



three.js journey



MODELARIUM 3D

Made by **Pablo Ortiz** and a **Pablo Cid**

Descripción

Para nuestro proyecto, utilizaremos una biblioteca llamada **three.js** que aprovecha las capacidades de **JavaScript** y **WebGL**. Esta notable combinación de tecnologías nos permitirá crear una experiencia web inmersiva en forma de una cautivadora galería, en la cual se exhibirán una amplia variedad de sorprendentes **modelos 3D** creados por los talentosos usuarios de nuestra plataforma.

Description

For our Project, we will be utilizing a powerful library called **three.js**, which harnesses the capabilities of **JavaScript** and **WebGL**. This remarkable combination of technologies will enable us to create an immersive web experience in the form of a captivating gallery, showcasing a myriad of stunning **3D models** contributed by the talented users of our platform.

Índice

Características Generales	4
Restricciones Generales	4
Aspectos Generales	5
Estudio	8
Justificación de la tecnología empleada	9
Requerimientos de Hardware y Software	10
Servidor	10
Cliente	11
Análisis y diseño	12
Casos de Uso	13
Base de datos	14
Diagrama de clases	15
Implementacion	15
Relación de ficheros	17
Evaluación y prueba	18
Criterios de Accesibilidad	18
Criterios de Usabilidad	18
Dispositivos/vistas	19
Software utilizado	26
Posibles Mejoras	27

Características Generales

1. **Subida de modelos 3D:** Implementar una funcionalidad que permita a los usuarios cargar y subir sus modelos 3D en formato .glb a la plataforma.
2. **Funcionalidad de compra y descarga:** Integrar un sistema que permita a los usuarios comprar y descargar los modelos 3D que deseen.
3. **Catálogo de modelos:** Crear un catálogo o galería donde los usuarios puedan explorar y buscar modelos 3D disponibles en el sitio.
4. **Categorización y etiquetado:** Implementar opciones de categorización y etiquetado para facilitar la búsqueda y navegación de los modelos 3D dentro del sitio.
5. **Perfiles de usuarios:** Crear perfiles de usuario donde los usuarios puedan administrar sus modelos subidos, compras realizadas y otras actividades relacionadas.




Restricciones Generales

1. **Almacenamiento y ancho de banda:** Considerar las restricciones de almacenamiento y ancho de banda necesarios para manejar la carga de modelos 3D y las descargas de los usuarios.
2. **Compatibilidad de navegadores:** Asegurarse de que la plataforma sea compatible con los navegadores web más utilizados, como Google Chrome, Mozilla Firefox y Safari.
3. **Seguridad de datos:** Implementar medidas de seguridad para proteger los datos de los usuarios, como la información de pago y los modelos subidos.

4. **Escalabilidad:** Si bien no se prevé un crecimiento importante del sitio, es importante tener en cuenta la posibilidad de escalabilidad para adaptarse a un aumento de usuarios en el futuro.
5. **Experiencia de usuario:** Diseñar una interfaz intuitiva y amigable para los usuarios, teniendo en cuenta la facilidad de navegación y la respuesta rápida de la plataforma.

Aspectos Generales

Vamos a hacer un resumen de los aspectos generales de nuestra página web a partir de nuestra idea inicial del proyecto.

Marcaremos en  verde los aspectos que hemos cubierto, en  rojo los aspectos no tratados y en  amarillo se resaltarán las modificaciones de los aspectos no tratados.

Estructura de la web

-Página principal:

Se mostrarán con animaciones de objetos 3D la información sobre el proyecto, los tipos de usuarios y la explicación de cómo interactuar con la página.

 (información innecesaria para un usuario) Cambio por una explicación breve en la página principal


-Galería de modelos 3D:

Aquí se mostrarán todos los modelos 3D de la web, se podrán ordenar por complejidad, fecha de subida...

También podrás ver los detalles del modelo que quieras.


 Eliminación de ordenar los modelos

-Contacto:

Los usuarios podrán contactar con nosotros a través de esta página. 

- Perfil: Donde se mostrará todo lo relevante al usuario logueado

- **Me gustas:** Se mostrarán los modelos que le gustan al usuario

- **Favoritos:** Donde se mostrarán los modelos que el usuario ha guardado como favoritos 

- **Datos:** Se mostrarán los datos del usuario logueado, y algunos se podrán modificar ✓
- **Cartera:** Se mostrará el saldo de la cuenta, los métodos de pago...
✓ ⚠ Implementado pero en una vista nueva (settings)
- **Modelos (creador):** Se mostrarán todos los modelos de un creador
 - **Todos:** Se muestran todos los modelos ✓
 - **Públicos:** Se muestran los modelos públicos ✓
 - **Privados:** Se muestran los modelos privados ✓

Sistema de comercio

Se podrán comprar todos los modelos 3D, es decir, el código y su precio variará dependiendo de la dificultad en la que se encuentren.

✓ ⚠ Al haber eliminado las dificultades el precio lo pone el creador y es aprobado por los administradores

Los Usuarios Creadores tendrán un descuento (10%) en todos los modelos e irá aumentando (20% - 35% - 50%) conforme se suban cada vez más modelos.

➡ El dinero ahora va directamente al creador

Tipos de usuarios

-Usuario no logueado:

- Solo podrán ojear los modelos de los creadores. ✓
- Ojear: podrá ver la visualización del modelo, su descripción y el perfil del usuario que lo ha publicado. ✓

- Usuario normal:

- Este usuario tendrá que someterse a un registro simple ✓
- Solo podrán ojear y dar "me gusta" a los modelos de creadores ✓
- Ojear: podrá ver la visualización del modelo, el código, su descripción, el perfil del usuario que lo ha publicado y podrá guardarlo como favorito.
✓ ⚠ No podrá ver ni descargar el código por cuestiones obvias.
- Editar cierta información de su perfil ✓

⚠ Los usuarios normales ahora también pueden comentar

- Creadores:

- Tienen todos los privilegios de ser un usuario normal, además de poder subir y borrar modelos 3D Three.js propios ✓
- Podrán comentar sobre los modelos de otros creadores ✓
- Podrán declarar los modelos propios como públicos u ocultos ✓
- Cuando un creador suba un modelo tendrá que rellenar distintos campos, como título, descripción... ✓
- Los creadores se someterán a un formulario de registro más ambicioso en el que se pedirán más requisitos que a un usuario corriente que entre otros campos tendrán que mandar un modelo 3D y que sea aprobado por los administradores ✓

- **Administradores:**

- crear y borrar usuarios de cualquier tipo ✓
- borrar comentarios, que estén fuera de lugar ✓
- Confirmaran peticiones de creaciones de modelos ✓
- Tendrán una bandeja de entrada de reportes de comentarios ✓

Cada usuario, ya sea creador o usuario corriente tendrá un perfil:

- **Características generales de un perfil de usuario normal:**

- Tipo de usuario ✓
- Foto de perfil (si tiene) ✓
- Descripción de usuario (si tiene) ✓
- Proyectos que le gustan ✓
- Datos de usuario ✓
- Sección de favoritos ✓
- Cartera ⚠ Solo los creadores

- **Características de un perfil creador:**

- Todas las de un perfil de usuario normal ✓
- Sección de creaciones ✓

Estudio

Primero vamos a echar un vistazo a páginas similares, aquí hay una lista de las más conocidas en el mercado:

1. CGTrader (<https://www.cgtrader.com/>)
2. TurboSquid (<https://www.turbosquid.com/>)
3. Sketchfab (<https://sketchfab.com/>)
4. BlendSwap (<https://www.blendswap.com/>)
5. 3DExport (<https://www.3dexport.com/>)
6. Hum3D (<https://hum3d.com/>)
7. CG Studio (<https://www.cgstudio.com/>)
8. Unity Asset Store (<https://assetstore.unity.com/>)

Otras Páginas

En estos tres ejemplos podemos observar cómo todas estas páginas tienen una biblioteca propia de modelos 3D al igual que nosotros y los cuales los usuarios o la propia página ha ido subiendo.

Los **permisos** necesarios para convertirse en creador y poder subir modelos en estas páginas son **sumamente restrictivos** ya que lo que buscan es **generar ventas**, de las cuales el usuario no se lleva prácticamente nada.

Modelarium3D

Nuestro objetivo a diferencia de estas páginas es que **cualquier persona** con un poco de conocimiento pueda subir sus modelos o tenerlos guardados en su perfil para propósitos personales.

Además las ganancias son **totalmente** para el creador de ese modelo, por lo tanto un buen diseñador puede ganar el dinero que se merece. Así creemos que podemos generar una **comunidad sana** y sostenible entre creadores.

Justificación de la tecnología empleada

En nuestro proyecto hemos empleado mucho tiempo al estudio de dos nuevas tecnologías que desconocíamos. Vamos a hacer un pequeño resumen de estas herramientas.



[Blender](#) ha sido una herramienta imprescindible para **Modelarium3D**. Este software de código abierto nos ha permitido realizar modelos, animaciones y renderizados en 3D de manera eficiente y precisa.

Gracias a Blender, hemos podido crear y editar diversos modelos 3D, los cuales mostramos en la propia página web.

Además, el hecho de que Blender sea de código abierto y gratuito nos ha beneficiado mucho, ya que nos ha permitido ahorrar en gastos y acceder a una comunidad activa de usuarios y desarrolladores que comparten conocimientos y mejoras constantes.

En conclusión, Blender ha desempeñado un papel fundamental en nuestro **Modelarium3D**, permitiéndonos realizar modelos y renderizados de alta calidad, y llevar nuestra plataforma al siguiente nivel.

Y para mostrar los modelos en nuestra web necesitábamos de una herramienta nueva y también desconocida, aquí es cuando entra en escena [Three.js](https://threejs.org/)



[Three.js](https://threejs.org/) ha sido una biblioteca también muy esencial para **Modelarium3D**. Esta biblioteca, basada en JavaScript y WebGL, nos permitió crear una experiencia web inmersiva con una galería de modelos 3D que creamos en Blender.

Gracias a la documentación, ejemplos y recursos compartidos por la comunidad, hemos podido acelerar el proceso de desarrollo y nos ha permitido resolver desafíos de manera efectiva.

Requerimientos de Hardware y Software

Servidor

Requerimientos de hardware:

1. Procesador de alto rendimiento: Para manejar las cargas de trabajo de procesamiento de los modelos 3D, es recomendable que sea un procesador potente y con múltiples núcleos.

2. Memoria RAM adecuada: Es recomendable tener una cantidad suficiente de memoria RAM para gestionar la carga de modelos 3D y las operaciones de renderizado.

3. Almacenamiento suficiente: Dado que los modelos 3D en formato .glb pueden ocupar espacio considerable, es necesario contar con un almacenamiento adecuado para alojar y gestionar los archivos de los usuarios.

4. Gran Ancho de banda: Para permitir una transferencia de datos rápida y eficiente al subir y descargar los modelos 3D, es importante contar con una conexión de alta velocidad y un ancho de banda suficiente.

Requerimientos de software:

1. Sistema operativo: Tanto windows como linux soportan las tecnologías que vamos a usar, aún así es más recomendable un servidor linux por su versatilidad de configuración y gestión.

2. Servidor web: Apache, para manejar las solicitudes de los usuarios y subir los modelos 3D almacenados en el servidor.

3. Base de datos: Necesitaríamos también nuestra base de datos en el servidor para almacenar información sobre los modelos 3D, los usuarios y otros datos relevantes. Lo óptimo sería usar MySQL, ya que hemos estado trabajando con una base de datos en local de este tipo.

4. Bibliotecas y frameworks: Utilizamos three.js para mostrar los modelos 3D en el navegador.

Cliente

Requerimientos de hardware:

1. Computadora o dispositivo compatible: Los usuarios necesitarán un equipo o dispositivo compatible con capacidades gráficas suficientes para visualizar los modelos 3D en su navegador web.

2. Tarjeta gráfica adecuada: Se recomienda una tarjeta gráfica dedicada o integrada con soporte para WebGL y aceleración 3D para una mejor experiencia de visualización de modelos 3D.

3. Conexión a Internet: Los usuarios necesitarán una conexión a Internet estable y de velocidad adecuada para cargar y descargar modelos 3D de manera eficiente.

Requerimientos de software:

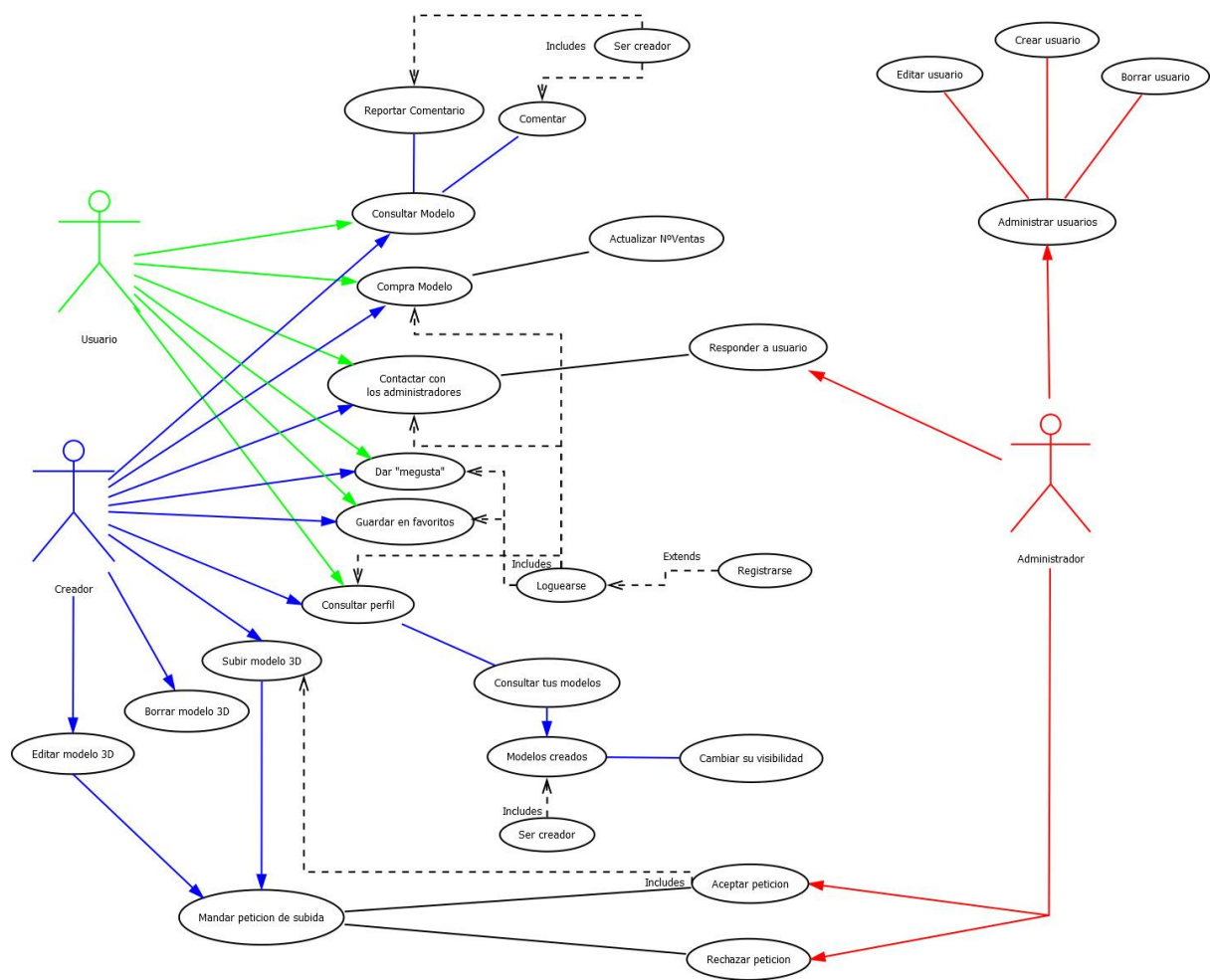
1. Navegador web actualizado: Los usuarios deben tener instalado un navegador web moderno y actualizado que admita las últimas tecnologías web, como WebGL y JavaScript. Se recomiendan navegadores como Google Chrome, Mozilla Firefox, Safari o Microsoft Edge.

2. Habilitar WebGL: El usuario debe asegurarse de que WebGL esté habilitado en la configuración del navegador web. Esto permitirá la visualización de los modelos 3D en formato .glb.

3. Soporte para formatos de archivo .glb: Es importante que los navegadores de los usuarios sean compatibles con el formato de archivo .glb para que puedan cargar y visualizar los modelos 3D correctamente.

Análisis y diseño

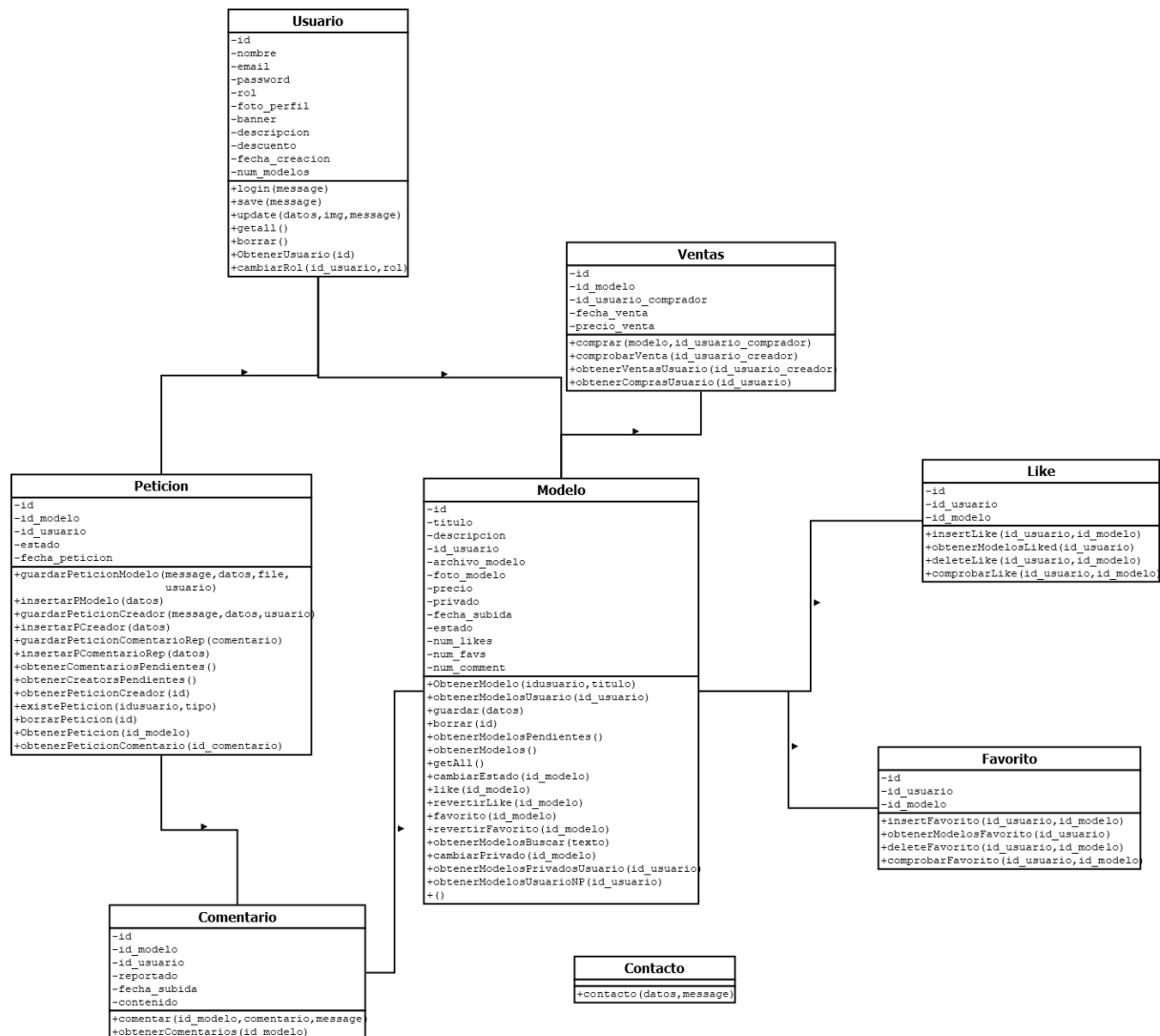
Casos de Uso



BASE DE DATOS MODELARIUM 3D: ENTIDAD-RELACION



Diagrama de clases



Implementacion

descripción de los elementos que se usan: hojas de estilo, plantillas, formularios, funciones de envío de datos, conexión, consultas y selección de la BD. Uso de

fichero de configuración.

Antes de nada hemos de decir que **TODO** el código ha sido hecho por nosotros **desde 0**, esto incluye absolutamente todos los archivos, base de datos, htaccess...

Frontend:

Archivos views: Estos son los archivos que muestran las vistas, en su mayoría están hechos con html y le incluimos la funcionalidad mediante PHP.

Archivos CSS: Estos son los archivos que han dado forma a la página, sus estilos. Son archivos muy extensos y elaborados ya que cada etiqueta html tiene su propia clase, distinguida de todas las demás.

Archivos JS: Estos son los archivos que han dado una experiencia para el usuario más intuitiva y más agradable a la vista, lo que incluye distintas funcionalidades.

Backend:

Base de datos: La base de datos está creada a mano y hemos ido haciendo modificaciones sobre esta según nos ha ido conveniendo hasta su estado final.

Router: Hemos utilizado una clase Router para la redirección de rutas desde el Index.php de Public.

Pages: Carga la vista que le pasamos.

Utils: Hemos incorporado un Utils porque utilizamos muchas funciones muchas veces y ha sido más fácil para nosotros recopilarlas todas aquí.

Email: La clase email se encarga de enviar emails.

Security: Se encarga de encriptar y desencriptar las contraseñas, entre otras cosas.

Controllers: Aquí encontramos el grueso de nuestra página, estos controladores se encargan de mandar y redirigir a cada enlace, vista, funcionalidad que hemos creado.

Models: Estos son los modelos que son llamados por los controladores. Cada modelo se encarga de una parte de la página, usuarios, likes, favoritos... Todos estos modelos tienen funciones que llaman a la base de datos para **Crear**, **Borrar** o **Modificar** sus tablas según su función.

Relación de ficheros

Vamos a hacer una descripción un poco más genérica ya que nuestro proyecto incluye 90 archivos y sería demasiado extenso explicar todos y cada uno de ellos, por eso vamos a hacer grupos.

Ficheros Controllers:
Acceso a sus modelos, por ejemplo UsuarioController → Models\Usuario. Algunos de los controladores usan el InterController, el cual es un controlador que sirve para conectar entre otros controladores, esto lo hemos hecho para evitar la recursividad a la hora de las llamadas entre controladores. También utilizan Utils para las funciones más frecuentes A estos controladores se les llama desde las vistas u otros controladores. Suelen usar \$GET, \$POST, \$SESSION, \$FILE.

Ficheros Models:
Cada uno de ellos es llamado por su propio controlador o por InterController. Contiene las funciones que conectan con la base de datos, elimina, crea, o cambia datos. Suelen trabajar con objetos o los crean a partir de sus propias funciones.

Ficheros Views:
Son todas las vistas que se muestran en la página web Estas vistas llaman a los controladores y obtienen la información de estos.

Evaluación y prueba

Nuestra herramienta principal de testeo hemos sido nosotros mismos, aún así también hemos contado con la aportación de nuestros amigos y familiares que se han prestado a ser nuestros usuarios Beta.

Todos los formularios de la página están hechos a base de pruebas, no solo pasan el control de la Base de datos o el de el propio html si no que también hemos implementado nuestras propias validaciones con php y todas funcionan correctamente.

Manual de estilos

Criterios de Accesibilidad

1. **Contraste y legibilidad:** Los colores utilizados en la página web tienen un contraste adecuado para facilitar la legibilidad de las personas con discapacidades visuales.
2. **Etiquetado semántico:** Utilizamos etiquetas HTML adecuadas para estructurar el contenido de manera semántica, lo que facilita la navegación y comprensión del sitio para personas que utilizan tecnologías de asistencia.
3. **Texto alternativo y descripciones:** Proporcionamos texto alternativo descriptivo para todas las imágenes y elementos visuales, permitiendo a las personas con discapacidades visuales acceder a la información a través de lectores de pantalla u otras tecnologías de asistencia.

Criterios de Usabilidad

1. **Diseño intuitivo y sencillo:** Mantenemos una interfaz de usuario limpia y fácil de entender, con una estructura y disposición de elementos coherentes. También facilitamos la navegación y el acceso a las funcionalidades principales de manera intuitiva.
2. **Carga rápida de páginas:** Hemos optimizado al máximo posible el rendimiento del sitio web para garantizar tiempos de carga rápidos, lo que mejora la experiencia del usuario y reduce la frustración.
3. **Información clara y concisa:** Hemos proporcionado instrucciones claras y concisas sobre cómo utilizar las funcionalidades de la página web, como la compra, visualización y descarga de modelos 3D.

4. **Pruebas de usabilidad:** Hemos realizado pruebas de usabilidad con usuarios reales(familiares y amigos) para identificar posibles áreas de mejora en la navegación, comprensión y usabilidad general de la página web.

Dispositivos/vistas

Las vistas para las que se ha diseñado el proyecto son todas las disponibles para dispositivos desde 320px hasta 4000px, los puntos de ruptura son: 320px, 500px, 650px, 800px, 1100px, 2000px, 1500px, 2000px, 4000px.

Color Palette

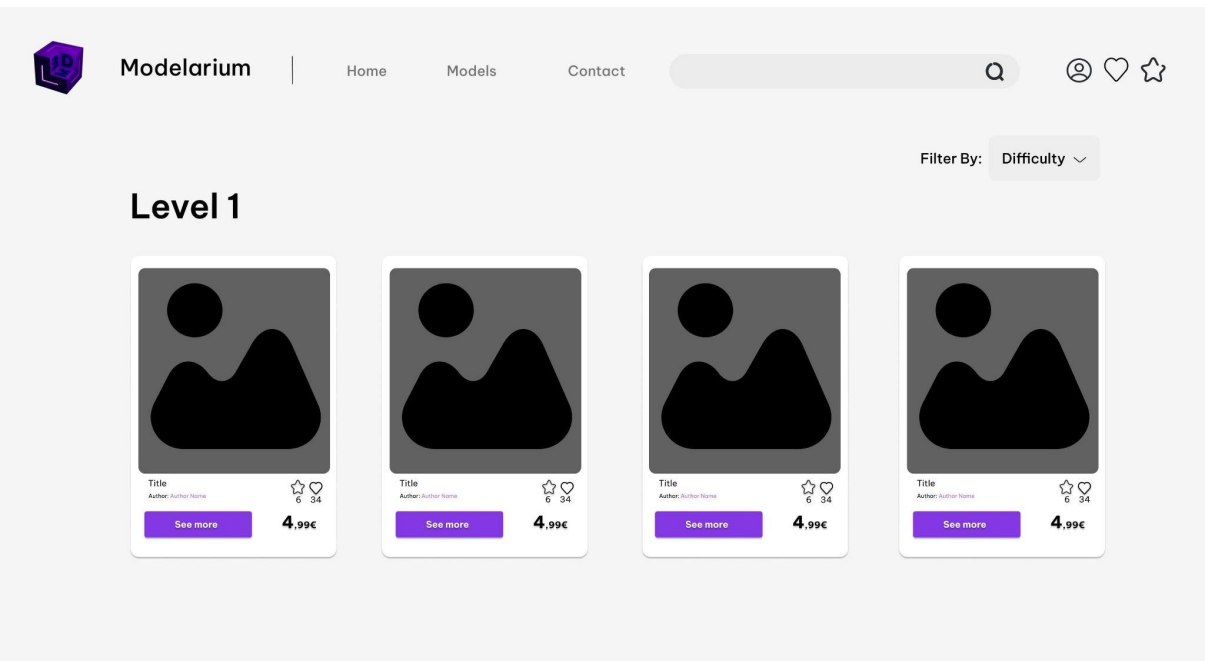
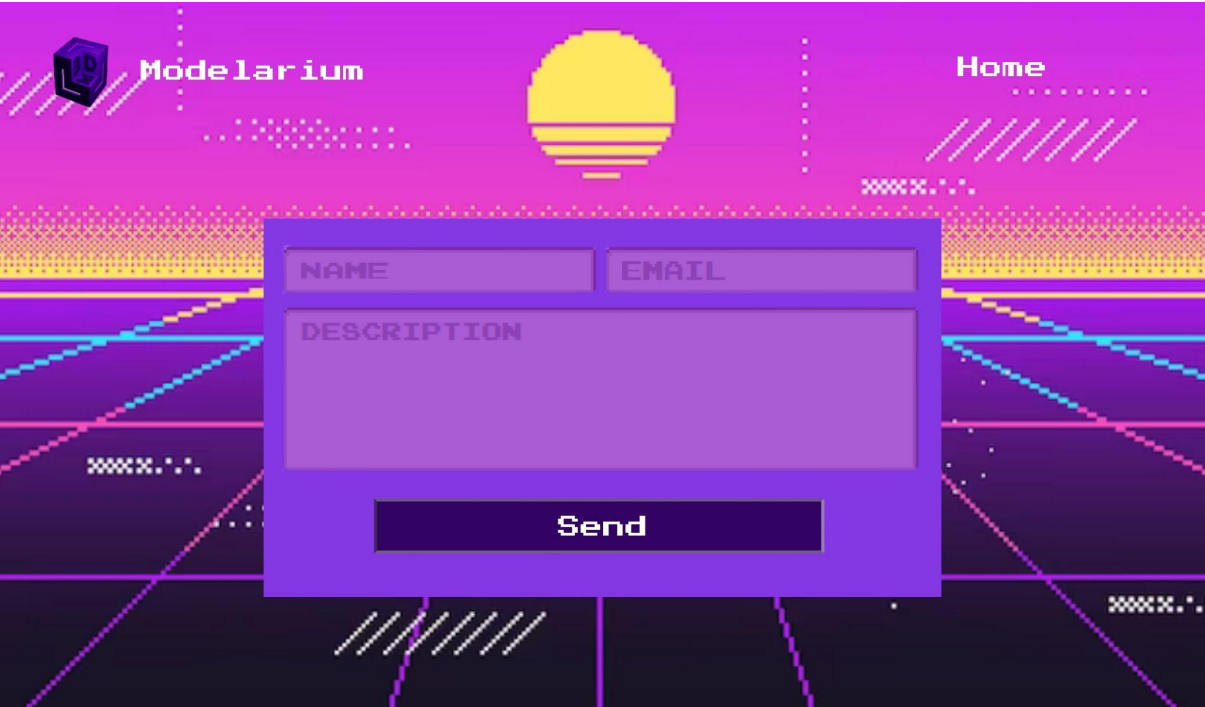


Fonts Used

Be Vietnam Pro, Arcade_N

Fonts Weight Used

Titles	Descriptions
Subtitles	Simple
Buttons	text






Modelarium 3D




Lorem ipsum dolor sit amet consectetur adipisicing elit. Assumenda laboriosam rerum, natus numquam veniam pariatur qui! Voluptatibus, vel. Eligendi tempora ipsa, numquam nostrum est obcaecati ullam necessitatibus minima voluptatibus deleniti?

[See models](#)[Login](#)



Modelarium


HomeModelsContact



Login




Login


Are you not registered? [Click here](#)





Modelarium

HomeModelsContact





Title

Author: [Author Name](#)  24 Likes  7 Favorites

Price


4,99€

Buy now


Description

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries,


Comments

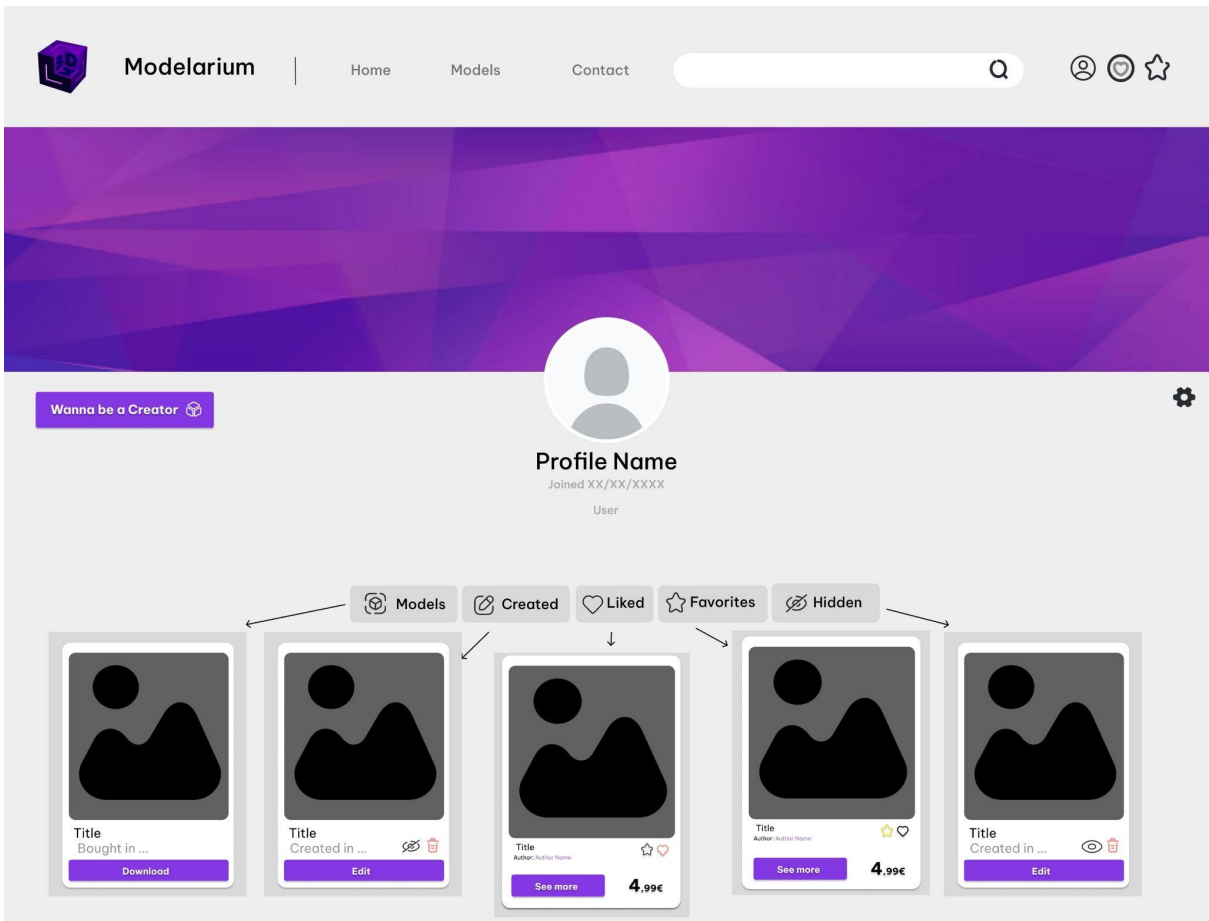
 You

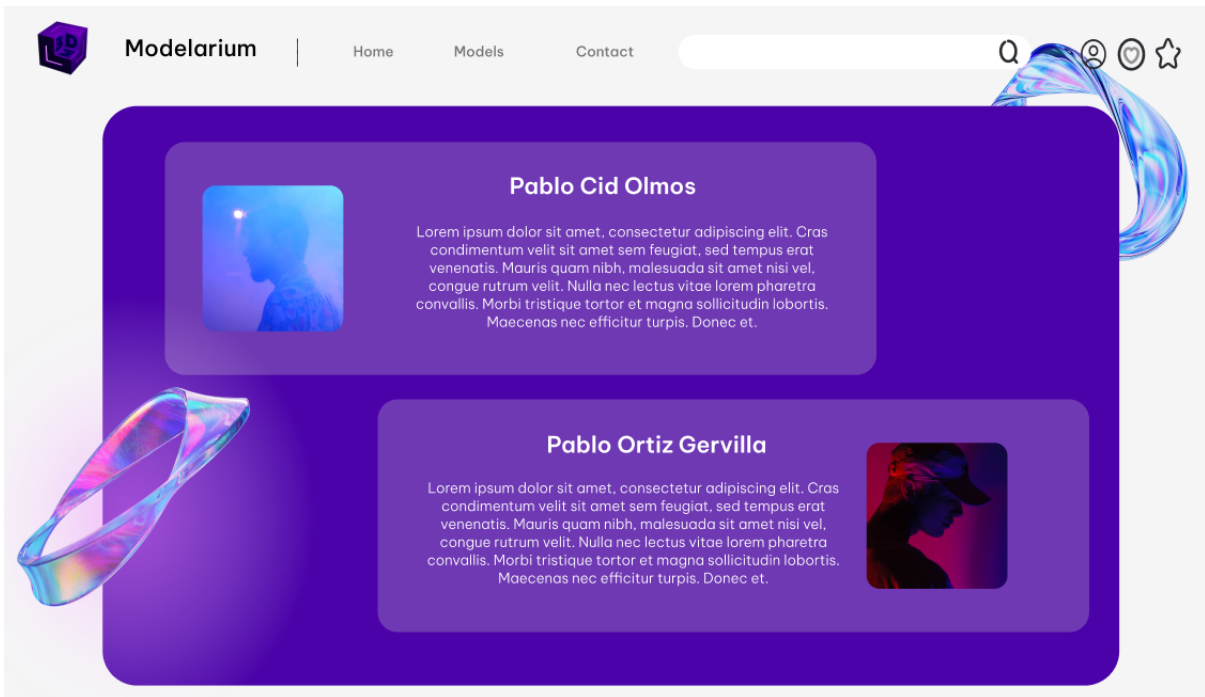
Comment

 Name

Lorem ipsum dolor sit amet consectetur adipiscing elit. Assumenda laboriosam rerum, natus numquam veniam pariatur quili Voluptatibus, vel. Eligendi tempora ipsa, numquam nostrum est obcaecati ullam necessitatibus







Software utilizado



[Blender](#) ha sido una herramienta imprescindible para **Modelarium3D**. Este software de código abierto nos ha permitido realizar modelos, animaciones y renderizados en 3D de manera eficiente y precisa.

Gracias a Blender, hemos podido crear y editar diversos modelos 3D, los cuales mostramos en la propia página web.

Además, el hecho de que Blender sea de código abierto y gratuito nos ha beneficiado mucho, ya que nos ha permitido ahorrar en gastos y acceder a una comunidad activa de usuarios y desarrolladores que comparten conocimientos y mejoras constantes.

En conclusión, Blender ha desempeñado un papel fundamental en nuestro **Modelarium3D**, permitiéndonos realizar modelos y renderizados de alta calidad, y llevar nuestra plataforma al siguiente nivel.

Y para mostrar los modelos en nuestra web necesitábamos de una herramienta nueva y también desconocida, aquí es cuando entra en escena [Three.js](#)



[Three.js](#) ha sido una biblioteca también muy esencial para **Modelarium3D**. Esta biblioteca, basada en JavaScript y WebGL, nos permitió crear una experiencia web inmersiva con una galería de modelos 3D que creamos en Blender.

Gracias a la documentación, ejemplos y recursos compartidos por la comunidad, hemos podido acelerar el proceso de desarrollo y nos ha permitido resolver desafíos de manera efectiva.

Posibles Mejoras

Aunque el proyecto está bastante completo hay varias cosas que no hemos sido capaces de implementar, más por inexperiencia en este campo que por no intentarlo.

1.- Nos habría gustado que el modelo de la habitación hubiese estado en la misma página y que la cámara saliese y entrase realmente en la sección de la página que le pertenece.

2.- También nos habría gustado que el usuario hubiera podido agarrar y mover el modelo 3D como quisiera en la vista `modelview.php`.

3.- Las animaciones de los objetos de la habitación habrían hecho que se viese mucho más llamativa y original, pero ya le habíamos dedicado demasiado tiempo al modelaje de objetos 3D y la animación es mucho más compleja y difícil de aprender, por eso tuvimos que dejar de lado esta idea, pero nos habría encantado que formase parte de el resultado final.

Como hemos comentado, seguro que somos capaces de implementar estas mejoras y seguro que encontramos muchas más, pero por falta de tiempo no hemos podido.