

PROYECTO:

Project Games



Realizado por:

Free Team

- *Álvaro Marcos Sánchez*
- *Sergio del Campo Alonso*
- *Walid Aouribi Chelki*



Desarrollo de Aplicaciones Web (DAW)

Valladolid, a 6 de febrero de 2026

Contenido

1.	Introducción	2
2.	Análisis	2
2.1.	<i>Dataset</i> utilizado	2
2.2.	Requisitos funcionales	4
2.3.	Diagramas de casos de uso.....	9
3.	Diseño.....	10
3.1.	Modelo de base de datos	10
3.2.	Diseño de la interfaz	13
4.	Desarrollo.....	17
4.1.	<i>Stack</i> tecnológico.....	17
4.2.	Estructura de carpetas	18
4.3.	Descripción del funcionamiento.....	19
5.	Pruebas	23
6.	Despliegue	24
7.	Sostenibilidad y "Green Coding"	28
8.	Conclusiones.....	30
8.1.	Autoevaluación	30
8.2.	Líneas futuras	30
9.	Bibliografía	30

1. Introducción

El proyecto consiste en una aplicación web centrada en el **listado de videojuegos gratuitos**, desarrollada a partir de un **conjunto de datos abierto sobre juegos free-to-play**. La temática principal es el **ocio digital accesible**, facilitando a los usuarios el descubrimiento de juegos sin coste.

La aplicación permite **registrarse e iniciar sesión**, consultar un **listado actualizado de juegos**, marcar títulos como **favoritos** y **publicar reseñas** con opiniones personales.

Su finalidad es cubrir la necesidad ciudadana de **acceder de forma centralizada, fiable y organizada a información sobre juegos gratuitos**, evitando la dispersión de datos en múltiples plataformas y ayudando a los usuarios a decidir qué juegos probar según las valoraciones de la comunidad.

2. Análisis

2.1. Dataset utilizado

Nombre: FreeToGame

Esa página es la documentación oficial de la **FreeToGame API**, una interfaz de programación que te permite obtener datos gratuitamente sobre videojuegos "Free-to-Play" (gratuitos) y MMOs.

URL: <https://www.freetogame.com/api-doc>

1. ¿Qué te permite hacer?

Te da acceso a una base de datos con más de 400 juegos. Puedes pedirle a su servidor (hacer un *request*) que te envíe información en formato JSON para usarla en tu web.

Lista de todos los juegos:

- Endpoint: /api/games
- Qué hace: Te devuelve una lista enorme con todos los juegos disponibles, incluyendo su título, miniatura (thumbnail), descripción corta y género.

Detalles de un juego específico:

- Endpoint: /api/game?id=452 (donde "452" es el ID del juego).
- Qué hace: Te da toda la ficha técnica: requisitos del sistema, capturas de pantalla, descripción larga, editor, etc.

Filtros (Muy útil para tu proyecto):

- Por plataforma: /api/games?platform=pc o /api/games?platform=browser (para separar juegos de PC o de navegador).
- Por categoría: /api/games?category=shooter (para obtener solo juegos de disparos, o estrategia, MOBA, etc.).
- Ordenar: /api/games?sort-by=release-date (para ver los más nuevos).

¿Por qué es genial para tu Stack Tecnológico?

Conectando esto con tu imagen anterior:

- **No requiere API Key (normalmente):** A diferencia de otras APIs complejas, esta suele ser abierta, lo que significa que puedes empezar a usarla inmediatamente con `fetch()` en tu código JavaScript sin registrarte ni pagar.
- **Tiene imágenes:** La API te devuelve URLs de imágenes (thumbnails), lo cual es perfecto para practicar con el componente `<Image />` de **Next.js** y darle estilos con **Tailwind**.

Campos utilizados:

- `id` → Identificador único del juego.
- `title` → Título o nombre del juego.
- `thumbnail` → URL de la imagen en miniatura del juego.
- `short_description` → Descripción breve.

- `game_url` → Enlace para jugar o ver más información.
- `genre` → Género o categoría del juego.
- `platform` → Plataforma en la que está disponible.
- `publisher` → Empresa que publica el juego.
- `developer` → Desarrolladora del juego.
- `release_date` → Fecha de lanzamiento.
- `freetogame_profile_url` → URL al perfil del juego en FreeToGame.com.

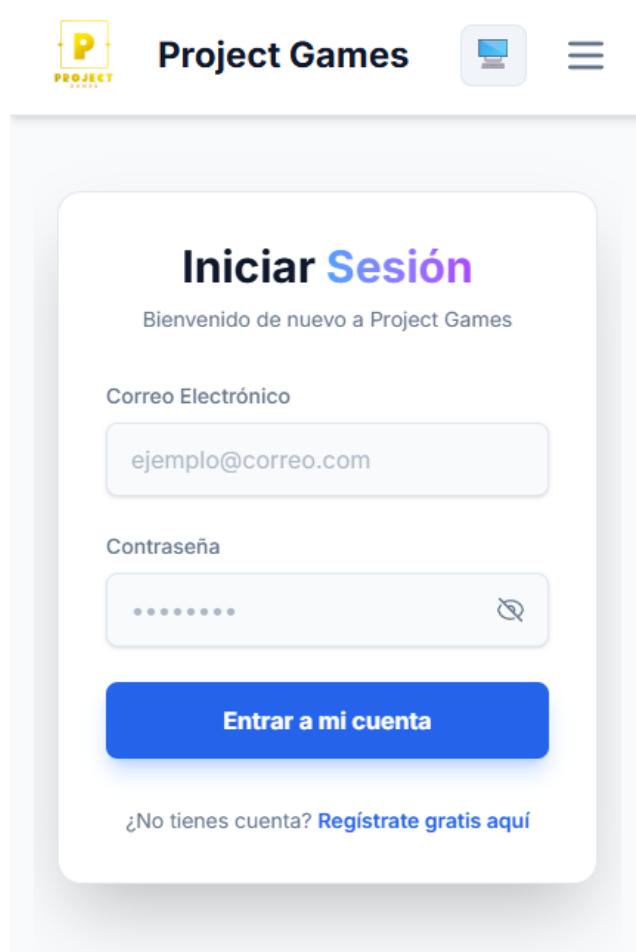
2.2. Requisitos funcionales

Listado de las funcionalidades implementadas.

- **Register (Registro):** Permite a un nuevo usuario crear una cuenta. Aquí es donde **PHP** recibe los datos (nombre, email, password) y los guarda en tu base de datos **MySQL**.

The screenshot shows the 'Crear Cuenta' (Create Account) page of the Project Games website. At the top, there's a header with the 'Project Games' logo and a menu icon. Below the header, the title 'Crear Cuenta' is displayed, followed by the sub-instruction 'Únete a la comunidad de Project Games'. The form consists of four input fields: 'NOMBRE' (User name), 'EMAIL' (Email address), 'CONTRASEÑA' (Password), and 'CONFIRMAR CONTRASEÑA' (Confirm Password). Each field has a placeholder text and a visibility toggle icon. Below the password fields is a reCAPTCHA section with a checkbox labeled 'No soy un robot', a reCAPTCHA logo, and links for 'Privacidad' and 'Términos'. A large blue 'Crear Cuenta' button is at the bottom of the form. At the very bottom, there's a link '¿Ya tienes cuenta? Inicia sesión aquí'.

- **Login (Inicio de sesión):** El proceso de autenticación. El sistema verifica que las credenciales coincidan con lo guardado en MySQL para dar acceso a funciones privadas como los "Favoritos".



Project Games

Tu biblioteca de juegos gratis favorita.

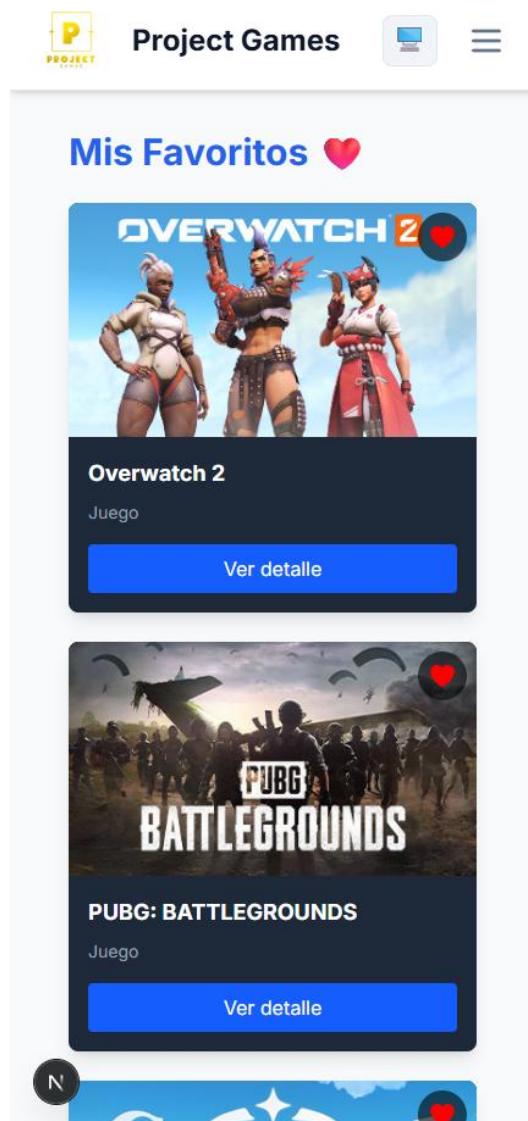
[Sobre nosotros](#) [Privacidad](#)

v1.0.0



© 2026 Project Games. Todos los derechos reservados.

- **Favoritos:** Permite al usuario "guardar" juegos de la API de FreeToGame que le gusten. En tu base de datos, tendrías una tabla que relacione el ID_usuario con el ID_juego de la API.



- **Search (Búsqueda):** Un campo de texto donde el usuario escribe el nombre de un juego. Next.js enviaría esa consulta a la API para mostrar solo los resultados que coincidan.

The screenshot shows the 'Listado de juegos' (Game List) screen of the Project Games application. At the top, there is a navigation bar with the 'Project Games' logo, a computer icon, and a menu icon. Below the navigation bar, there are three search/filter input fields: 'Buscar por nombre...', 'Plataforma', and 'Género'. The main content area displays two game cards. The first card is for 'Overwatch 2', featuring three female characters from the game. The card includes the title 'Overwatch 2', the genre 'Shooter', and a blue 'Ver detalle' (View details) button. The second card is for 'PUBG', showing a group of players in a battle royale setting with parachutes in the background. A small circular icon with the letter 'N' is visible in the bottom-left corner of the PUBG card.

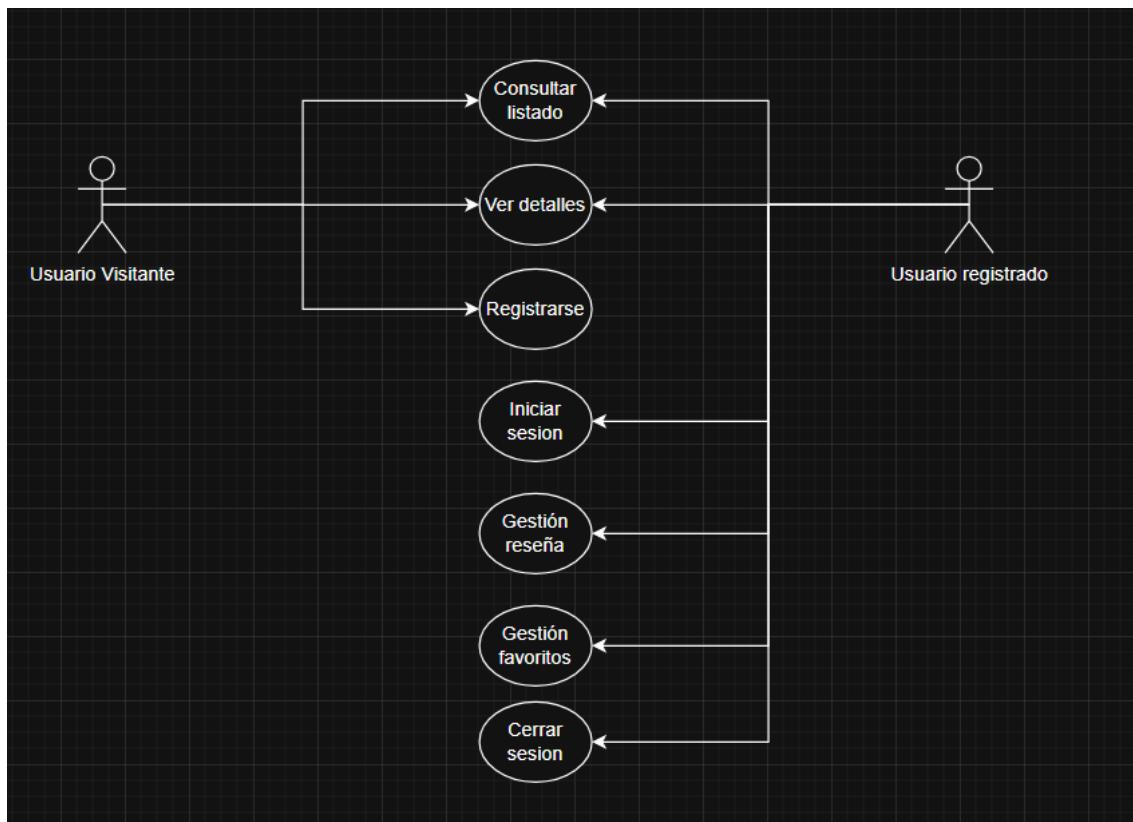
- **Filter (Filtros):** Utiliza los parámetros de la API (como `/api/games?category=shooter`) para organizar los juegos por género, plataforma o fecha de lanzamiento.
- **Reseñas:** Los usuarios pueden escribir su opinión sobre un juego. Esto requiere que crees una tabla en MySQL para almacenar los comentarios y luego mostrarlos en la ficha del juego.

The screenshot shows a user profile page for 'alv'. At the top, there's a yellow circular icon with a blue border containing a sad face emoji. Below it, the username 'alv' is displayed in bold black font, followed by the email 'alv@gmail.com'. A description 'prueba video' is present. Two buttons are shown: a light blue one with a person icon and the text '23 años', and a light blue one with a male symbol and the text 'Masculino'. A large blue button labeled 'Editar Perfil' is centered below these. Below this section, there are two tabs: 'Favoritos (4)' and 'Mis Reseñas (6)', with 'Mis Reseñas' being the active tab. Under 'Mis Reseñas', there are two reviews: one for 'Lost Ark' (4/5 stars, 'Buen juego', published 5/2/2026) and one for 'War Thunder' (1/5 stars, 'unoo', published 5/2/2026). A small circular icon with the letter 'N' is visible on the left side.

- **Perfil:** Una sección donde el usuario ve sus datos personales, su lista de favoritos y las reseñas que ha escrito.



2.3. Diagramas de casos de uso



3. Diseño

3.1. Modelo de base de datos

Nombre	Fields	Tamaño	Creado	Actualizado	Motor	Comentario	Tipo
cache	0	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
cache_locks	0	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
failed_jobs	0	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
favorites	17	48,0 KiB	2026-02-02 13:10:18		InnoDB		Table
games	405	192,0 KiB	2026-02-02 13:10:18		InnoDB		Table
jobs	0	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
job_batches	0	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
migrations	9	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
password_reset...	0	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
personal_acces...	0	16,0 KiB	2026-02-02 13:10:18		InnoDB		Table
reviews	8	48,0 KiB	2026-02-03 13:01:25		InnoDB		Table
sessions	209	176,0 KiB	2026-02-02 13:10:18		InnoDB		Table
users	8	32,0 KiB	2026-02-02 13:10:18		InnoDB		Table

FAVORITOS

Columnas:	+	Agregar	X	Borrar	▲ Subir	▼ Bajar			
#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado	Comentario	Collation
1	id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT	
2	user_id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...	
3	game_id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...	
4	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
5	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	

GAMES

Columns:										
#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado	Comentario	Collation	Expres
1	id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	external_id	INT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>		
3	title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>		<i>utf8mb4_unicode_ci</i>
4	thumbnail	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
5	short_descripti...	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
6	game_url	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
7	genre	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
8	platform	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
9	publisher	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
10	developer	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
11	release_date	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
12	freetogame_pr...	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
13	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		
14	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		

REVIEWS

Columns:										
#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado	Comentario	Collation	Expres
1	id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	user_id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>		
3	game_id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>		
4	comment	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>		<i>utf8mb4_unicode_ci</i>
5	rating	TINYINT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'5'		
6	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		
7	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		

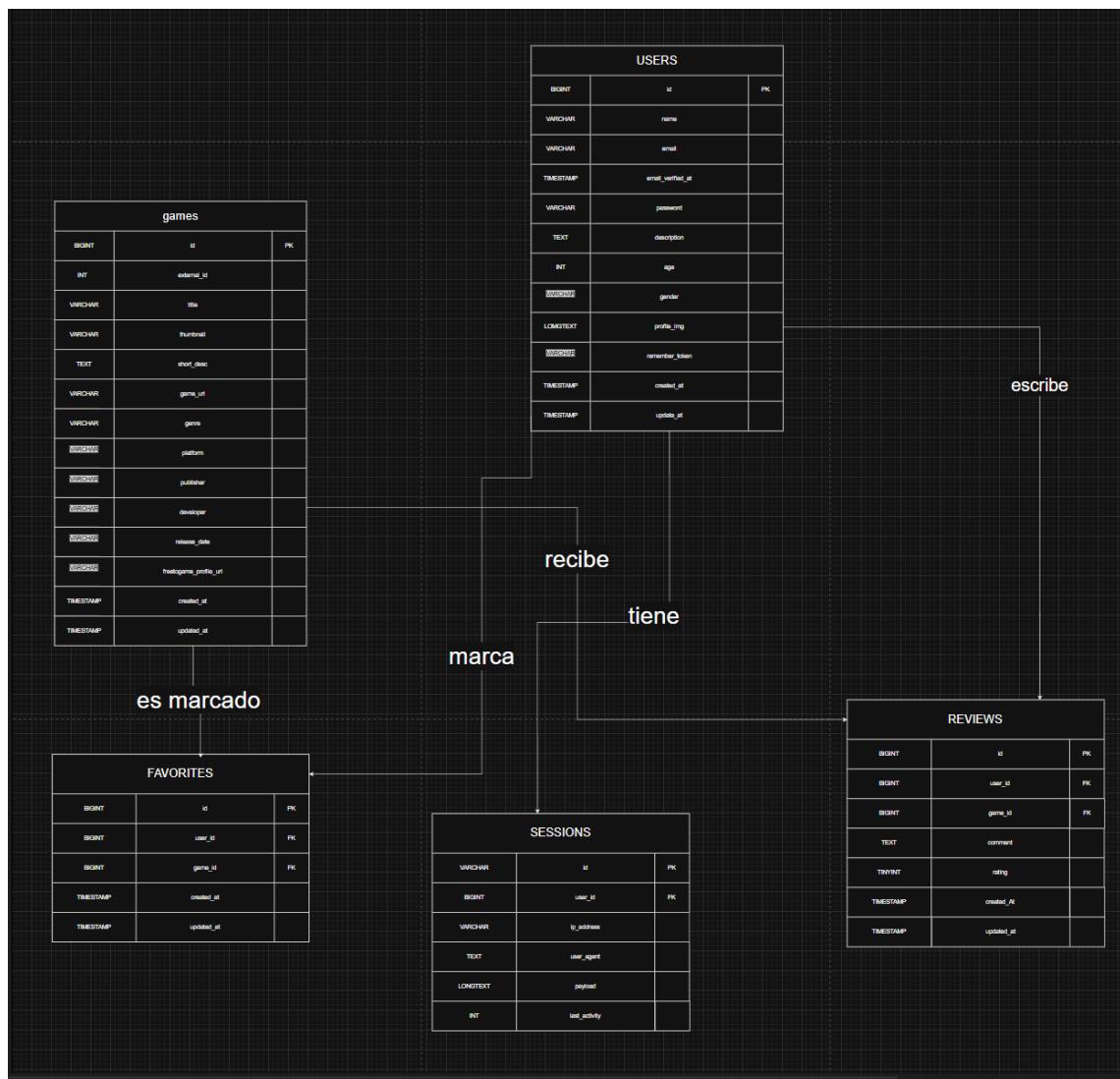
SESSIONS

Columns:										
#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado	Comentario	Collation	Expres
1	id	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>	<i>utf8mb4_unicode_ci</i>	
2	user_id	BIGINT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		
3	ip_address	VARCHAR	45	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
4	user_agent	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<i>NULL</i>		<i>utf8mb4_unicode_ci</i>
5	payload	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>		<i>utf8mb4_unicode_ci</i>
6	last_activity	INT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<i>Sin valor predeter...</i>		

USERS

Columnas:		Agregar	Borrar	Subir	Bajar				
#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado	Comentario	Collation
1	id	BIGINT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	name	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...		utf8mb4_unicode_ci
3	email	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...		utf8mb4_unicode_ci
4	email_verified_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
5	password	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sin valor predeter...		utf8mb4_unicode_ci
6	description	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci
7	age	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
8	gender	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci
9	profileImage	LONGTEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci
10	remember_token	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci
11	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
12	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

- **Diagrama Entidad-Relación:**



- **Justificación:** Explicación de las tablas creadas (ej. usuarios, favoritos) y cómo se relacionan.

El diseño de la base de datos se ha realizado siguiendo el modelo relacional para garantizar la integridad, eficiencia y escalabilidad del sistema. A partir del diagrama Entidad-Relación presentado, se estructuran las siguientes tablas principales que dan soporte a la lógica de negocio de la aplicación:

1. Tablas de Entidad Principal

- **Users (Usuarios):** Esta tabla almacena la información relativa a los usuarios registrados en la plataforma. Además de los campos esenciales para la autenticación y seguridad, como email, password y remember_token, se han incluido atributos para enriquecer el perfil social del usuario dentro de la comunidad: description, age, gender y profile_img. El campo email_verified_at permite gestionar la confirmación de cuentas.
- **Games (Videojuegos):** Actúa como el catálogo central de la aplicación. Aunque los datos provienen de una API externa, es necesario persistir esta información localmente para establecer relaciones con usuarios (favoritos y reseñas) sin depender constantemente de peticiones externas.
 - Destaca el campo **external_id**, que almacena el identificador original de la API *FreeToGame*, diferenciándolo de la clave primaria id interna de nuestra base de datos.
 - Se guardan detalles estáticos como title, thumbnail, genre, platform y developer para agilizar la carga de las vistas.

2. Tablas de Interacción y Relación

- **Reviews (Reseñas):** Esta tabla materializa la relación entre usuarios y juegos, permitiendo a la comunidad aportar contenido.
 - Relaciona mediante claves foráneas a un usuario (user_id) con un juego específico (game_id).
 - Almacena la opinión cualitativa en el campo comment y la valoración cuantitativa en rating (tipo TINYINT, para puntuaciones numéricas).
- **Favorites (Favoritos):** Tabla pivote que resuelve la relación de cardinalidad N:M (muchos a muchos) entre usuarios y videojuegos.
 - Su función es estrictamente relacional, almacenando pares de user_id y game_id para permitir a cada usuario gestionar su propia biblioteca personalizada de títulos de interés.

3. Tablas de Infraestructura y Sesión

- **Sessions (Sesiones):** Tabla técnica necesaria para la gestión de la persistencia de usuarios en el sistema. Almacena información crítica de seguridad y auditoría como la dirección IP (ip_address), el navegador utilizado (user_agent) y la fecha de última actividad (last_activity), vinculándolas al usuario correspondiente mediante user_id.

Relaciones del Sistema

El modelo define las siguientes relaciones estructurales:

- **Usuarios - Reseñas (1:N):** Un usuario puede redactar múltiples reseñas, pero cada reseña pertenece a un único autor.
- **Juegos - Reseñas (1:N):** Un mismo videojuego puede recibir múltiples valoraciones de distintos usuarios.
- **Usuarios - Favoritos - Juegos (N:M):** Un usuario puede marcar múltiples juegos como favoritos y, a su vez, un juego puede aparecer en la lista de favoritos de múltiples usuarios. Esta relación se gestiona mediante la tabla intermedia favorites.

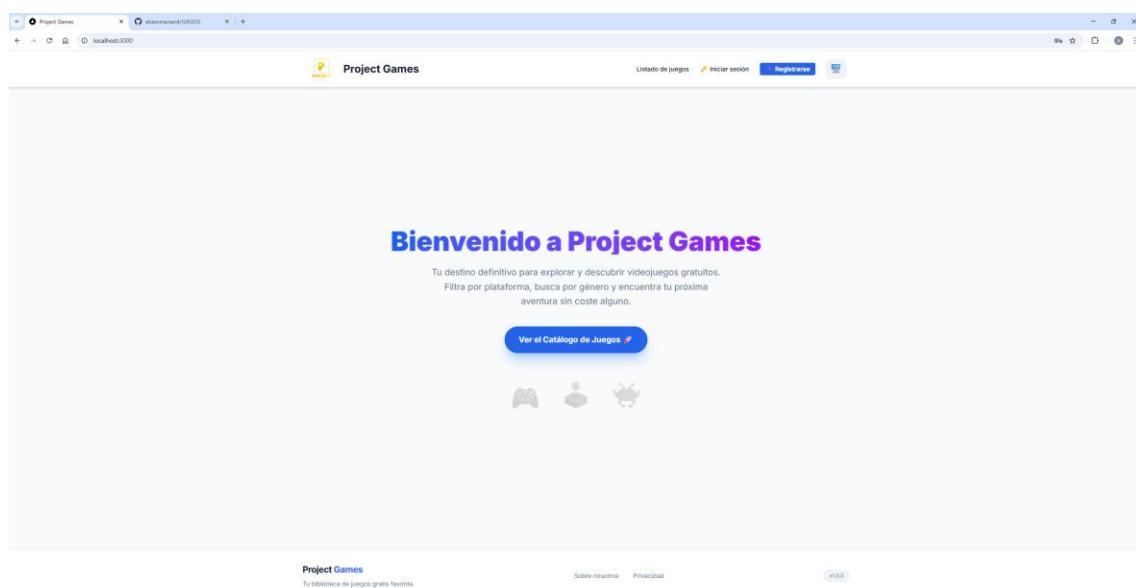
3.2. Diseño de la interfaz

- **Prototipo** del *layout* principal para las vistas *desktop* y *mobile*.

DESKTOP/PC

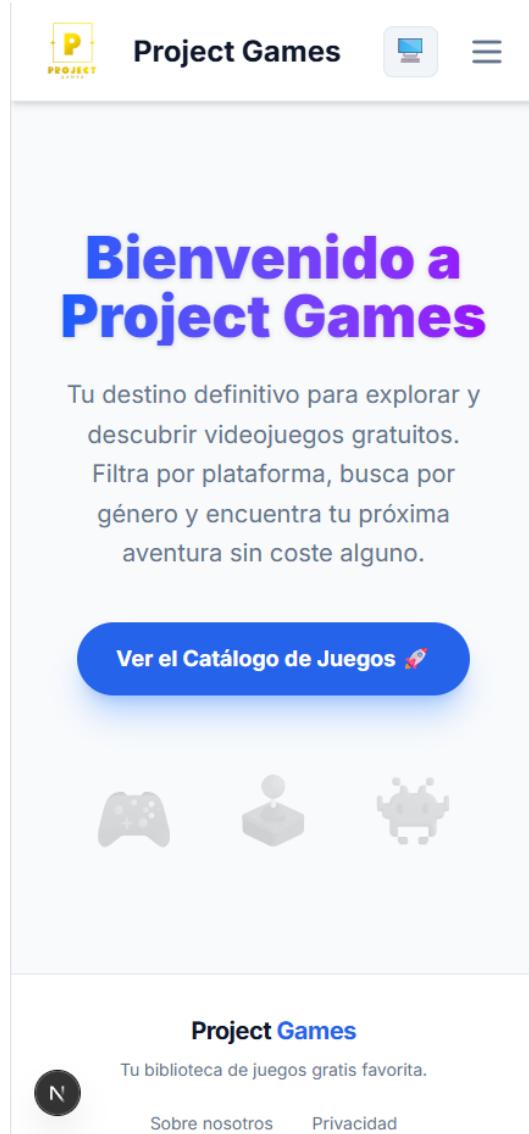
INICIO:

Desktop: Se ve la barra de menú completa arriba y el contenido espaciado.



MOVIL

Mobile: Se ve cómo el menú cambia al icono de hamburguesa y el texto se ajusta al ancho



- Breve **guía de estilos** general que incluya la paleta de colores utilizada y la selección de fuentes.

Colores y tipografía

Para el desarrollo de la interfaz de "Project Games", se ha optado deliberadamente por la implementación de una familia tipográfica de estilo *sans serif* (sin remates). Esta elección no es meramente estética, sino que responde a la necesidad funcional de garantizar una legibilidad óptima en entornos digitales. Dada la naturaleza tecnológica del proyecto —una biblioteca de videojuegos web—, la tipografía *sans serif* aporta una apariencia limpia, moderna y minimalista, eliminando el ruido visual y permitiendo que el usuario se centre en el contenido (las portadas y títulos de los juegos). Además, su versatilidad asegura una visualización nítida y fluida tanto en encabezados de gran tamaño como en textos descriptivos más pequeños, adaptándose coherentemente a la diversidad de dispositivos desde los que accederá el público objetivo.

- **Colores:**



#2563EB

#FFFFFF



#111827



#6B7280



#EF4444

- **Tipografía:** Sans Serif

4. Desarrollo

4.1. Stack tecnológico

Entorno: Especificar stack tecnológico y versión utilizada. Ej. Lenguajes utilizados en cada parte de la aplicación, librerías o frameworks utilizados.

Aquí se listan las tecnologías que hacen que la aplicación funcione. Es una mezcla curiosa porque combina tecnologías modernas de JavaScript con tecnologías clásicas de servidor.

- **Next.js v16.1.5** : Es el "esqueleto" y los "músculos" de la parte visual de tu web. Es un framework moderno (basado en React) que permite crear sitios web muy rápidos y dinámicos.
- **Tailwind v4.0.0** : Es la "pintura y decoración". Se usa para dar estilo (colores, márgenes, tamaños) de forma muy rápida directamente en el código.
- **Laragon v8.2.3** : Es el "terreno de construcción". Es un programa para Windows que instala y gestiona automáticamente cosas complejas como el servidor y la base de datos para que no tengas que configurarlas una por una.
- **Apache v2.4.62** : Es el "camarero". Es el software del servidor web que recibe las peticiones de los usuarios y les entrega las páginas web.
- **MySQL v8.4.3** : Es el "archivador". Es la base de datos donde se guarda toda la información (usuarios, contraseñas, productos, etc.).

- **PHP v8.4.16**: Es el "cerebro" del servidor. Es el lenguaje de programación que procesa los datos, habla con la base de datos y decide qué mostrar.

Software: Estos son los programas que el programador usa para construir la aplicación, pero que no son parte de la aplicación en sí.

- **Visual Studio Code**: Es el "taller". El editor de texto donde se escribe todo el código.
- **Gemini**: Es el "asistente inteligente". La IA utilizada para ayudar a generar código, resolver dudas o redactar documentación.
- **Antigravity**: Esto es menos común. Una herramienta de utilidad interna del proyecto para facilitar tareas específicas. En este contexto, parece ser una utilidad de apoyo.

4.2. Estructura de carpetas

Enumerar la estructura de archivos final añadiendo una breve explicación del contenido de cada directorio o fichero.

```

PROYECTO-GAMES
|
└── freec1/ (VISTA - Frontend Next.js)
    ├── src/app/ → [Páginas y Rutas]
    │   ├── page.js (Inicio) & layout.js (Estructura)
    │   ├── games/[id]/page.js (Detalle del juego)
    │   ├── favoritos/page.js, login/page.js, reviews/page.js
    │   └── Profile/page.js, register/page.js, about/page.jsx
    ├── src/components/ → [Componentes UI]
    │   └── Navbar.jsx, GameCard.jsx, Gamelist.jsx, Filters.jsx, Footer.jsx
    ├── src/context/ → [Estado Global]
    │   └── AuthContext.js (Sesión de usuario)
    └── src/services/ → [Puente API]
        └── api.js (Peticiones al servidor Laravel)

    └── freesv/ (CONTROLADOR Y MODELO - Backend Laravel)
        ├── app/Http/Controllers/ → [CONTROLADORES - Lógica]
        │   ├── GameController.php, FavoriteController.php, ReviewController.php
        │   ├── UserController.php, AboutController.php, AuthController.php
        │   └── Auth/ (LoginController.php, RegisterController.php)
        ├── app/Models/ → [MODELOS - Datos]
        │   ├── User.php, Game.php, Favorite.php, Review.php
        ├── app/Http/Requests/ → [Validación de entrada]
        │   └── LoginRequest.php, RegisterRequest.php, ApiRequest.php
        ├── app/Services/ → [Lógica Externa]
        │   └── FreeToGameService.php (Consumo de API de terceros)
        ├── database/ → [Estructura y Datos]
        │   ├── migrations/ (Tablas: create_games_table.php, etc.)
        │   └── seeders/ (Carga inicial: GameSeeder.php)
        └── routes/ → [Rutas de Acceso]
            └── api.php (Endpoints que usa el archivo api.js de Next.js)

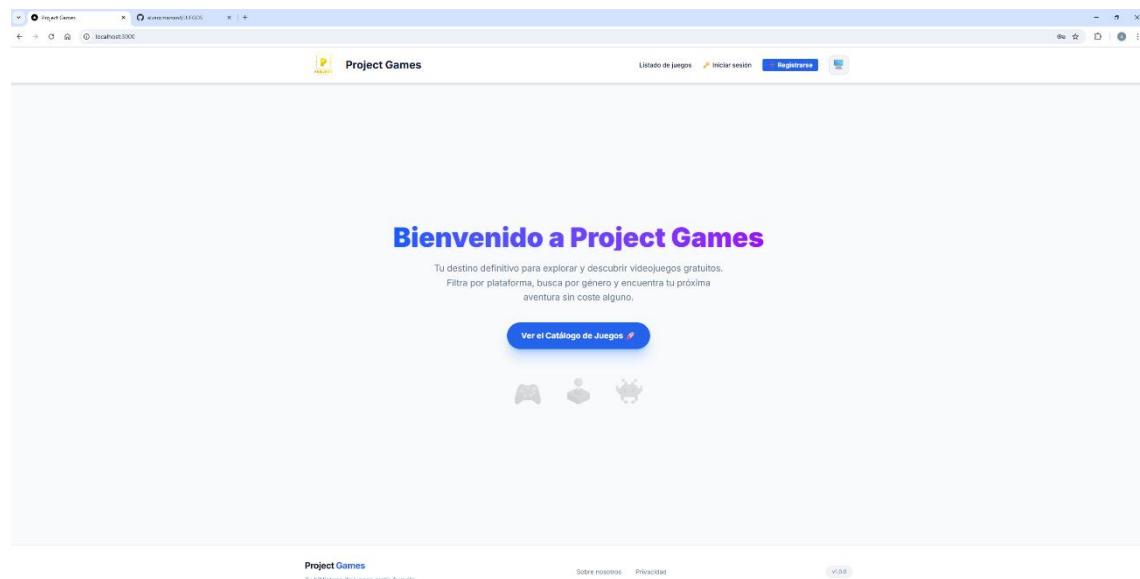
└── Configuración: .env (Backend), .env.local (Frontend), jsconfig.json

```

4.3. Descripción del funcionamiento

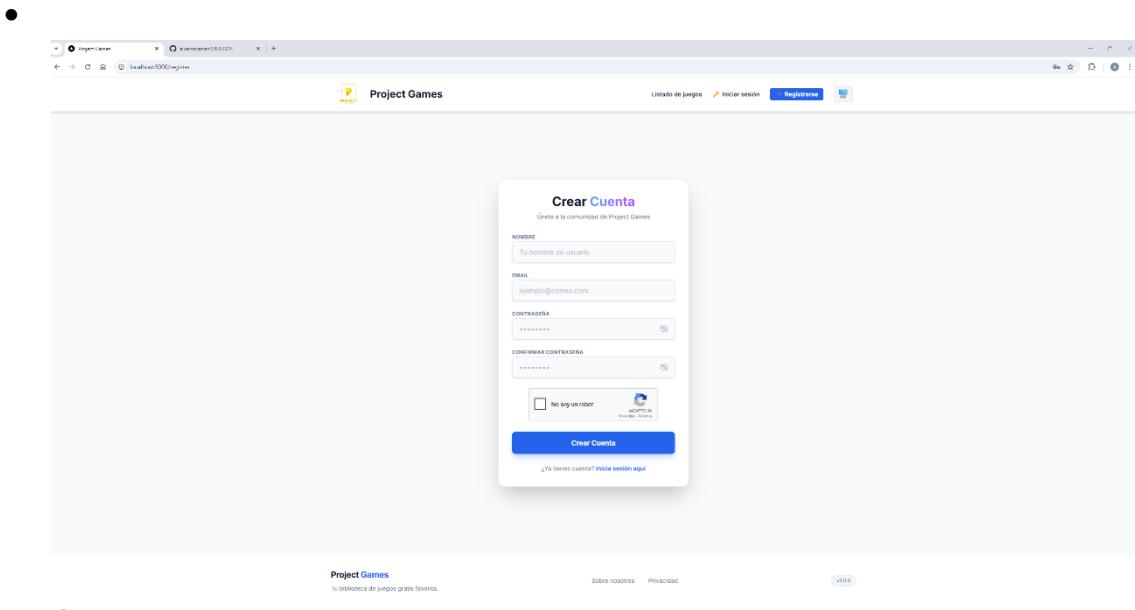
Breve manual de usuario.

1. Página de Bienvenida



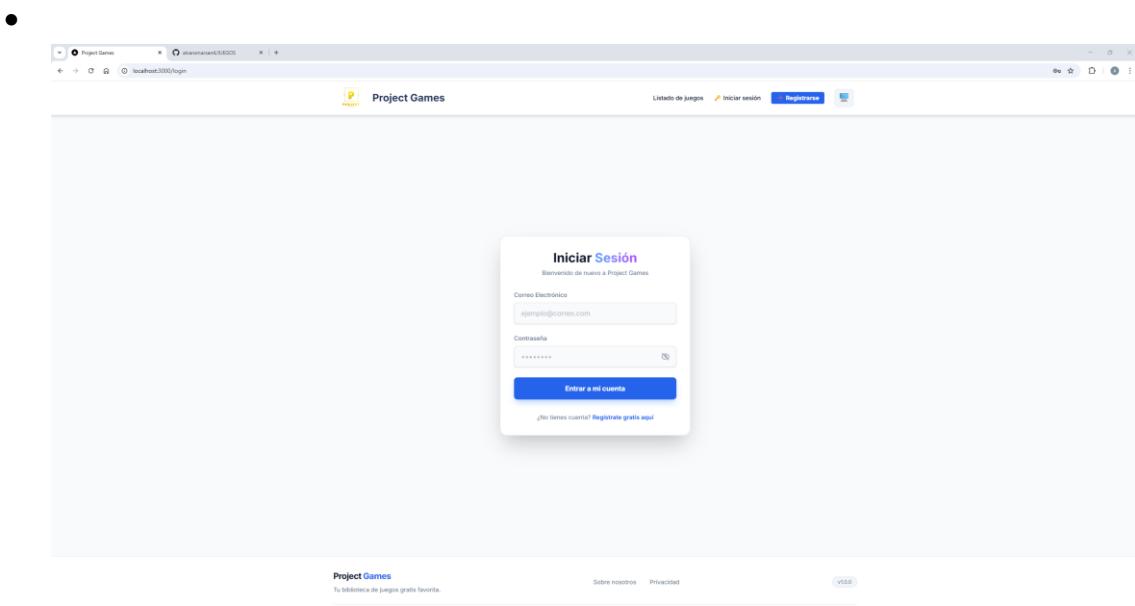
-  **Acción:** Pantalla inicial de carga.
- "Al acceder a la web, serás recibido por la página de inicio. Pulsa el botón central '**Ver el Catálogo de Juegos** 

2. Registro de Cuenta



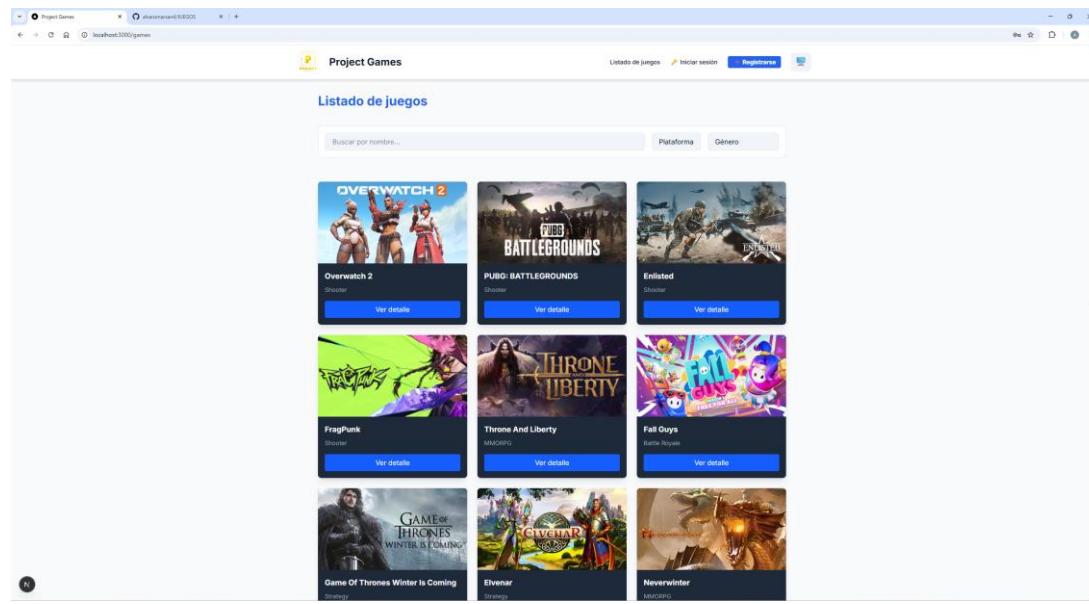
- **Acción:** Creación de un nuevo perfil.
- "Si no tienes cuenta, ve a la sección '**Registrarse**'. Deberás introducir tu nombre de usuario, email y contraseña. No olvides marcar la casilla de verificación '**No soy un robot**' para completar el proceso de seguridad."

3. Inicio de Sesión



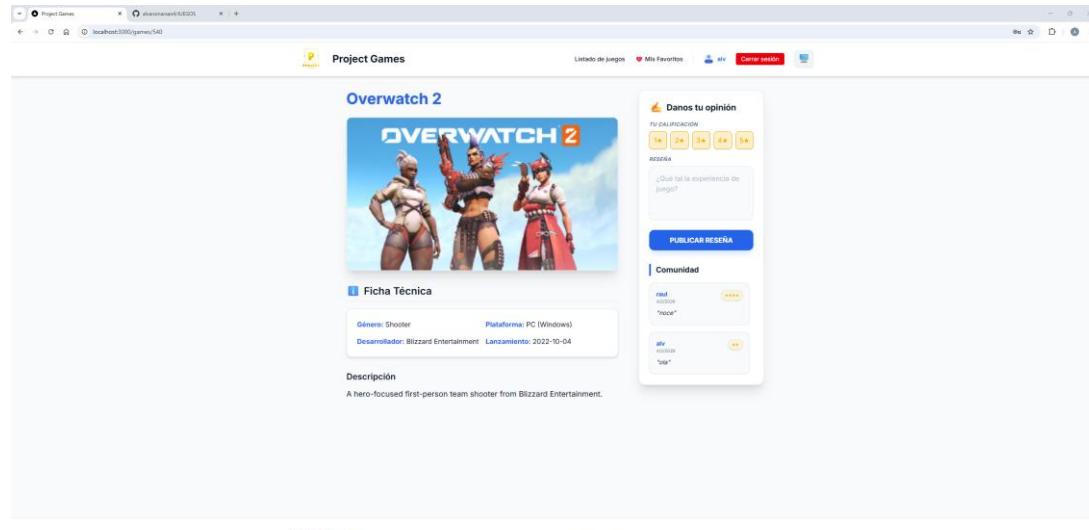
- **Acción:** Acceso para usuarios existentes.
- "Desde la pestaña '**Iniciar sesión**', introduce tus credenciales para acceder a tus funciones personalizadas como favoritos y reseñas."

4. Exploración del Catálogo y Filtros



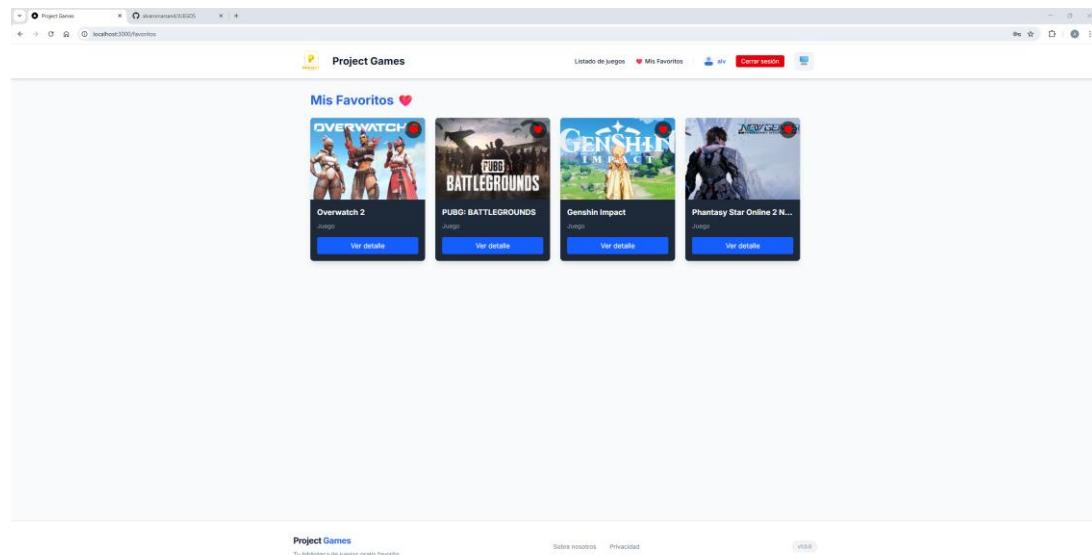
-
- **Acción:** Búsqueda de contenido.
- **Texto para el manual:** "En el '**Listado de juegos**', puedes usar la barra de búsqueda para encontrar títulos específicos o utilizar los botones de '**Plataforma**' y '**Género**' para filtrar los resultados según tus preferencias."

5. Ficha Detallada del Juego y Comunidad



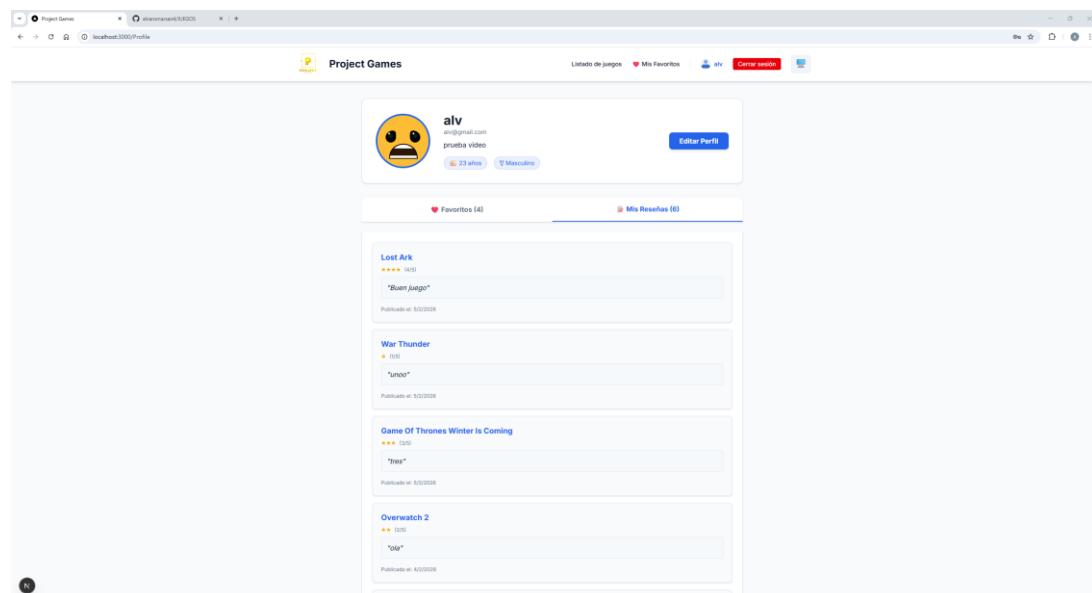
-
- **Acción:** Consultar información y opinar.
- "Al hacer clic en '**Ver detalle**', accederás a la ficha técnica del juego. Aquí puedes leer la descripción y ver las opiniones de la '**Comunidad**'. Para dejar tu reseña, selecciona una puntuación de estrellas, escribe tu comentario y pulsa '**PUBLICAR RESEÑA**'."

6. Gestión de Favoritos



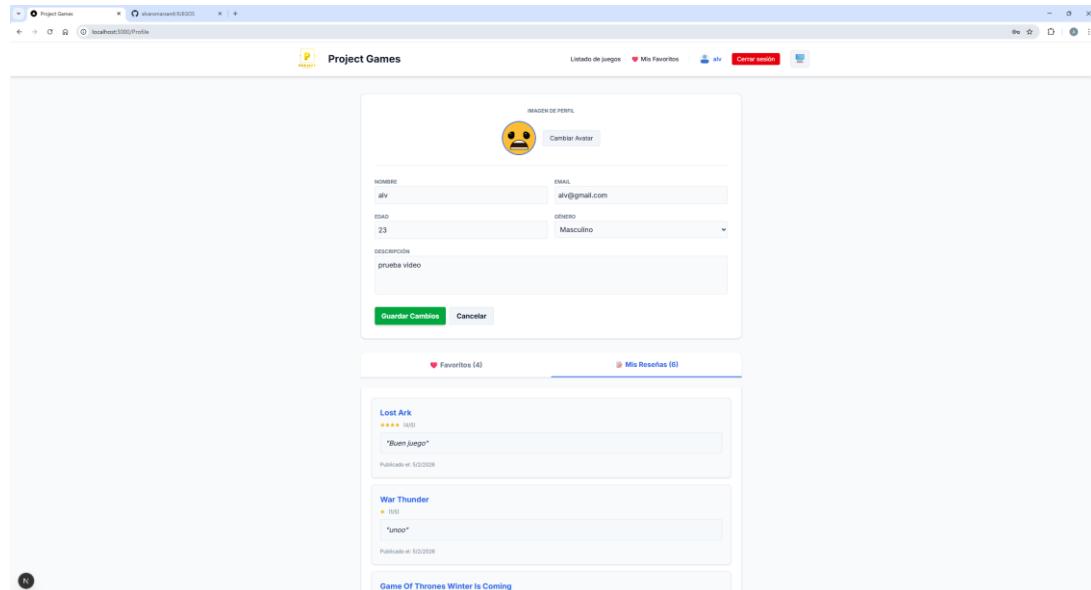
-
- **Acción:** Organización personal.
- "Puedes marcar tus juegos preferidos pulsando el ícono del **corazón rojo** en la tarjeta del juego. Todos estos títulos se guardarán automáticamente en tu sección de '**Mis Favoritos**' para un acceso rápido."

7. Perfil de Usuario y Reseñas Propias



-
- **Acción:** Visualización del estado del usuario.
- "En tu perfil, puedes ver un resumen de tu actividad: tus juegos favoritos y el historial de todas las reseñas que has publicado, junto con la fecha y la puntuación otorgada."

8. Edición de Datos Personales



- **Acción:** Configuración del perfil.
- "Para personalizar tu cuenta, pulsa en '**Editar Perfil**'. Podrás cambiar tu avatar, actualizar tu correo, edad, género y añadir una descripción personal. Pulsa '**Guardar Cambios**' para finalizar."

5. Pruebas

Documentar una breve prueba de **usabilidad** en la que se encomienda realizar al menos una tarea a un usuario y un análisis automático de **accesibilidad**. Ej. Con WAVE [1].

¿Qué se prueba?	Resultado esperado	Resultado obtenido
Buscador de juegos: Introducir el nombre de un juego (ej. "Overwatch") en la barra de búsqueda.	La lista debe filtrarse automáticamente para mostrar solo los juegos que coincidan con el texto introducido.	OK
Eliminación de Favoritos: Hacer clic en el ícono del corazón en un juego que ya es favorito.	El juego debe desaparecer de la lista de "Mis Favoritos" y el contador del perfil debe reducirse.	OK

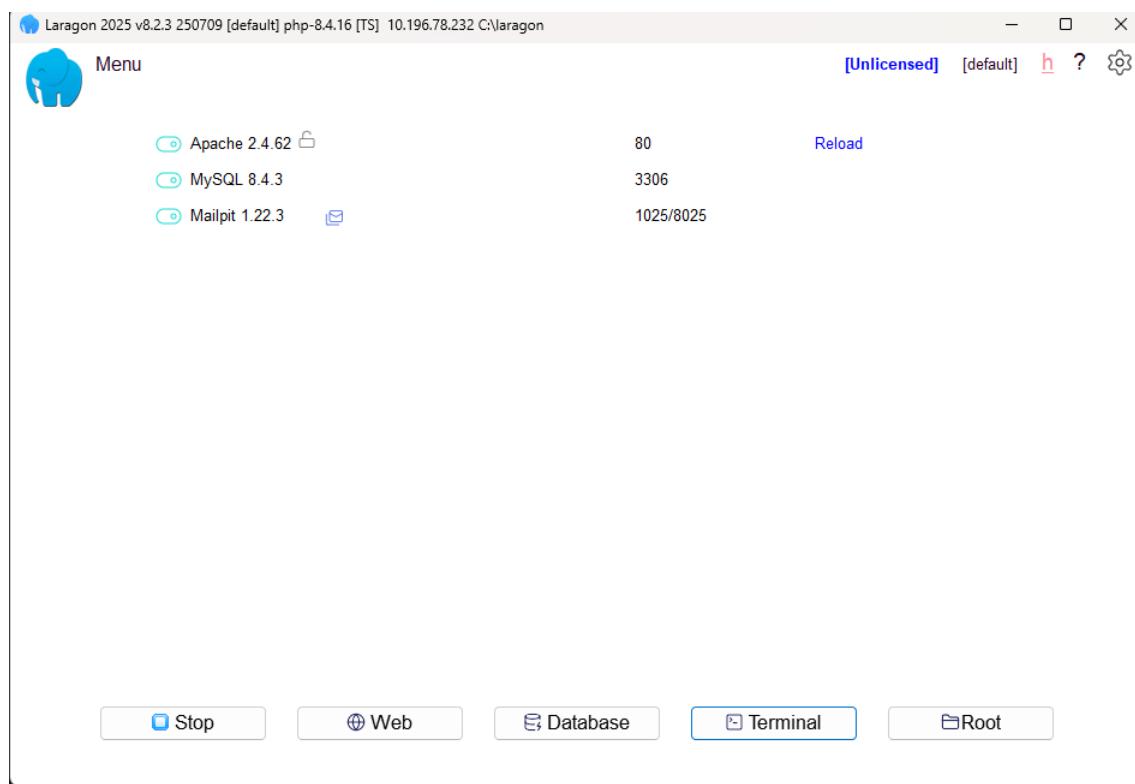
Cierre de Sesión: Pulsar el botón "Cerrar sesión" en la barra de navegación superior.	El usuario debe ser desconectado y redirigido a la página de inicio o login, ocultando las opciones privadas.	OK
Integridad de Reseñas: Publicar una reseña y verificar que el nombre del autor sea el correcto.	La reseña en la sección "Comunidad" debe mostrar el nombre del usuario logueado (ej. "alv") y la calificación elegida.	OK
Adaptabilidad de Avatar: Cambiar la imagen de perfil desde la configuración.	El nuevo avatar debe visualizarse correctamente tanto en la página de perfil como en los comentarios de los juegos.	OK
Validación de Login: Intentar acceder con un correo no registrado o contraseña incorrecta.	El sistema debe mostrar un mensaje de error y denegar el acceso a la cuenta.	OK
Navegación por Pestañas: Cambiar entre las pestañas "Favoritos" y "Mis Reseñas" en el perfil.	El contenido inferior debe comutar entre la lista de juegos y la lista de comentarios sin recargar la página completa.	OK

6. Despliegue

Instrucciones de instalación en local:

Paso 1:

Ejecutar laravel



Instalación Servidor

Paso 2:

Abrir dos terminales (Para ejecutar el servidor y el cliente)



Paso 3:

Crear la estructura de carpetas con

```
C:\laragon\www(master)
λ mkdir proyecto-games
```

```
C:\laragon\www\Proyecto-games(main -> origin)
λ composer create-project laravel/laravel freesv
tjs Laragon
```

```
C:\laragon\www\Proyecto-games(main -> origin)
λ cd freesv
vis Laragon
C:\laragon\www\Proyecto-games\freesv(main -> origin)
λ
Laragon Next
```

Paso 4:

Cambiar el .env de freesv

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=TFG_games
DB_USERNAME=root
DB_PASSWORD=
```

Paso 5:

Instalar el composer para que tenga la versión más nueva

```
C:\laragon\www\Proyecto-games\freesv(main -> origin)
λ composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Nothing to install, update or remove
Generating optimized autoload files
Class App\Http\Controllers\Api\AboutController located in ./app/Http/Controllers/AboutController.php
with psr-4 autoloading standard (rule: App\ => ./app). Skipping.
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

  INFO  Discovering packages.

  laravel/pail ...
  laravel/sail ...
  laravel/sanctum ...
  laravel/tinker ...
  nesbot/carbon ...
  nunomaduro/collision ...
  nunomaduro/termwind ...

81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!

C:\laragon\www\Proyecto-games\freesv(main -> origin)
λ
```

Paso 6:

Generar la clave de aplicación

```
C:\laragon\www\Proyecto-games\freesv(main -> origin)
λ php artisan key:generate
```

```
INFO Application key set successfully.
```

```
C:\laragon\www\Proyecto-games\freesv(main -> origin)
λ |
```

Paso 7:

Iniciar el servidor

```
C:\laragon\www\Proyecto-games\freesv(main -> origin)
λ php artisan serve
```

```
INFO Server running on [http://127.0.0.1:8000].
```

```
Press Ctrl+C to stop the server
```

Instalacion cliente

Paso 8:

Crear la estructura de carpetas

```
C:\laragon\www\Proyecto-games(main -> origin)
λ npx create-next-app@latest freecl
```

```
C:\laragon\www\Proyecto-games(main -> origin)
λ cd freecl
```

```
C:\laragon\www\Proyecto-games\freecl(main -> origin)
λ |
```

Paso 9:

Crear .env.local y rellenarlo con estos datos

```
NEXT_PUBLIC_API_URL=http://127.0.0.1:8000/api
NEXT_PUBLIC_RECAPTCHA_SITE_KEY=6Lf7LmAsAAAAAMc-UyxKj1VAWKaQSzq0I6wUK0Uw
```

Paso 10:

Ejecutar la parte cliente

```
C:\laragon\www\Proyecto-games\freecl(main -> origin)
λ npm run dev

> freecl@0.1.0 dev
> next dev

▲ Next.js 16.1.5 (Turbopack)
- Local:          http://localhost:3000
- Network:        http://10.196.78.232:3000
- Environments:  .env.local

✓ Starting...
|
```

- **URLs:**

- Del repositorio de Git: <https://github.com/alvaromarsan4/JUEGOS>
- Del sitio web puesto en producción: http://localhost:3000

7. Sostenibilidad y "Green Coding"

Justificar las decisiones tomadas para reducir el impacto ambiental:

7.1 Estrategia de Caché

En el proyecto se ha utilizado la **estrategia de caché predeterminada** que ofrecen los frameworks Laravel y Next.js, sin realizar configuraciones personalizadas adicionales.

La justificación de esta decisión se basa en el uso de frameworks que ya están optimizados para un funcionamiento eficiente, evitando implementar soluciones adicionales que podrían aumentar la complejidad sin un beneficio proporcional.

Por parte del backend, Laravel incorpora de forma nativa mecanismos de optimización como:

- **Caché de configuración y rutas**, que evita recalcular la configuración de la aplicación y el enrutado en cada petición.
- **Uso del caché interno del framework** para la resolución de servicios y dependencias.
- **Persistencia de sesiones**, evitando consultas repetitivas a la base de datos cuando no es necesario.

La justificación de estos mecanismos radica en que **reducen el número de operaciones repetidas**, disminuyendo las llamadas innecesarias a la API y a la base de datos. Esto supone un menor uso de recursos del servidor y, en consecuencia, un menor consumo energético por cada petición atendida.

En el frontend, Next.js aplica por defecto:

- **Caché del navegador** para recursos estáticos como JavaScript, CSS e imágenes.
- **Reutilización de datos en la navegación interna**, evitando recargas completas de la página y nuevas peticiones al backend cuando la información no ha cambiado.

La justificación de esta estrategia es que **evitar descargas y peticiones repetidas reduce el tráfico de red**, lo que disminuye la carga del servidor y el consumo energético del dispositivo del usuario.

7.2 Optimización de recursos

Aunque no se ha realizado una optimización manual, los frameworks utilizados incluyen **optimizaciones automáticas** orientadas a un uso eficiente de los recursos.

- **Minificación automática de CSS y JavaScript durante el proceso de build:**
 - Laravel, mediante Vite, genera archivos optimizados para producción.
 - Next.js minifica y comprime los recursos de forma automática.

La justificación de esta optimización es que **archivos más pequeños requieren menos ancho de banda y menos tiempo de procesamiento**, lo que reduce el consumo energético asociado a la transferencia y ejecución del código.

- **Carga diferida (lazy loading):**
 - Next.js implementa carga diferida de páginas y componentes, descargando solo el código necesario para cada vista.
 - Las imágenes se cargan de forma optimizada, priorizando las visibles en pantalla.

La justificación de esta decisión es que **no se cargan recursos innecesarios**, evitando trabajo adicional del navegador y del servidor, lo que contribuye a un uso más eficiente de los recursos computacionales.

7.3 Reflexión

El uso de mecanismos de caché y optimización, incluso en su configuración por defecto, supone un **ahorro energético significativo** frente a un sistema basado en consultas directas constantes.

Evitar peticiones innecesarias implica:

- Menor carga en el servidor backend.
- Menos accesos repetidos a la base de datos.
- Reducción del tráfico de red entre cliente y servidor.

La justificación de este ahorro energético se basa en que **cada operación computacional y cada transferencia de datos conlleva un consumo de energía**. Al reducir estas operaciones, se disminuye el uso de CPU, memoria y red, tanto en el servidor como en los dispositivos clientes. En conjunto, estas decisiones permiten que la aplicación sea más eficiente y sostenible desde el punto de vista medioambiental.

8. Conclusiones

8.1. Autoevaluación

- **Dificultades encontradas:** Tuvimos problemas al hacer el apartado de favoritos ya que nos saltaba error de CSRF TOKEN cada vez que lo seleccionábamos y en las reseñas tuvimos problemas al mostrar las reseñas porque nos aparecía de primeras, pero luego se borraba
- **Valoración de la metodología:** Cada uno se le asignaba una tarea que hacer y cuando se completaba la poníamos en común en un ordenador para que todo funcionase.

8.2. Líneas futuras

- **Propuestas de futuro:** Para comercializarlo pondríamos anuncios en la página web para que nos pagasen por salir y un sistema premium que puedes publicar tus propios juegos gratis y darte a conocer.

9. Bibliografía

• Laravel Documentation

Laravel LLC. (2024). *Laravel Documentation*.
Disponible en: <https://laravel.com/docs>

- **Next.js Documentation**

Vercel Inc. (2024). *Next.js Documentation*.

Disponible en: <https://nextjs.org/docs>

- **PHP Manual**

The PHP Group. (2024). *PHP Manual*.

Disponible en: <https://www.php.net/manual/es/>

- **React Documentation**

Meta Platforms, Inc. (2024). *React Documentation*.

Disponible en: <https://react.dev/>

- **diagrams.net (draw.io)**

JGraph Ltd. (2024). *diagrams.net*.

Disponible en: <https://www.diagrams.net/>