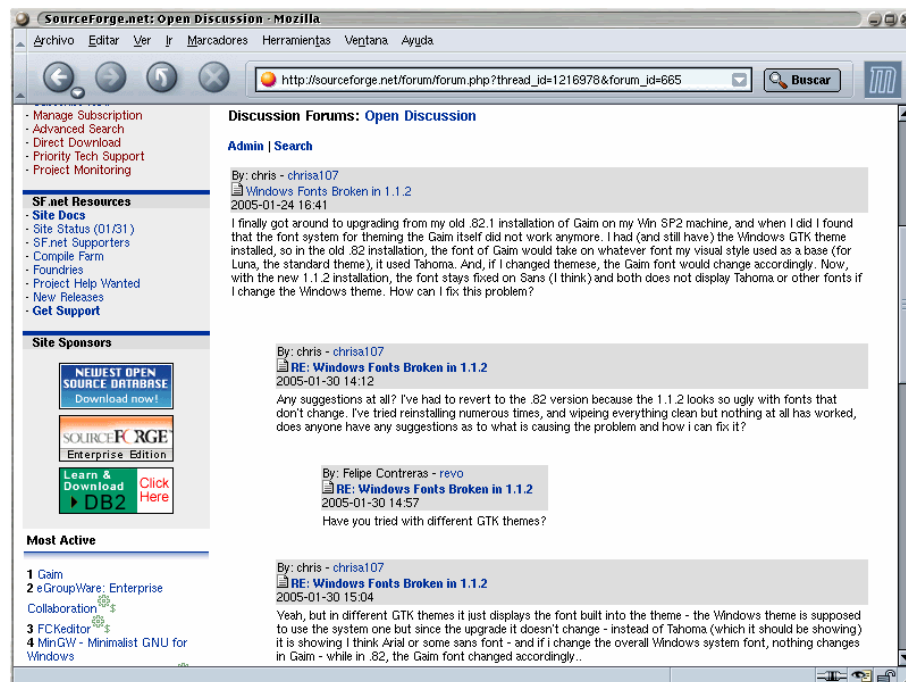


Figura 1.1: Página con enlaces a hilos

From msapiro at value.net Fri Jan 14 19:48:51 2005  
 From: msapiro at value.net (Mark Sapiro)  
 Date: Fri Jan 14 19:49:04 2005  
 Subject: [Mailman-Users] mailman issues  
 In-Reply-To: <003c01c4fa40\$1d99b4c0\$94592252@gaurav7klgnyif>  
 Message-ID: <PC173020050114104851057801b04d55@msapiro>

gaurav\_gold wrote:

>How can people reply to the mailing list? How do i turn off  
 this feature? How can i also enable a feature where if someone  
 replies the newsletter the email gets deleted?

See the FAQ

>Mailman FAQ: <http://www.python.org/cgi-bin/faqw-mm.py>  
 article 3.11

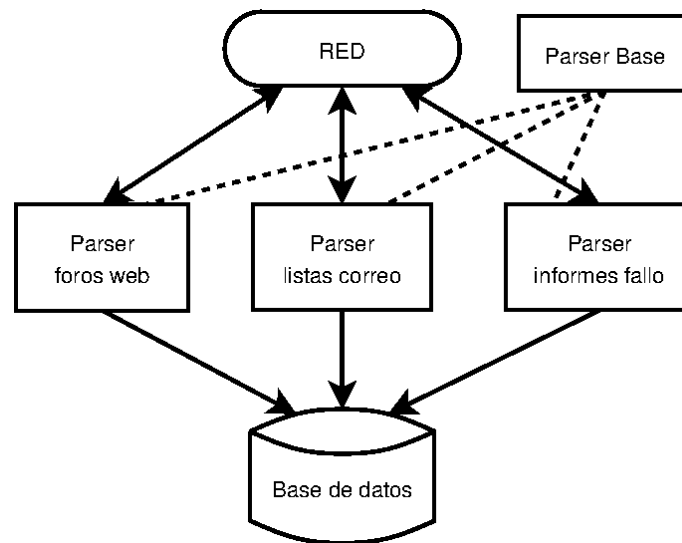


Figura 1.2: Estructura del parser básico

## 1.2. Estructura de la memoria

En esta sección se debería introducir la estructura de la memoria.

Así:

- En el primer capítulo se hace una intro al proyecto.
- En el capítulo 2 (ojo, otra referencia automática) se muestran los objetivos del proyecto.
- A continuación se presenta el estado del arte en el capítulo 3.
- ...

# Capítulo 2

## Objetivos

### 2.1. Objetivo general

Aquí vendría el objetivo general en una frase: Mi trabajo fin de grado consiste en crear de una herramienta de análisis de los comentarios jocosos en repositorios de software libre alojados en la plataforma GitHub.

Recuerda que los objetivos siempre vienen en infinitivo.

### 2.2. Objetivos específicos

Los objetivos específicos se pueden entender como las tareas en las que se ha desglosado el objetivo general. Y, sí, también vienen en infinitivo.

### 2.3. Planificación temporal

A mí me gusta que aquí pongáis una descripción de lo que os ha llevado realizar el trabajo. Hay gente que añade un diagrama de GANTT. Lo importante es que quede claro cuánto tiempo llevas (tiempo natural, p.ej., 6 meses) y a qué nivel de esfuerzo (p.ej., principalmente los fines de semana).



# Capítulo 3

## Estado del arte

En este capítulo se explicarán todas las tecnologías, herramientas y accesorios necesarios para el desarrollo y funcionamiento de este proyecto.

### 3.1. Lenguajes de Marcado y Programación

- HTML: Lenguaje estándar para crear páginas web, define la estructura y el contenido utilizando etiquetas.
- JavaScript: Lenguaje de programación utilizado en desarrollo web para agregar interactividad y dinamismo a las páginas.
- TeX: Sistema de composición tipográfica utilizado principalmente para la creación de documentos científicos y técnicos de alta calidad.

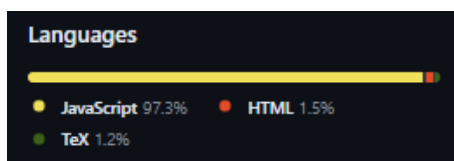


Figura 3.1: Porcentaje de lenguajes usados.

## 3.2. Importaciones de Bibliotecas JavaScript

- **THREE.js:** Biblioteca JavaScript de código abierto para crear y renderizar gráficos en 3D en el navegador web.
- **VRButton.js:** Módulo de Three.js que facilita la integración de botones para activar la funcionalidad de realidad virtual (VR) en aplicaciones web.
- **XRControllerModelFactory.js:** Módulo de Three.js que proporciona una fábrica para crear modelos de controladores de realidad extendida (XR) para su uso en aplicaciones de realidad virtual y aumentada.
- **Stats.js:** Módulo de Three.js que ofrece una utilidad para monitorear el rendimiento (FPS, uso de memoria) de aplicaciones web 3D en tiempo real.
- **OrbitControls.js:** Módulo de Three.js que proporciona controles de órbita para permitir al usuario rotar, acercar y alejar la cámara en una escena 3D de manera interactiva.

## 3.3. Herramientas de Desarrollo

- **Firefox:** Navegador web de código abierto conocido por su enfoque en la privacidad y la personalización.
- **VisualStudioCode:** Editor de código fuente de Microsoft altamente personalizable y de código abierto, conocido por su rendimiento y amplia gama de extensiones.
- **GitHub:** Plataforma de desarrollo colaborativo de software basada en la nube, que permite a los desarrolladores alojar, revisar, colaborar y desplegar proyectos de software, incluidas aplicaciones web, de manera eficiente y transparente.
- **Oculus Quest 2:** Gafas de realidad virtual autónomas producidas por Oculus (una subsidiaria de Facebook), conocidas por su alta calidad y facilidad de uso. Estas gafas fueron proporcionadas por la universidad para el desarrollo del proyecto.
- **Web XR API Emulator:** Herramienta de desarrollo que permite probar y depurar experiencias de realidad virtual y aumentada basadas en WebXR directamente en el navegador web.



# Capítulo 4

## Diseño e implementación

En este capítulo, se detalla el desarrollo del proyecto, incluyendo su estructura, los componentes en el entorno inmersivo, su funcionamiento, y las interacciones implementadas.

### 4.1. Arquitectura general

El proyecto se estructura en cinco componentes enlazados entre si, representados en la figura 4.2. Estos componentes serán explicados a continuación:

- `index.html`: Representamos la estructura básica de una página web para una experiencia de realidad virtual (VR). Se define un documento HTML con metadatos y referencias a archivos externos, incluyendo la biblioteca Three.js para gráficos 3D.

Además, se importa un módulo de JavaScript que inicializa la aplicación VR, creando una instancia de la clase `.Appz` asignándola a la ventana del navegador.

Este archivo proporciona una base para el desarrollo de aplicaciones de realidad virtual en la web.

- `app.js`: Este script es una aplicación que utiliza la biblioteca Three.js para crear una escena 3D interactiva. Que realiza:

1. Importa las bibliotecas necesarias de Three.js para crear la escena.
2. Carga varias texturas de imágenes para aplicarlas a los materiales de los objetos en la escena.

3. Define algunas variables y objetos necesarios para el funcionamiento de la aplicación.
  4. Crea una clase App que inicializa la escena, la cámara, la iluminación y los controles.
  5. Dentro de la clase App, hay métodos para manejar el control de los dispositivos de entrada de VR, como los controladores y sus eventos.
  6. Contiene un método para construir los modelos de los controladores de VR.
  7. Otro método se encarga de manejar la lógica de la animación y la interacción del usuario con la escena.
  8. Y finalmente, tenemos un método de renderizado que se ejecuta en bucle y actualiza la escena en cada fotograma.
- imágenes: En la arquitectura del proyecto, tenemos una carpeta llamada "imágenes" que contiene todas las imágenes utilizadas en la aplicación. Dentro de esta carpeta, tenemos subcarpetas para organizar las imágenes de acuerdo con su propósito.
- Cada imagen se carga en la aplicación utilizando el constructor `THREE.TextureLoader().load()`, proporcionando la ruta relativa desde el punto donde se está ejecutando el código hacia la ubicación de la imagen en la estructura de carpetas.
- sceneObjets: Este archivo es un módulo de funciones que proporciona diferentes objetos tridimensionales (3D) utilizando la biblioteca Three.js. Cada función devuelve un objeto con geometría y material específicos.

1. Sphere: Devuelve una esfera roja con una geometría esférica y un material básico.
2. Side: Devuelve un objeto rectangular con una textura proporcionada, usado como pared en la escena.
3. Box: Devuelve una caja con una textura proporcionada, usado como UA y Proxy en la escena.
4. Wall: Devuelve un plano grande con una textura proporcionada, usado como pantalla en la escena.
5. Logo: Devuelve un plano con una textura proporcionada, representando logotipo.

6. Floor: Devuelve un plano grande con una textura proporcionada, usado para el suelo de la escena.

Cada función utiliza diferentes geometrías y materiales para crear los objetos 3D. También configura propiedades específicas de los materiales, como envoltura de textura y repetición.

- **jsm**: Esta carpeta contiene módulos JavaScript utilizados en el proyecto. Estos módulos son archivos independientes que proporcionan funcionalidades específicas para la aplicación. Aquí está la estructura de la carpeta:

1. **build**: En esta subcarpeta se encuentran los archivos de construcción de la biblioteca Three.js, que proporcionan la funcionalidad principal de renderización en 3D. El archivo principal utilizado es `three.module.js`, que importa todas las funcionalidades esenciales de Three.js.
2. **webxr**: Aquí están los módulos relacionados con WebXR, una API que permite crear experiencias de realidad virtual (VR) y realidad aumentada (AR) en el navegador. Por ejemplo, `VRButton.js` proporciona un botón para activar la experiencia de realidad virtual, mientras que `XRControllerModelFactory.js` permite crear modelos visuales para los controladores de VR.
3. **libs**: Esta subcarpeta contiene bibliotecas adicionales utilizadas en el proyecto. En particular, `stats.module.js` proporciona una herramienta para medir el rendimiento de la aplicación mediante la visualización de estadísticas, como el número de fotogramas por segundo.
4. **controls**: Aquí están los módulos relacionados con el control de la cámara y la escena en la aplicación. Por ejemplo, `OrbitControls.js` proporciona controles para permitir al usuario mover y rotar la cámara alrededor de la escena, lo que facilita la navegación en entornos 3D.

Por lo tanto, la carpeta '**jsm**' organiza los módulos JavaScript utilizados en el proyecto, proporcionando funcionalidades esenciales, soporte para WebXR, herramientas de medición de rendimiento y controles de cámara. Estos módulos se importan en la aplicación según sea necesario para agregar las características deseadas.

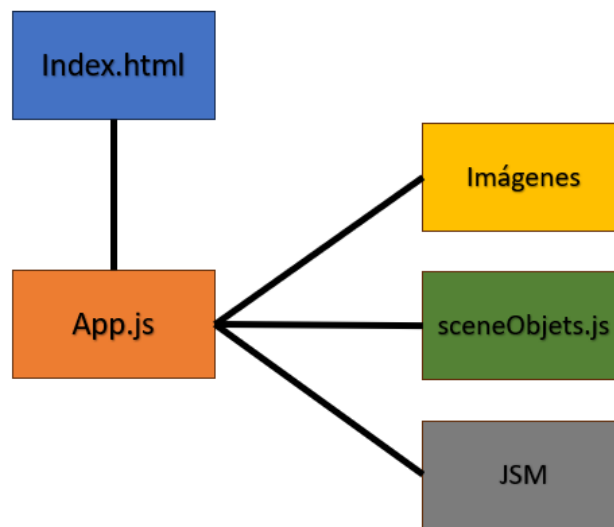


Figura 4.1: Estructura del proyecto

## 4.2. Escena

La escena virtual implementada se desarrolla utilizando la biblioteca Three.js, que permite la creación de gráficos 3D en un entorno web. Este proyecto configura un entorno interactivo que simula un sistema de comunicación, donde se pueden visualizar los intercambios de mensajes entre diferentes usuarios y un proxy.

### Estructura Básica

- **Cámara y Controles:** La cámara se configura con una perspectiva que permite al usuario tener una vista adecuada del entorno virtual. Se utiliza OrbitControls para permitir al usuario interactuar con la vista, rotando y acercando/alejando la escena.
- **Iluminación:** Se añade iluminación hemisférica para simular una luz suave y direccional para destacar objetos específicos dentro de la escena.
- **Renderizado:** Se establece un renderizado con antialiasing para mejorar la calidad visual de los bordes de los objetos en la escena.

### Objetos en la Escena

- **Suelo y Paredes:** Se utilizan texturas cargadas para crear un suelo y paredes que encierran la escena, proporcionando un fondo estático.

- **Objetos Interactivos:** Se crean varias cajas y esferas que representan a diferentes usuarios (UA1, UA2) y un proxy. Estos objetos se pueden interactuar mediante eventos controlados por los controladores de realidad virtual.
- **Indicadores Visuales:** Se utilizan logos y texturas para indicar estados como "startz" "stop", mostrando visualmente el flujo de la simulación.

### Interacción y Dinámica

- **Inicio de Eventos:** La interacción comienza cuando el usuario activa controles específicos en los controladores de realidad virtual. Esto puede alterar el estado de los objetos (por ejemplo, cambiar el color para indicar actividad) y cambiar texturas que representan diferentes mensajes enviados y recibidos en la comunicación.
- **Simulación de Mensajes:** Los objetos esféricos se mueven entre las cajas para simular el envío de mensajes. La ruta y la dirección de estos objetos dependen de las interacciones del usuario y el flujo del protocolo simulado.
- **Animación de Paquetes RTP:** En ciertos puntos, esferas adicionales se mueven para simular la transmisión de paquetes RTP, indicando el intercambio de media en una llamada establecida.

### Implementación Técnica

- **Controladores XR:** Se configuran controladores para manejar la interacción en un entorno de realidad extendida (XR). Esto incluye la gestión de eventos de selección y conexión de dispositivos.
- **Animación y Actualización de la Escena:** Los objetos esféricos se mueven entre las cajas para simular el envío de mensajes. La ruta y la dirección de estos objetos dependen de las interacciones del usuario y el flujo del protocolo simulado.
- **Animación de Paquetes RTP:** La lógica para mover objetos y actualizar estados se ejecuta dentro de un bucle de animación que recalcula posiciones y estados basados en la interacción del usuario.

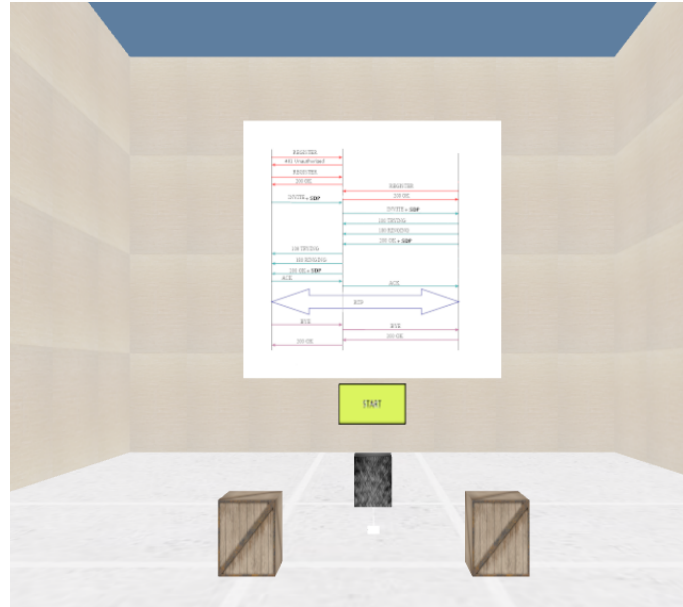


Figura 4.2: Escena del proyecto

### 4.3. Controles de Realidad Virtual

La implementación de controles de realidad virtual en la aplicación proporciona una interfaz interactiva que mejora significativamente la experiencia del usuario, permitiendo una manipulación intuitiva y directa de la escena virtual.

#### 4.3.1. Configuración de los Controladores VR

Los controladores VR son dispositivos físicos que los usuarios sostienen en sus manos y que detectan sus movimientos y gestos en el espacio tridimensional. Estos están equipados con una variedad de sensores y botones que permiten una gama amplia de interacciones:

- **Botones y Gatillos:** Utilizados para realizar selecciones y activar eventos dentro de la aplicación. Por ejemplo, un usuario puede iniciar la transmisión de mensajes al presionar el botón "Start".
- **Sensores de Movimiento:** Capturan el posicionamiento y la orientación de las manos del usuario, permitiendo manipular objetos o navegar por la escena con movimientos naturales.

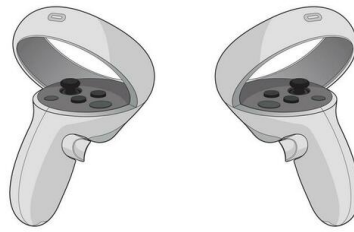


Figura 4.3: Controlador VR

### 4.3.2. Interacción Mediante Controladores VR

La interacción con la escena se realiza a través de los controladores VR de la siguiente manera:

- **Selección y Manipulación:** Los usuarios apuntan a objetos virtuales con los controladores y utilizan botones para seleccionarlos. Esto puede incluir activar elementos dentro de la escena.
- **Navegación:** Mediante el uso del gatillo en los controladores, los usuarios pueden desplazarse por la escena virtual, acercándose o alejándose de los objetos o desplazándose lateralmente.
- **Interacción Contextual:** La aplicación cambia la funcionalidad de la escena basándose en el estado del objeto con el que el usuario está interactuando, cambiando modos de visualización o ajustando parámetros específicos de los objetos.

### 4.3.3. Implementación Técnica

La implementación técnica de los controladores VR en la aplicación utiliza la API de WebXR, integrada con Three.js, para gestionar la entrada de los dispositivos de realidad virtual. El código configura cada controlador para responder a eventos como ‘selectstart’ y ‘selectend’, lo que permite detectar interacciones como pulsaciones de botones y liberaciones. Adicionalmente, se emplean rayos virtuales (‘raycasting’) para determinar qué objetos están siendo apuntados por los controladores, facilitando así una interacción precisa.

En la implementación de la aplicación, hacemos un uso específico de los botones trigger de los controladores VR para facilitar una interacción intuitiva y efectiva dentro del entorno

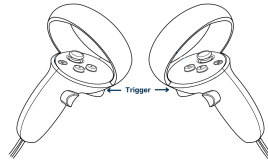


Figura 4.4: Botones controlador VR

virtual. Cada trigger tiene un propósito bien definido que mejora la experiencia del usuario y la funcionalidad de la aplicación:

- **Trigger del Controlador Izquierdo:** Este botón se utiliza para la navegación dentro de la escena. Al presionar el trigger del controlador izquierdo, el usuario puede desplazarse hacia donde esté mirando con las gafas de realidad virtual, lo que permite una navegación intuitiva y natural. Esta funcionalidad es fundamental para explorar diferentes áreas de la escena de manera fluida.
- **Trigger del Controlador Derecho:** El trigger del controlador derecho está configurado para interactuar con objetos específicos de la escena. Cuando se presiona este botón, se puede seleccionar o activar elementos dentro de la escena, como iniciar una simulación, modificar parámetros de un objeto, o ejecutar acciones específicas relacionadas con los objetos con los que se interactúa. Esta interacción es esencial para manipular elementos de la aplicación y para participar activamente en las simulaciones y demostraciones que se presentan en el entorno virtual.

La programación de estos triggers se realiza a través del manejo de eventos en JavaScript, utilizando la API de WebXR para detectar y responder a las acciones del usuario. Este diseño permite un control preciso sobre la aplicación, ofreciendo a los usuarios una manera clara y directa de interactuar con la interfaz y con los elementos virtuales de la escena.



## **Capítulo 5**

# **Experimentos y validación**

Este capítulo se introdujo como requisito en 2019. Describe los experimentos y casos de test que tuviste que implementar para validar tus resultados. Incluye también los resultados de validación que permiten afirmar que tus resultados son correctos.



# Capítulo 6

## Resultados

En este capítulo, se presentan los resultados derivados de las interacciones con los objetos de la escena en la aplicación de realidad virtual desarrollada. Se describirá cómo distintas selecciones de objetos generan resultados visuales y funcionales diferentes, lo que demuestra la dinámica y la reactividad de la aplicación ante las acciones del usuario.

### 6.1. Descripción de las Interacciones

Las interacciones en la aplicación permiten a los usuarios seleccionar objetos virtuales, los cuales alteran el estado de la escena y desencadenan eventos específicos.

Como se mencionó en la Sección 4.3.3, para lograr estos resultados es necesario utilizar controladores VR. Estos dispositivos permiten a los usuarios moverse por la escena e interactuar con los objetos de manera intuitiva y efectiva. La selección se realiza apuntando hacia el objeto con el controlador derecho y activando el trigger.

Cada objeto tiene asociadas consecuencias específicas que se detallan a continuación:

#### 6.1.1. Interacción con UA1

Este objeto representa un usuario dentro del entorno virtual y está visualizado como una caja de madera ubicada en la parte izquierda de la escena.

Al interactuar con UA1, este cambiará de color y podrán obtener dos posibles resultados, los cuales dependen de si el intercambio de paquetes ha sido inicializado o no.

### Intercambio de paquetes no iniciado

Si el intercambio de paquetes no ha sido inicializado, en la pantalla situada en la pared central se mostrarán los atributos del usuario en estado vacío, indicando los valores que deberían ser completados, mostrados en la figura 6.1. Esta visualización es esencial para comprender los requisitos iniciales de configuración del usuario en la red. Estos tributos son:

- **'account'**: ['username', 'passwd']
- **'uaserver'**: ['ip', 'puerto']
- **'rtpaudio'**: ['puerto']
- **'regproxy'**: ['ip', 'puerto']
- **'log'**: ['path']
- **'audio'**: ['path']

### Intercambio de paquetes iniciado

Una vez que el intercambio de paquetes ha sido iniciado, en la misma pantalla se actualizarán los atributos del usuario con valores específicos y correctos para su funcionamiento mostrados en la figura 6.6. Este cambio refleja cómo el sistema procesa y responde a las interacciones, ofreciendo un feedback visual del estado operativo del usuario.

#### 6.1.2. Interacción con el Proxy

Este objeto representa un servidor proxy en el entorno virtual y está visualizado como una caja con textura oscura, ubicada en la parte central de la escena.

Al interactuar con el Proxy, este cambiará de color y dependiendo del estado de la interacción con los usuarios (UA1 y UA2), se pueden obtener diferentes resultados, influenciados por si el intercambio de paquetes ha sido iniciado o no.



Figura 6.1: Estado inicial del UA1



Figura 6.2: Atributos UA1 con valores específicos



Figura 6.3: Estado inicial del Proxy

### Intercambio de paquetes no iniciado

Si el intercambio de paquetes no ha sido inicializado, la interacción con el Proxy mostrará en la pantalla sus atributos con la información que deben ser completados, donde no se procesan aún los paquetes, mostrados en la figura 6.3. Estos campos son:

- **'server':** ['name', 'ip', 'puerto']
- **'database':** ['path', 'passwdpath']
- **'log':** ['path']

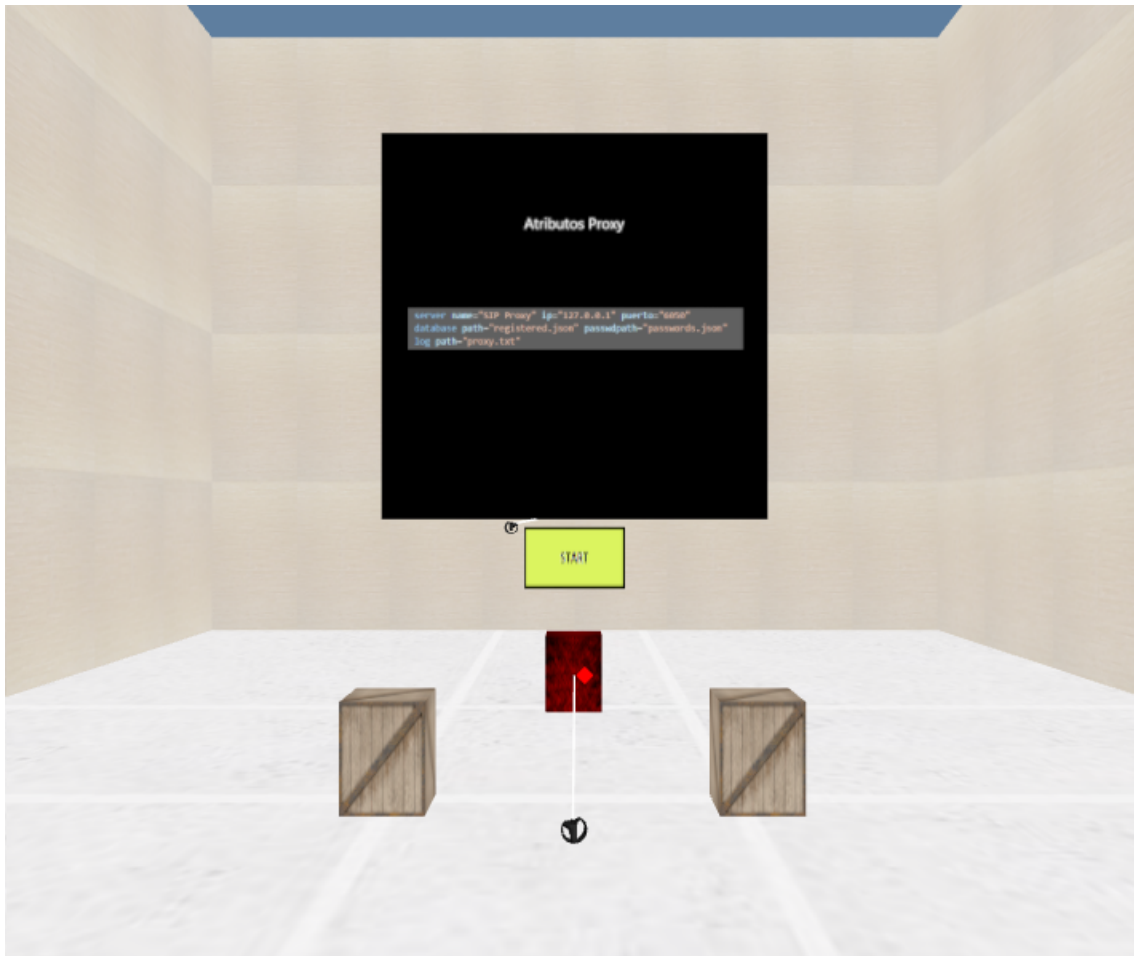


Figura 6.4: Atributos Proxy con valores específicos

### Intercambio de paquetes iniciado

Una vez que el intercambio de paquetes ha sido iniciado, el Proxy procesará y dirigirá los paquetes entre los usuarios correspondientes, reflejando este flujo en la interfaz con actualizaciones dinámicas. Al seleccionar el Proxy se mostrarán sus atributos con valores específicos y correctos para su funcionamiento, mostrados en la figura 6.4.

### 6.1.3. Interacción con UA2

Este objeto representa un usuario dentro del entorno virtual y está visualizado como una caja de madera ubicada en la parte derecha de la escena.

Al interactuar con UA2, este cambiará de color y podrán obtener dos posibles resultados, los cuales dependen de si el intercambio de paquetes ha sido inicializado o no.



### Intercambio de paquetes no iniciado

Si el intercambio de paquetes no ha sido inicializado, en la pantalla situada en la pared central se mostrarán los atributos del usuario en estado vacío, indicando los valores que deberían ser completados, mostrados en la figura 6.5. Esta visualización es esencial para comprender los requisitos iniciales de configuración del usuario en la red. Estos atributos son:

- **'account'**: ['username', 'passwd']
- **'uaserver'**: ['ip', 'puerto']
- **'rtpaudio'**: ['puerto']
- **'regproxy'**: ['ip', 'puerto']
- **'log'**: ['path']
- **'audio'**: ['path']

### Intercambio de paquetes iniciado

Una vez que el intercambio de paquetes ha sido iniciado, en la misma pantalla se actualizarán los atributos del usuario con valores específicos y correctos para su funcionamiento mostrados en la figura ???. Este cambio refleja cómo el sistema procesa y responde a las interacciones, ofreciendo un feedback visual del estado operativo del usuario.

#### 6.1.4. Interacción con paquetes

En el entorno de realidad virtual, los paquetes se representan en forma de esferas que simulan la transmisión de datos en una red. Estas esferas visualizan claramente el flujo de información, con un origen, un destino y un color claramente definidos, dependiendo del tipo de intercambio en el que participan.

Al seleccionar uno de estos paquetes, el sistema mostrará información detallada contenida en el paquete, incluyendo una representación visual del flujo de transmisión que indica su origen, destino y la dirección del flujo.

El proceso y secuencia de intercambio de paquetes en este proyecto se ilustra en la siguiente figura 6.7, donde se puede observar el orden específico seguido durante las simulaciones.



Figura 6.5: Estado inicial del UA2



Figura 6.6: Atributos UA2 con valores específicos

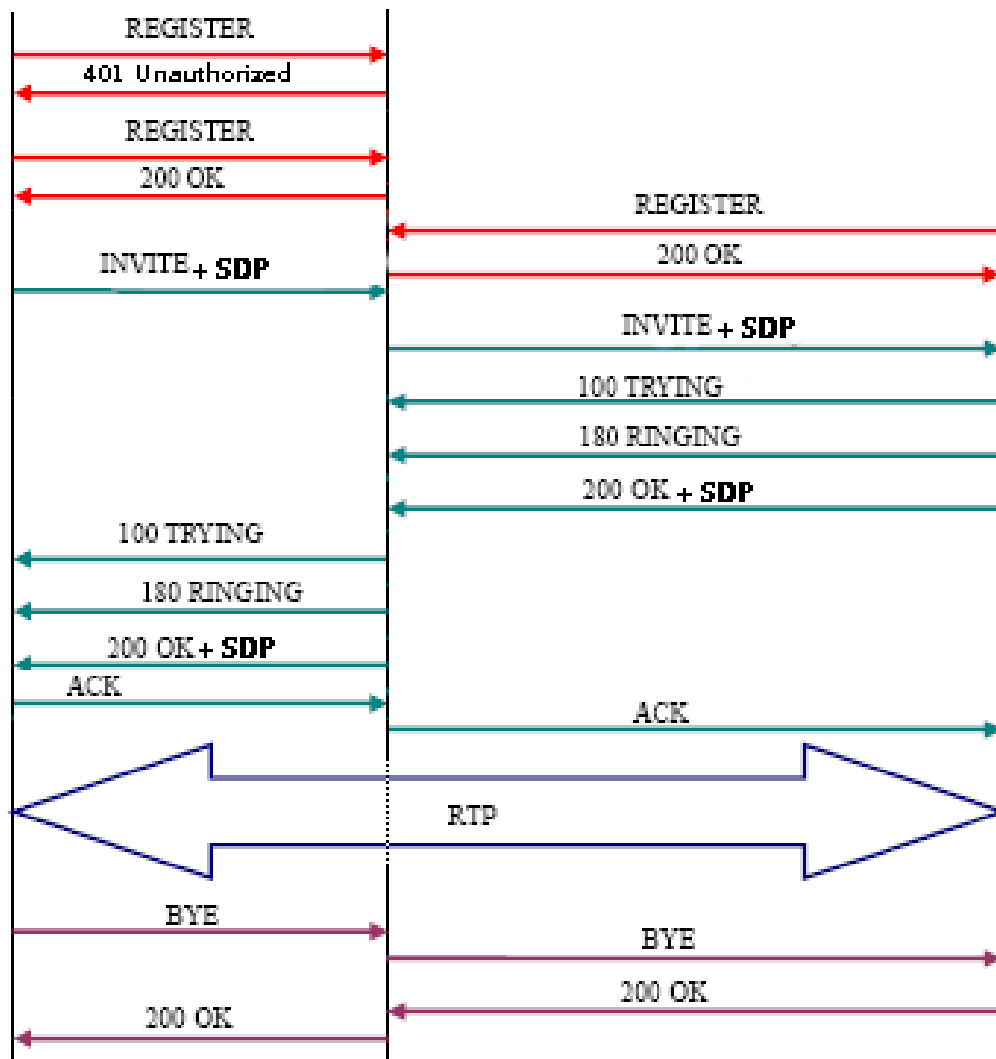


Figura 6.7: Secuencia del intercambio de paquetes

# Capítulo 7

## Conclusiones

### 7.1. Consecución de objetivos

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir qué se ha conseguido y qué no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemas se han encontrado y qué medidas se han tomado para mitigar esos problemas.

Y si has llegado hasta aquí, siempre es bueno pasarle el corrector ortográfico, que las erratas quedan fatal en la memoria final. Para eso, en Linux tenemos `aspell`, que se ejecuta de la siguiente manera desde la línea de *shell*:

```
aspell --lang=es_ES -c memoria.tex
```

### 7.2. Aplicación de lo aprendido

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM. Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a

2. b

### **7.3. Lecciones aprendidas**

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. Aquí viene uno.
2. Aquí viene otro.

### **7.4. Trabajos futuros**

Ningún proyecto ni software se termina, así que aquí vienen ideas y funcionalidades que estaría bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFMs.

# **Apéndice A**

## **Manual de usuario**

Esto es un apéndice. Si has creado una aplicación, siempre viene bien tener un manual de usuario. Pues ponlo aquí.

