



ESCUELA DE INGENIERÍA DE FUENLABRADA

GRADO EN INGENIERÍA TELEMÁTICA

TRABAJO FIN DE GRADO

MAPAS DE TERRENO PARA REALIDAD VIRTUAL

Autor : José Fuhui Pérez Contreras

Tutor : Dr. David Moreno Lumbresas

Curso académico 2024/2025



©2025 José Fuhui Pérez Contreras

Algunos derechos reservados

Este documento se distribuye bajo la licencia

“Atribución-CompartirIgual 4.0 Internacional” de Creative Commons,
disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>

*Dedicado a
mis padres*

Agradecimientos

Quiero agradecer a todas las personas que me han ayudado a lo largo de mi carrera, y en especial a mis amigos de clase y a mi tutor, David, por su apoyo y ayuda en la realización de este trabajo. También quiero agradecer a mis padres por su apoyo incondicional y por haberme dado la oportunidad de estudiar en esta universidad.

Resumen

Este proyecto tiene como objetivo la creación de un componente en tres dimensiones compatible con la realidad virtual. El componente utiliza *APIS* de *Mapbox* y *OpenStreetMap* para la creación de mapas de terreno en tres dimensiones. Se pretende que el componente pueda ser utilizado en aplicaciones de realidad virtual y que sea fácil de integrar en otros proyectos.

El componente se ha desarrollado utilizando *JavaScript* y *HTML*, apoyándose en los entornos de trabajo de *Node.js*, *A-Frame* (y, por tanto, *WebGL* y *Three.js*) y *BabiaXR*. La intención es que esté disponible en la web para que cualquier persona pueda utilizarlo y contribuir a su desarrollo.

Los mapas son una herramienta muy útil para la visualización de datos y la navegación en entornos tridimensionales. Durante toda la historia se han valorado los mapas como una herramienta fundamental para la exploración y el conocimiento del mundo. En la actualidad, con el auge de la realidad virtual y la realidad aumentada, los mapas han cobrado una nueva dimensión y se están convirtiendo en una herramienta esencial para la creación de experiencias inmersivas. Es por ello que pienso que este proyecto es relevante y puede tener un impacto significativo en el campo de la realidad virtual y la visualización de datos.

Summary

This project aims to create a three-dimensional component compatible with virtual reality. The component uses APIs from *Mapbox* and *OpenStreetMap* to generate three-dimensional terrain maps. The goal is for the component to be usable in virtual reality applications and to be easy to integrate into other projects.

The component has been developed using *JavaScript* and *HTML*, relying on the *Node.js*, *A-Frame* (and therefore *WebGL* and *Three.js*) and *BabiaXR* frameworks. The intention is to make it available on the web so that anyone can use it and contribute to its development.

Maps are a very useful tool for data visualization and navigation in three-dimensional environments. Throughout history, maps have been valued as a fundamental tool for exploration and understanding of the world. Nowadays, with the rise of virtual reality and augmented reality, maps have taken on a new dimension and are becoming an essential tool for creating immersive experiences. That is why I believe this project is relevant and may have a significant impact in the field of virtual reality and data visualization.

Índice general

1. Introducción	1
1.1. Contexto	2
1.2. Estructura de la memoria	2
2. Objetivos	5
2.1. Objetivo general	5
2.2. Objetivos específicos	6
2.3. Planificación temporal	6
3. Estado del arte	9
3.1. Babia-XR	9
3.1.1. A-Frame	10
3.1.2. Three.js	11
3.1.3. WebXR	12
3.2. HTML	12
3.3. JavaScript y Node.js	13
3.3.1. Node.js y herramientas asociadas	14
3.4. APIs de Mapbox y <i>OpenStreetMap</i>	14
3.5. Gafas de Realidad Virtual: Meta Quest 3	15
3.6. Editor de Código: Visual Studio Code	17
3.7. Apoyo de herramientas de inteligencia artificial	17
4. Diseño e implementación	19
4.1. Arquitectura general	19

5. Experimentos y validación	21
6. Resultados	23
7. Conclusiones	25
7.1. Consecución de objetivos	25
7.2. Aplicación de lo aprendido	25
7.3. Lecciones aprendidas	26
7.4. Trabajos futuros	26
A. Manual de usuario	27
Bibliografía	29

Índice de figuras

2.1.	Diagrama de Gantt del TFG.	8
3.1.	Sitio web oficial de <i>A-Frame</i>	11
3.2.	Código HTML del componente.	13
3.3.	Código fuente en <i>JavaScript</i> del componente.	14
3.4.	Sitio web oficial de <i>OpenStreetMap</i>	15
3.5.	Gafas prestadas por la universidad.	16
4.1.	Estructura del parser básico	20

Capítulo 1

Introducción

En los últimos años, el desarrollo de tecnologías inmersivas como la realidad virtual (VR) y la realidad aumentada (AR) ha propiciado nuevas formas de interactuar con la información digital. En este contexto, los mapas tridimensionales se presentan como una herramienta clave para enriquecer experiencias inmersivas, facilitar la navegación en entornos complejos y mejorar la visualización de datos geoespaciales. Este proyecto se enmarca en dicha línea de innovación, proponiendo la creación de un componente web en tres dimensiones que permita representar mapas de terreno realistas y compatibles con dispositivos de realidad virtual.

El componente desarrollado hace uso de diversas tecnologías modernas del ecosistema web, como *A-Frame*, *WebGL* y *Three.js*, facilitando la creación de entornos 3D interactivos en navegadores estándar. A su vez, se apoya en servicios de terceros como *Mapbox* y *OpenStreetMap*, que proporcionan datos geográficos y texturas necesarias para representar el terreno con un alto nivel de detalle. Todo ello se integra mediante *JavaScript*, *HTML* y herramientas de desarrollo como *Node.js*, dentro del ecosistema de la plataforma *BabiaXR*, orientada a facilitar el desarrollo de experiencias inmersivas en la web.

Uno de los principales objetivos del proyecto es que el componente resultante sea modular, reutilizable y accesible para otros desarrolladores o investigadores que quieran incorporarlo en sus propias aplicaciones. Por este motivo, se ha priorizado su disponibilidad en la web y la posibilidad de que terceros puedan contribuir a su mejora.

Además del reto técnico, el proyecto plantea una reflexión sobre el papel actual de los mapas en la era digital. A lo largo de la historia, los mapas han sido instrumentos fundamentales para la exploración, la planificación y la representación del mundo. Hoy, con el auge de las

tecnologías inmersivas, su potencial se amplía hacia nuevas formas de interacción, análisis y visualización de datos. En este sentido, este proyecto busca aportar una herramienta útil y abierta que contribuya a la evolución del uso de los mapas en entornos virtuales.

1.1. Contexto

Los mapas han sido históricamente una herramienta fundamental para representar el territorio y facilitar la navegación, la planificación y la toma de decisiones. En la actualidad, su importancia ha ido en aumento gracias a los avances tecnológicos que permiten su digitalización, interacción y visualización tridimensional en tiempo real.

En el ámbito de la Ingeniería Telemática, los mapas no solo se utilizan como elementos visuales, sino también como interfaces que permiten acceder, consultar y analizar grandes volúmenes de datos georreferenciados. Esta relación directa con las bases de datos espaciales establece un nexo claro con asignaturas cursadas durante el grado, como *Laboratorio de Bases de Datos*, donde se han trabajado conceptos fundamentales como la estructuración eficiente de datos, su recuperación y la optimización de consultas.

El uso de mapas tridimensionales y tecnologías asociadas como *WebXR* abre un abanico de posibilidades en diferentes áreas. Desde la educación inmersiva y la visualización científica, hasta la planificación urbana, el turismo o la simulación de entornos complejos. Al permitir representar información en contextos espaciales, los mapas se convierten en una poderosa herramienta de comunicación y análisis.

Este proyecto surge del interés por combinar estas capacidades con tecnologías modernas de visualización en entornos web, apoyado además por el hecho de que *Babia XR*, la herramienta empleada, ha sido desarrollada por docentes e investigadores de la Universidad Rey Juan Carlos. Este hecho no solo facilita el acceso al entorno de trabajo, sino que también refuerza el valor académico del proyecto al fomentar la colaboración con desarrollos propios del entorno universitario.

1.2. Estructura de la memoria

Esta memoria se estructura de la siguiente manera:

- En este capítulo inicial se presenta el contexto general del proyecto, los antecedentes que lo motivan y los objetivos que persigue. Además, se describe la estructura del documento para orientar al lector.
- El capítulo 2 expone los objetivos generales y específicos del proyecto, junto con la planificación temporal diseñada para su desarrollo.
- A continuación, en el capítulo 3 se analiza el estado del arte, incluyendo una revisión de tecnologías afines, soluciones previas y herramientas relevantes para el desarrollo de experiencias interactivas en realidad virtual.
- El capítulo 4 aborda el diseño y la implementación del componente desarrollado, con especial atención a su arquitectura, funcionalidades principales y decisiones técnicas adoptadas durante el proceso.
- En el capítulo 5 se describen los experimentos realizados y la validación del prototipo, incluyendo los criterios utilizados para evaluar su funcionamiento.
- El capítulo 6 recoge los resultados obtenidos tras la fase de pruebas, valorando en qué medida se han alcanzado los objetivos planteados.
- Por último, el capítulo 7 recoge las conclusiones generales del trabajo, analiza el grado de cumplimiento de los objetivos, reflexiona sobre el aprendizaje adquirido y propone posibles líneas de trabajo futuro.
- Como material complementario, se incluye un apéndice con un manual de usuario, así como la bibliografía utilizada en la elaboración del proyecto.

Capítulo 2

Objetivos

2.1. Objetivo general

El objetivo principal de este proyecto es desarrollar un componente que permita la visualización e interacción con mapas tridimensionales en un entorno de realidad virtual accesible directamente desde un navegador web. La meta es que cualquier usuario pueda acceder a la experiencia sin necesidad de instalar software adicional, accediendo a un entorno inmersivo y navegable desde la propia aplicación web.

Para ello, se ha trabajado con tecnologías web avanzadas como *A-Frame* y *WebXR*, que permiten renderizar escenas tridimensionales y gestionar la interacción en entornos de realidad virtual utilizando únicamente *HTML* y *JavaScript*. Además, se hace uso de servicios de mapas y datos de elevación para representar de forma realista el terreno, mejorando la sensación de inmersión.

Uno de los aspectos clave del proyecto es la ****interacción en tiempo real****, es decir, que el usuario pueda moverse por el mapa, hacer zoom, cambiar la textura o incluso modificar parámetros del terreno sin abandonar la aplicación. Esta capacidad convierte al componente en una herramienta versátil que puede ser aplicada en múltiples contextos, desde la enseñanza y la divulgación científica, hasta el urbanismo, el turismo o la planificación territorial.

2.2. Objetivos específicos

Además del objetivo general, el desarrollo del componente plantea una serie de objetivos específicos que buscan dotar al sistema de funcionalidades concretas y mejorar su utilidad práctica. Estos objetivos se resumen a continuación:

- **Generar una malla 3D del terreno:** Obtener y representar el relieve de una región geográfica mediante una malla tridimensional, a partir de datos abiertos ofrecidos por APIs como *OpenStreetMap*, Mapbox o servicios similares que proporcionen información topográfica o de elevación.
- **Implementar múltiples capas de visualización:** Permitir al usuario seleccionar entre diferentes tipos de capas superpuestas sobre la malla del terreno, tales como vista satelital, vista de calle o una capa personalizada con colores definidos manualmente. Esta funcionalidad facilita distintos modos de análisis y exploración del mapa.
- **Navegación libre por el mapa:** Ofrecer la posibilidad de desplazarse libremente por el mapa tridimensional utilizando controles de navegación intuitivos, además de permitir acciones como hacer *zoom in* y *zoom out* para explorar distintas escalas de visualización del terreno.
- **Funcionalidades personalizables por el usuario:** Incorporar opciones de personalización que permitan al usuario ajustar parámetros como el tamaño del mapa, su posición dentro del entorno 3D, el nivel de detalle o el escalado vertical del relieve. Estas características permiten adaptar la experiencia a distintos casos de uso o preferencias individuales.

2.3. Planificación temporal

La realización de este Trabajo de Fin de Grado se ha desarrollado a lo largo de varios meses, compatibilizándose con las prácticas curriculares en empresa y la asistencia regular a clases presenciales. Esto ha condicionado la disponibilidad horaria, centrando el grueso del trabajo en fines de semana y períodos no lectivos. A continuación, se detalla la planificación temporal seguida:

- **Septiembre de 2024:** Primer contacto con el tutor del TFG, David Moreno Lumbreras, para explorar posibles temáticas del trabajo.
- **Octubre de 2024:** Reunión con el tutor para concretar la propuesta definitiva, centrada en el desarrollo de un componente para mapas en entornos de realidad virtual, en el marco de la plataforma Babia XR.
- **Octubre - diciembre de 2024:** Fase de documentación y aprendizaje. Se estudian tecnologías clave como *A-Frame* y Babia XR, y se experimenta con componentes ya existentes dentro de esta plataforma. Esta etapa se desarrolló mayoritariamente durante los fines de semana debido a la carga lectiva y laboral (prácticas curriculares y clases los martes y miércoles).
- **Diciembre de 2024:** La universidad facilita el acceso a unas gafas de realidad virtual, permitiendo comenzar el desarrollo y pruebas en entorno real.
- **Febrero de 2025:** Finalizado el periodo de exámenes, se retoma el trabajo de manera más intensiva, comenzando el desarrollo del componente personalizado. En este periodo de tiempo también implementé los conocimientos en *JavaScript* adquiridos en la asignatura de *Aplicaciones Telemáticas* cursada paralelamente
- **Marzo de 2025:** Se alcanza un hito importante: la generación de una malla del terreno y la implementación de una primera capa visual obtenida desde la API de *OpenStreetMap*.
- **Abril de 2025:** Desarrollo de un sistema de capas múltiples, incluyendo la creación de una interfaz gráfica de usuario para gestionar su visualización, así como la incorporación de un mecanismo para trabajar con el token de Mapbox.
- **Mayo de 2025:** Implementación de controles interactivos compatibles con realidad virtual (botones visibles en las gafas VR), así como la posibilidad de modificar coordenadas y niveles de zoom. Redacción y finalización progresiva de la memoria del TFG.
- **Finales de mayo e inicios de junio de 2025:** Corrección y revisión final de la memoria, preparación de la defensa.

Cabe destacar que, aunque las fases descritas se presentan de forma lineal, muchas tareas se han solapado parcialmente. La dedicación al proyecto ha sido principalmente en fines de semana y festivos, dada la simultaneidad con otras actividades académicas y profesionales. Esto también ha significado el no poder tener un plan de reuniones con el tutor, pasando a programar las reuniones (muchas de ellas telemáticamente) cuando podía crearse un hueco libre y el proyecto hubiera avanzado lo suficiente.

A continuación, se muestra un diagrama de Gantt del proceso completo:

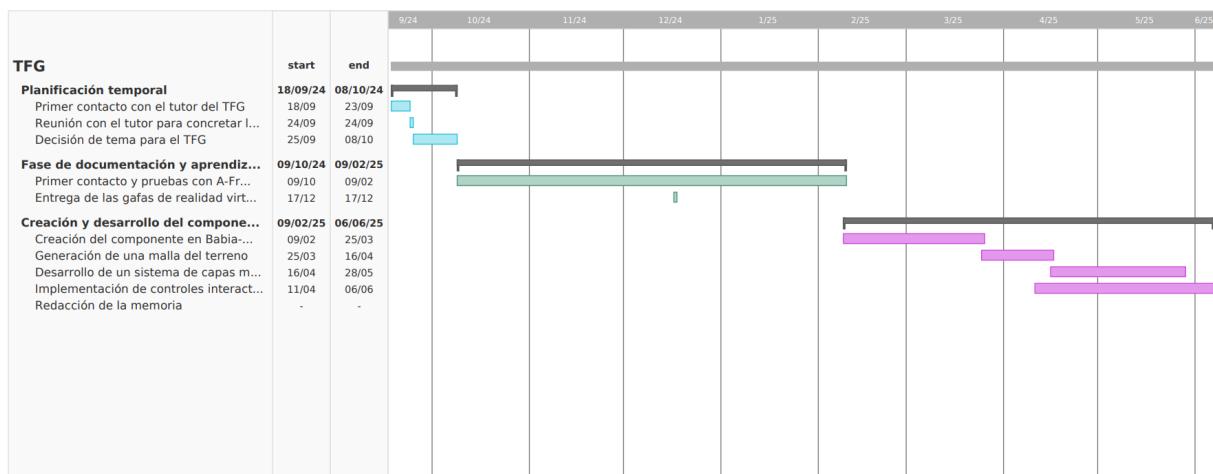


Figura 2.1: Diagrama de Gantt del TFG.

Capítulo 3

Estado del arte

3.1. Babia-XR

Babia XR es una plataforma de desarrollo para entornos inmersivos y experiencias interactivas en realidad virtual (VR), realidad aumentada (AR) y realidad mixta (MR). Se trata de una iniciativa impulsada por investigadores y docentes de la Universidad Rey Juan Carlos (URJC), lo que refuerza su vinculación con el ámbito académico y su aplicación en proyectos de investigación y desarrollo tecnológico.

El objetivo principal de Babia XR es facilitar la creación de aplicaciones inmersivas mediante tecnologías web abiertas, como *WebXR*, *A-Frame* y *Three.js*. Estas tecnologías permiten ejecutar contenidos tridimensionales directamente en navegadores compatibles, sin necesidad de instalar software adicional. Esto representa una ventaja significativa en términos de accesibilidad, portabilidad y compatibilidad con múltiples dispositivos, incluyendo gafas de realidad virtual como las Meta Quest.

En el contexto de este trabajo, Babia XR se utiliza como entorno base para la integración del componente de visualización de mapas tridimensionales. La elección de esta plataforma responde a dos motivos principales:

1. **Necesidad técnica:** proporciona la infraestructura necesaria para desplegar escenas VR complejas, integrando capas de información geoespacial, controles de navegación y representaciones 3D sobre terrenos reales.
2. **Vinculación institucional:** al tratarse de una herramienta desarrollada dentro de la propia

universidad, su uso refuerza la colaboración entre estudiantes y líneas de investigación del centro, además de facilitar el acceso a soporte técnico y documentación específica.

Babia XR se apoya en un conjunto de tecnologías modernas y bien consolidadas:

- *A-Frame*: framework basado en HTML para crear experiencias VR accesibles desde navegadores.
- *Three.js*: biblioteca JavaScript que permite renderizar gráficos 3D en WebGL con gran flexibilidad.
- *WebXR*: estándar para ejecutar experiencias inmersivas directamente en el navegador, compatible con múltiples dispositivos.

Gracias a esta combinación, Babia XR permite un desarrollo ágil y modular, facilitando la incorporación de componentes personalizados, como el sistema de teselas con desplazamiento que se ha implementado en este proyecto.

3.1.1. A-Frame

A-Frame es un framework de código abierto desarrollado inicialmente por Mozilla para facilitar la creación de experiencias de realidad virtual usando HTML. Su diseño declarativo permite definir escenas tridimensionales mediante etiquetas similares a las del propio lenguaje HTML, lo cual lo hace accesible incluso para desarrolladores con conocimientos limitados de gráficos 3D o WebGL.

A-Frame está construido sobre *Three.js*, lo que le permite mantener una gran potencia gráfica bajo una capa de abstracción más sencilla. Además, es extensible mediante componentes personalizados, lo que permite adaptar el comportamiento de los elementos de la escena a las necesidades específicas del proyecto.

En este trabajo, *A-Frame* actúa como base para representar los elementos visuales del mapa tridimensional, así como para integrar los controles de usuario, cámaras, luces y otros objetos interactivos.

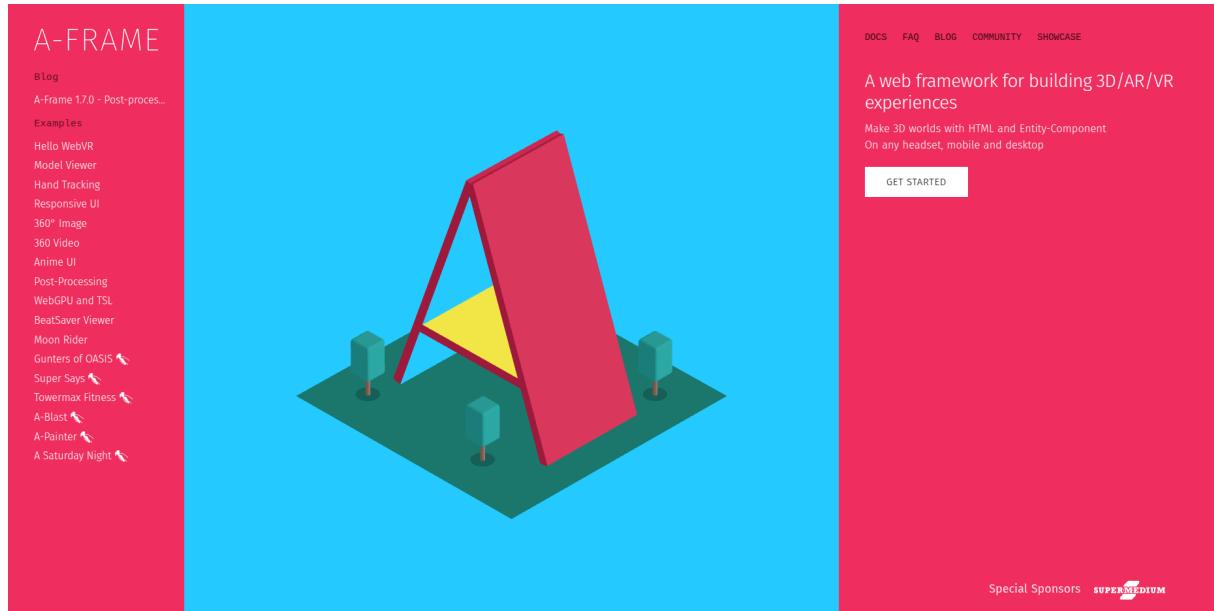


Figura 3.1: Sitio web oficial de *A-Frame*

3.1.2. Three.js

Three.js es una biblioteca JavaScript muy utilizada para renderizar gráficos tridimensionales en el navegador utilizando WebGL. Proporciona una interfaz de alto nivel que simplifica el uso de funcionalidades avanzadas de gráficos 3D, como luces dinámicas, materiales complejos, sombras en tiempo real y animaciones.

En el contexto de Babia XR, Three.js es el motor gráfico que subyace bajo *A-Frame*. Aunque normalmente no se accede directamente a su API cuando se utiliza *A-Frame*, su presencia es fundamental para permitir el renderizado eficiente de los objetos en la escena.

Algunos componentes personalizados del proyecto, especialmente los relacionados con el desplazamiento de texturas y la carga dinámica de teselas, se benefician de las capacidades avanzadas de Three.js, ya que permiten modificar directamente atributos de materiales y geometrías a bajo nivel.

WebGL

WebGL (Web Graphics Library) es una API de JavaScript para renderizar gráficos 2D y 3D dentro de cualquier navegador web compatible sin necesidad de plugins. Está basada en OpenGL ES 2.0 y permite acceder a la aceleración gráfica del hardware de forma segura. En el

contexto del proyecto, WebGL es la tecnología subyacente sobre la que se construyen bibliotecas como Three.js y A-Frame, lo que permite representar escenas tridimensionales complejas directamente en el navegador.

3.1.3. WebXR

WebXR es un estándar del W3C que proporciona una interfaz unificada para acceder a dispositivos de realidad virtual y aumentada desde navegadores web. Permite detectar dispositivos compatibles (como gafas VR o móviles con capacidades AR), obtener sus posiciones y orientaciones en tiempo real, y renderizar escenas de forma inmersiva.

A diferencia de sus predecesores (como WebVR), WebXR está diseñado para soportar tanto realidad virtual como aumentada desde una misma API, y su adopción por parte de los principales navegadores modernos garantiza su viabilidad a largo plazo.

Babia XR utiliza WebXR para habilitar la ejecución de escenas inmersivas en dispositivos como las Meta Quest, permitiendo al usuario entrar en la experiencia VR directamente desde el navegador. Esta capacidad resulta especialmente útil en entornos educativos y de investigación, donde la simplicidad de despliegue es una prioridad.

3.2. HTML

HTML (*HyperText Markup Language*) es el lenguaje estándar para estructurar contenido en la web. En este proyecto, HTML se emplea como base para organizar y enlazar los elementos visuales de la interfaz del componente, además de ser fundamental para la integración de *A-Frame*, que utiliza etiquetas personalizadas dentro de documentos HTML para definir entidades 3D y escenas interactivas.

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <title>A-Frame Babia Components</title>
7      <meta name="description" content="Babia Map example" />
8      <script src="https://aframe.io/releases/1.5.0/aframe.min.js"></script>
9      <script src="https://cdn.jsdelivr.net/gh/donmccurdy/aframe-extras@7.2.0/dist/aframe-extras.min.js"></script>
10     <script src="https://unpkg.com/aframe-environment-component@1.3.1/dist/aframe-environment-component.min.js"></script>
11     <script src="https://unpkg.com/aframe-text-geometry-component@0.5.1/dist/aframe-text-geometry-component.min.js"></script>
12     <script src="../../dist/aframe-babia-components.js"></script>
13     <link rel="stylesheet" href="../../assets/gitlab-corner.css">
14 </head>
15
16 <body>
17     <a-scene background="color: #A0F3FF" id="AframeScene">
18         <!-- <a-entity environment> -->
19         <a-light type="directional" color="#ffffff" intensity="1.5" position="5 10 5"></a-light>
20
21         <a-assets>
22             <a-asset-item id="optimerBoldFont" src="https://rawgit.com/mrdoob/three.js/dev/examples/fonts/optimer_bold.typeface.json"></a-asset-item>
23         </a-assets>
24
25         <a-entity id="map-terrain" class="collidable" babia-map="zoom: 5; x: 15; y: 12;" position="0 -150 0"></a-entity>
26
27         <a-entity id="right-desktop-ui-position" position="0.25 1.5 -0.5">
28             <!-- Botón para subir de capa -->
29             <a-entity id="zoomInBtn"
30                 geometry="primitive: plane; height: 0.075; width: 0.2"
31                 material="color: blue"
32                 text="value: Zoom In; align: center"
33                 position="0 0.05 -0.3"
34                 rotation="-30 0 0"
35                 class="collidable interactive-button"></a-entity>
36
37             <!-- Botón para bajar de capa -->
38             <a-entity id="zoomOutBtn"
39                 geometry="primitive: plane; height: 0.075; width: 0.2"
40                 material="color: purple"
41                 text="value: Zoom Out; align: center"
42                 position="0 -0.05 -0.25"
43                 rotation="-30 0 0"
44                 class="collidable interactive-button"></a-entity>
45         </a-entity>
46
47         <a-entity id="left-desktop-ui-position" position="-0.25 1.5 -0.5">
48             <!-- Botón para subir de capa -->
49             <a-entity id="zoomOutBtn"
50                 geometry="primitive: plane; height: 0.075; width: 0.2"
51                 material="color: purple"
52                 text="value: Zoom Out; align: center"
53                 position="0 -0.05 -0.25"
54                 rotation="-30 0 0"
55                 class="collidable interactive-button"></a-entity>
56     </a-scene>
57 </body>
58
59 <script>
60     document.addEventListener("DOMContentLoaded", () => {
61         const sceneEl = document.querySelector("#AframeScene");
62
63         sceneEl.addEventListener("loaded", () => {
64             const mapTerrainEl = sceneEl.querySelector("#map-terrain");
65             const zoomInEl = sceneEl.querySelector("#zoomInBtn");
66             const zoomOutEl = sceneEl.querySelector("#zoomOutBtn");
67             const leftEl = sceneEl.querySelector("#left-desktop-ui-position");
68             const rightEl = sceneEl.querySelector("#right-desktop-ui-position");
69
70             mapTerrainEl.addEventListener("click", () => {
71                 mapTerrainEl.babiaMap.zoom += 0.5;
72                 mapTerrainEl.babiaMap.x += 10;
73                 mapTerrainEl.babiaMap.y += 10;
74             });
75
76             zoomInEl.addEventListener("click", () => {
77                 mapTerrainEl.babiaMap.zoom += 0.5;
78             });
79
80             zoomOutEl.addEventListener("click", () => {
81                 mapTerrainEl.babiaMap.zoom -= 0.5;
82             });
83
84             leftEl.addEventListener("click", () => {
85                 mapTerrainEl.babiaMap.x -= 10;
86             });
87
88             rightEl.addEventListener("click", () => {
89                 mapTerrainEl.babiaMap.x += 10;
90             });
91
92             mapTerrainEl.addEventListener("dragstart", (e) => {
93                 e.preventDefault();
94             });
95
96             mapTerrainEl.addEventListener("drag", (e) => {
97                 mapTerrainEl.babiaMap.x += e.clientX - mapTerrainEl.clientX;
98                 mapTerrainEl.babiaMap.y += e.clientY - mapTerrainEl.clientY;
99             });
100            mapTerrainEl.addEventListener("dragend", (e) => {
101                e.preventDefault();
102            });
103        });
104    });
105 </script>

```

Figura 3.2: Código HTML del componente.

3.3. JavaScript y Node.js

JavaScript es el lenguaje de programación principal del entorno web, y se utiliza tanto en el desarrollo de interfaces interactivas como en la lógica del componente desarrollado. Permite manipular elementos HTML, responder a eventos del usuario y controlar el comportamiento del componente en tiempo real.

```

aframe-babia-components > components > visualizers > babia-map.js >  init
182   buildMesh: function (data, el) {
183     const fragmentShader = `
184     `;
185
186     // CREAR TEXTURA DESDE IMAGEN
187     async function loadDisplacementMap(url) {
188       let img = new Image();
189       img.src = url;
190       img.crossOrigin = "Anonymous"; // Evitar problemas CORS
191
192       return new Promise((resolve, reject) => {
193         img.onload = () => {
194           const canvas = document.createElement("canvas");
195           canvas.width = img.width;
196           canvas.height = img.height;
197           const ctx = canvas.getContext("2d");
198
199           // Dibujar la imagen en el canvas de forma invertida (la orientación de la imagen es diferente en canvas respecto a WebGL)
200           ctx.translate(0, canvas.height);
201           ctx.scale(1, -1);
202           ctx.drawImage(img, 0, 0);
203           const imageData = ctx.getImageData(0, 0, img.width, img.height);
204           const data = imageData.data;
205
206           // Crear un array para la textura de desplazamiento
207           const displacement = new Float32Array(img.width * img.height);
208
209           let elevation;
210           let auxMinHeight = 1;
211           let auxMaxHeight = 0;
212
213           for (let i = 0; i < data.length; i += 4) {
214             const R = data[i];
215             const G = data[i + 1];
216             const B = data[i + 2];
217
218             // Fórmula de Mapbox terrain-RGB -> Elevación en metros
219             elevation = (R * 256 * 256 + G * 256 + B) * 0.1 - 10000;
220
221             // Normalizar para Three.js (opcional, si necesitas valores entre 0 y 1)
222             elevation = (elevation + 10000) / 20000; // Elevación entre 0 y 1
223
224             // Comprobar el valor más bajo y más alto
225             if (elevation < auxMinHeight) {
226               auxMinHeight = elevation;
227             }
228             if (elevation > auxMaxHeight) {
229               auxMaxHeight = elevation;
230             }
231           }
232         }
233       });
234     }
235
236     const displacementMap = await loadDisplacementMap(data.url);
237
238     const material = new THREE.MeshStandardMaterial({
239       map: displacementMap,
240       displacementMap: displacement,
241       displacementScale: 100
242     });
243
244     el.innerHTML = '';
245     el.appendChild(material);
246   }
247 
```

Figura 3.3: Código fuente en *JavaScript* del componente.

3.3.1. Node.js y herramientas asociadas

Node.js es un entorno de ejecución para JavaScript fuera del navegador. Aunque el componente se ejecuta en el cliente, Node.js se utiliza durante el desarrollo para gestionar dependencias, ejecutar scripts y automatizar tareas. Para ello, se hace uso de npm (Node Package Manager), el gestor de paquetes de Node.js, y de nvm (Node Version Manager), que facilita la instalación y gestión de múltiples versiones de Node.js en un mismo sistema. Estas herramientas son necesarias para compilar, organizar y desplegar el componente de manera eficiente.

3.4. APIs de Mapbox y *OpenStreetMap*

Mapbox y *OpenStreetMap* son servicios que proporcionan datos cartográficos. *OpenStreetMap* (OSM) es una iniciativa colaborativa que ofrece mapas del mundo creados por una comunidad de voluntarios. Por su parte, Mapbox utiliza datos de OSM y otras fuentes para ofrecer un

conjunto de APIs avanzadas orientadas al desarrollo web, como mapas personalizados, capas de estilo y servicios de geolocalización.

En este proyecto se hace uso de estas APIs para obtener datos de elevación y estilos visuales (como capas satelitales o vistas de calle) que se aplican sobre la malla del terreno generada en el componente. Mapbox requiere un *token* de acceso personal, mientras que OSM puede utilizarse directamente mediante peticiones a sus servidores abiertos.

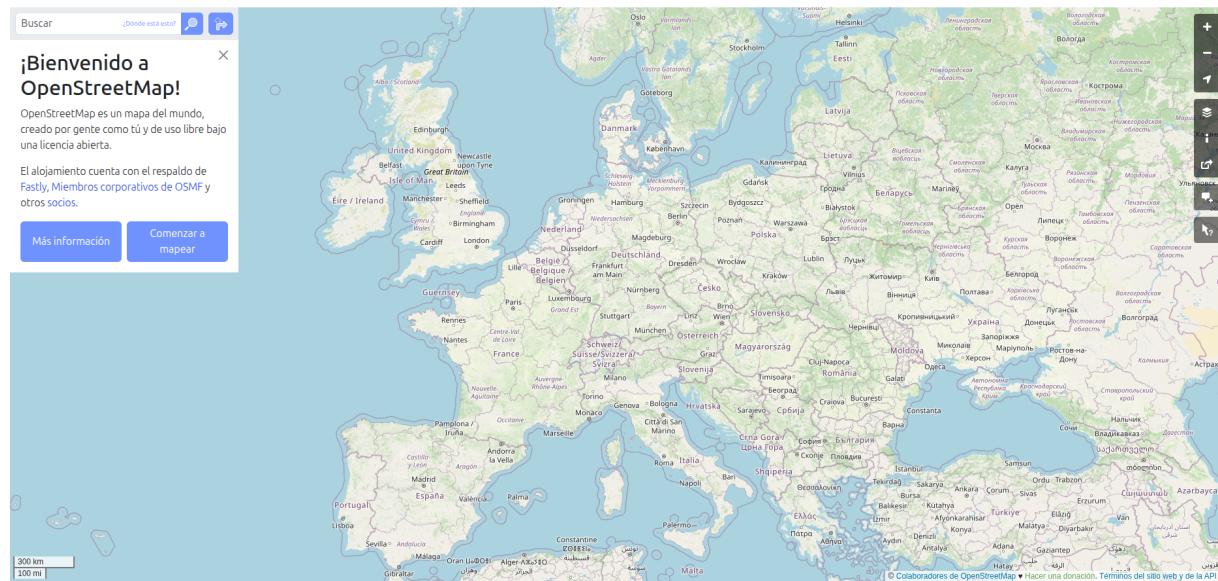


Figura 3.4: Sitio web oficial de *OpenStreetMap*.

3.5. Gafas de Realidad Virtual: Meta Quest 3

Las Meta Quest 3, desarrolladas por Meta (anteriormente Oculus), son unas gafas de realidad virtual autónomas que permiten ejecutar aplicaciones VR sin necesidad de un ordenador externo. Disponen de controladores hapticos, sensores de movimiento y capacidad de realidad mixta (passthrough a color), lo que las convierte en una herramienta ideal para pruebas e implementación de experiencias inmersivas.

Gracias al soporte de WebXR en su navegador interno, estas gafas permiten ejecutar el componente desarrollado directamente desde la web, lo que facilita el proceso de validación en un entorno real.



Figura 3.5: Gafas prestadas por la universidad.

3.6. Editor de Código: Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft, ampliamente adoptado por la comunidad de desarrolladores. Ofrece integración con sistemas de control de versiones, depuración, resaltado de sintaxis y soporte para extensiones que facilitan el desarrollo web y VR.

Durante el desarrollo del proyecto, se ha utilizado como entorno principal de trabajo, aprovechando su integración con herramientas como Git, su terminal integrada para la ejecución de comandos Node.js y su compatibilidad con tecnologías web modernas.

3.7. Apoyo de herramientas de inteligencia artificial

Durante el desarrollo del proyecto, se ha contado con el apoyo de herramientas basadas en inteligencia artificial como ChatGPT, que han sido útiles para resolver dudas técnicas, refinar la redacción de la documentación, generar fragmentos de código y comprender tecnologías específicas de forma más ágil.

Estas herramientas, si bien no sustituyen el conocimiento técnico, han servido como soporte complementario para acelerar el proceso de desarrollo, ahorrar tiempo en tareas repetitivas y mejorar la calidad de la documentación generada.

Capítulo 4

Diseño e implementación

Aquí viene todo lo que has hecho tú (tecnológicamente). Puedes entrar hasta el detalle. Es la parte más importante de la memoria, porque describe lo que has hecho tú. Eso sí, normalmente aconsejo no poner código, sino diagramas.

4.1. Arquitectura general

Si tu proyecto es un software, siempre es bueno poner la arquitectura (que es cómo se estructura tu programa a “vista de pájaro”).

Por ejemplo, puedes verlo en la figura 4.1. L^AT_EX pone las figuras donde mejor cuadran. Y eso quiere decir que quizás no lo haga donde lo hemos puesto... Eso no es malo. A veces queda un poco raro, pero es la filosofía de L^AT_EX: tú al contenido, que yo me encargo de la maquetación.

Recuerda que toda figura que añadas a tu memoria debe ser explicada. Sí, aunque te parezca evidente lo que se ve en la figura 4.1, la figura en sí solamente es un apoyo a tu texto. Así que explica lo que se ve en la figura, haciendo referencia a la misma tal y como ves aquí. Por ejemplo: En la figura 4.1 se puede ver que la estructura del *parser* básico, que consta de seis componentes diferentes: los datos se obtienen de la red, y según el tipo de dato, se pasará a un *parser* específico y bla, bla, bla...

Si utilizas una base de datos, no te olvides de incluir también un diagrama de entidad-relación.

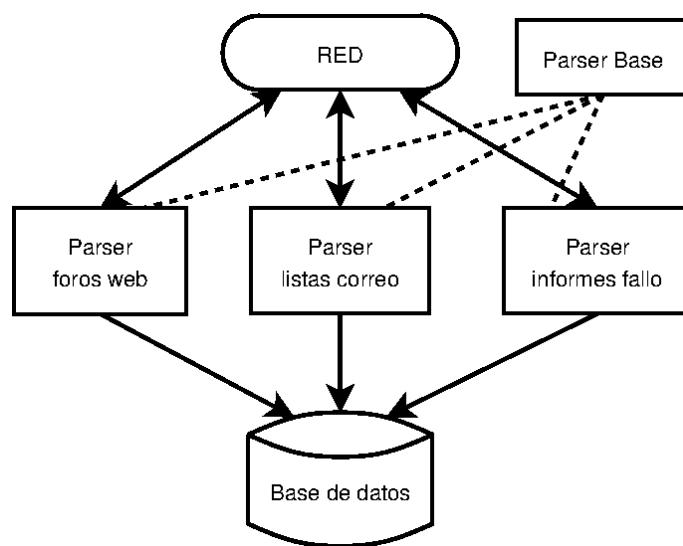


Figura 4.1: Estructura del parser básico

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 incluyen los resultados de tu trabajo fin de grado.

Si es una herramienta de análisis lo que has realizado, aquí puedes poner ejemplos de haberla utilizado para que se vea su utilidad.

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.

Bibliografía