



Fase de análisis y diseño

Índice

➤ Justificación proyecto.....	3
➤ Análisis funciones.....	4
➤ Tecnologías utilizadas.....	5
➤ Base de datos.....	6
➤ Flujo navegación	9
➤ Formato logotipo.....	12
➤ Paleta colores.....	13
➤ Tipografía.....	14
➤ Imágenes.....	15
➤ Mock Up.....	16

Justificación del proyecto

El desarrollo de este proyecto surge de la pasión por los videojuegos, un pasatiempo que ha acompañado a generaciones y que se ha consolidado como una de las industrias de entretenimiento más grandes y dinámicas del mundo. La idea de crear una página web enfocada en la colección de videojuegos responde tanto a un interés personal como a una necesidad detectada en el mercado actual.

Los videojuegos no solo representan una forma de entretenimiento, sino también una herramienta de conexión social, aprendizaje y creatividad. A pesar de su popularidad, los usuarios muchas veces enfrentan dificultades para encontrar plataformas intuitivas, organizadas y con una experiencia personalizada que les permita organizar juegos de manera sencilla, eficiente y con acceso a contenido relevante.

El proyecto tiene su origen en la observación de estas carencias y en el deseo de ofrecer una solución innovadora que combine facilidad de uso con funcionalidades atractivas, como la búsqueda optimizada, filtros avanzados, colecciones personalizables y herramientas para registrar el progreso en los videojuegos. Todo esto busca crear un espacio en línea que no solo sea una página web mas, sino una comunidad para aficionados.



Análisis de funcionamiento

La aplicación permitirá organizar y gestionar colecciones de videojuegos con las siguientes funciones clave:

- **Búsqueda y filtrado:** Buscar juegos por nombre, género, plataforma... Obteniendo los resultados en tiempo real.
- **Gestión de colecciones:** Crear y personalizar listas como "Favoritos" o "Juegos completados".
- **Registro de progreso:** Actualizar el estado de un juego ("Jugando", "Terminado").
- **Recomendaciones:** Sugerencias basadas en preferencias y colecciones del usuario.

Consideraciones y limitaciones:

- Se requerirá registro o inicio de sesión previo para gestionar colecciones.
- No incluirá funciones de compra, descarga o estadísticas avanzadas de plataformas externas.



Playforge

Tecnologías utilizadas

Para el desarrollo del proyecto, se emplearán las siguientes tecnologías:

- **Frontend:** HTML y CSS, que permitirán diseñar una interfaz modular, rápida y moderna, optimizando la experiencia visual y de navegación para los usuarios.
- **Backend:** JavaScript, PHP y MySQL, herramientas clave para gestionar datos dinámicos de manera eficiente, garantizar la escalabilidad del sistema y proporcionar funciones seguras para el registro e inicio de sesión de los usuarios.

Estas tecnologías han sido seleccionadas por su flexibilidad, rapidez en el desarrollo y capacidad para ofrecer una experiencia de usuario sólida y alineada con los objetivos del proyecto.



Base de datos

La base de datos estará diseñada para gestionar usuarios, videojuegos y colecciones de manera eficiente, partiendo de un Modelo Entidad-Relación que se adaptará al modelo relacional propuesto.

•Entidades principales:

- **Usuario:** Representa a los usuarios registrados en la plataforma.
- **Videojuego:** Contiene información de los videojuegos disponibles.
- **Colección:** Representa las listas personalizadas creadas por los usuarios.

•Relaciones:

- **Usuario - Colección:** Un usuario puede tener múltiples colecciones.
- **Colección - Videojuego:** Una colección puede incluir múltiples videojuegos, y un videojuego puede estar en varias colecciones.



Videojuegos

Almacena información sobre los videojuegos disponibles.

- **id_videojuego** (PK, INT): Identificador único del videojuego.
- **titulo** (VARCHAR(150)): Nombre del videojuego.
- **genero** (VARCHAR(50)): Género principal del juego (acción, aventura).
- **plataforma** (VARCHAR(50)): Plataforma del videojuego (Nintendo, PC) .
- **año_lanzamiento** (YEAR): Año de lanzamiento del videojuego.
- **desarrollador** (VARCHAR(100)): Nombre del estudio/desarrollador.

Usuarios

Gestiona la información básica de los usuarios registrados.

- **id_usuario** (PK, INT): Identificador único del usuario.
- **nombre_usuario** (VARCHAR(50)): Nombre de usuario único.
- **email** (VARCHAR(100), UNIQUE): Correo electrónico del usuario.
- **contraseña** (VARCHAR(255)): Contraseña cifrada.
- **fecha_creacion** (DATETIME): Fecha y hora de creación de la cuenta.

Colecciones

Representa las colecciones personalizadas creadas por los usuarios.

- **id_coleccion** (PK, INT): Identificador único de la colección.
- **id_usuario** (FK, INT): Relación con el usuario que creó la colección.
- **nombre_coleccion** (VARCHAR(100)): Nombre de la colección.
- **fecha_creacion** (DATETIME): Fecha de creación de la colección.

Coleccion_Videojuego

Define la relación muchos a muchos entre colecciones y videojuegos.

- **id_coleccion** (FK, INT): Relación con una colección específica.
- **id_videojuego** (FK, INT): Relación con un videojuego específico.
- **fecha_agregado** (DATETIME): Fecha en que el videojuego fue agregado a la colección.

Relaciones entre tablas:

- **Usuarios ↔ Colecciones:** Relación 1:N (Un usuario puede tener varias colecciones).
- **Colecciones ↔ Videojuegos:** Relación N:M (Una colección puede incluir múltiples videojuegos, y un videojuego puede estar en varias colecciones).



Playforge

```

1 -- Creación de la base de datos
2 CREATE DATABASE IF NOT EXISTS gestion_videojuegos;
3 USE gestion_videojuegos;
4
5 -- Tabla Usuarios
6 CREATE TABLE Usuarios (
7     id_usuario INT AUTO_INCREMENT,
8     nombre_usuario VARCHAR(50),
9     email VARCHAR(100),
10    contraseña VARCHAR(255),
11    fecha_creacion DATETIME,
12    PRIMARY KEY (id_usuario)
13 );
14
15 -- Tabla Videojuegos
16 CREATE TABLE Videojuegos (
17     id_videojuego INT AUTO_INCREMENT,
18     titulo VARCHAR(150),
19     genero VARCHAR(50),
20     plataforma VARCHAR(50),
21     año_lanzamiento YEAR,
22     desarrollador VARCHAR(100),
23     PRIMARY KEY (id_videojuego)
24 );
25
26 -- Tabla Colecciones
27 CREATE TABLE Colecciones (
28     id_coleccion INT AUTO_INCREMENT,
29     id_usuario INT,
30     nombre_coleccion VARCHAR(100),
31     fecha_creacion DATETIME,
32     PRIMARY KEY (id_coleccion)
33 );
34
35 -- Tabla Coleccion_Videojuego
36 CREATE TABLE Coleccion_Videojuego (
37     id_coleccion INT,
38     id_videojuego INT,
39     fecha_agregado DATETIME
40 );
41

```

Sin restricciones

Con restricciones

```

1 -- Creación de la base de datos
2 CREATE DATABASE IF NOT EXISTS gestion_videojuegos;
3 USE gestion_videojuegos;
4
5 -- Tabla Usuarios
6 CREATE TABLE Usuarios (
7     id_usuario INT AUTO_INCREMENT,
8     nombre_usuario VARCHAR(50) NOT NULL UNIQUE,
9     email VARCHAR(100) NOT NULL UNIQUE,
10    contraseña VARCHAR(255) NOT NULL,
11    fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP,
12    PRIMARY KEY (id_usuario)
13 );
14
15 -- Tabla Videojuegos
16 CREATE TABLE Videojuegos (
17     id_videojuego INT AUTO_INCREMENT,
18     titulo VARCHAR(150) NOT NULL,
19     genero VARCHAR(50),
20     plataforma VARCHAR(50),
21     año_lanzamiento YEAR,
22     desarrollador VARCHAR(100),
23     PRIMARY KEY (id_videojuego)
24 );
25
26 -- Tabla Colecciones
27 CREATE TABLE Colecciones (
28     id_coleccion INT AUTO_INCREMENT,
29     id_usuario INT NOT NULL,
30     nombre_coleccion VARCHAR(100) NOT NULL,
31     fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP,
32     PRIMARY KEY (id_coleccion),
33     FOREIGN KEY (id_usuario) REFERENCES Usuarios(id_usuario) ON DELETE CASCADE
34 );
35
36 -- Tabla Coleccion_Videojuego
37 CREATE TABLE Coleccion_Videojuego (
38     id_coleccion INT NOT NULL,
39     id_videojuego INT NOT NULL,
40     fecha_agregado DATETIME DEFAULT CURRENT_TIMESTAMP,
41     PRIMARY KEY (id_coleccion, id_videojuego),
42     FOREIGN KEY (id_coleccion) REFERENCES Colecciones(id_coleccion) ON DELETE CASCADE,
43     FOREIGN KEY (id_videojuego) REFERENCES Videojuegos(id_videojuego) ON DELETE CASCADE
44 );

```


Flujo de navegación

Ruta de los Datos:

- **Entrada de datos:** Datos introducidos por el usuario en formularios (nombre de usuario, contraseña, videojuegos, colecciones).
- **Procesos internos:** El backend recibe las peticiones HTTP, las valida y procesa los datos.
- **Almacenamiento:** Datos guardados en la base de datos en las tablas correspondientes.
- **Consulta y recuperación:** Consulta a la base de datos y envía resultados al frontend para mostrarlos.
- **Actualización y eliminación:** Cambios reflejados en la base de datos.

Ficheros de Datos Generados:

- **JSON/CSV:** Exportación de colecciones de videojuegos con datos relevantes (títulos, progreso, etc.).
- **Logs:** Registro de accesos, errores y actividades (inicios de sesión, interacciones con la base de datos).

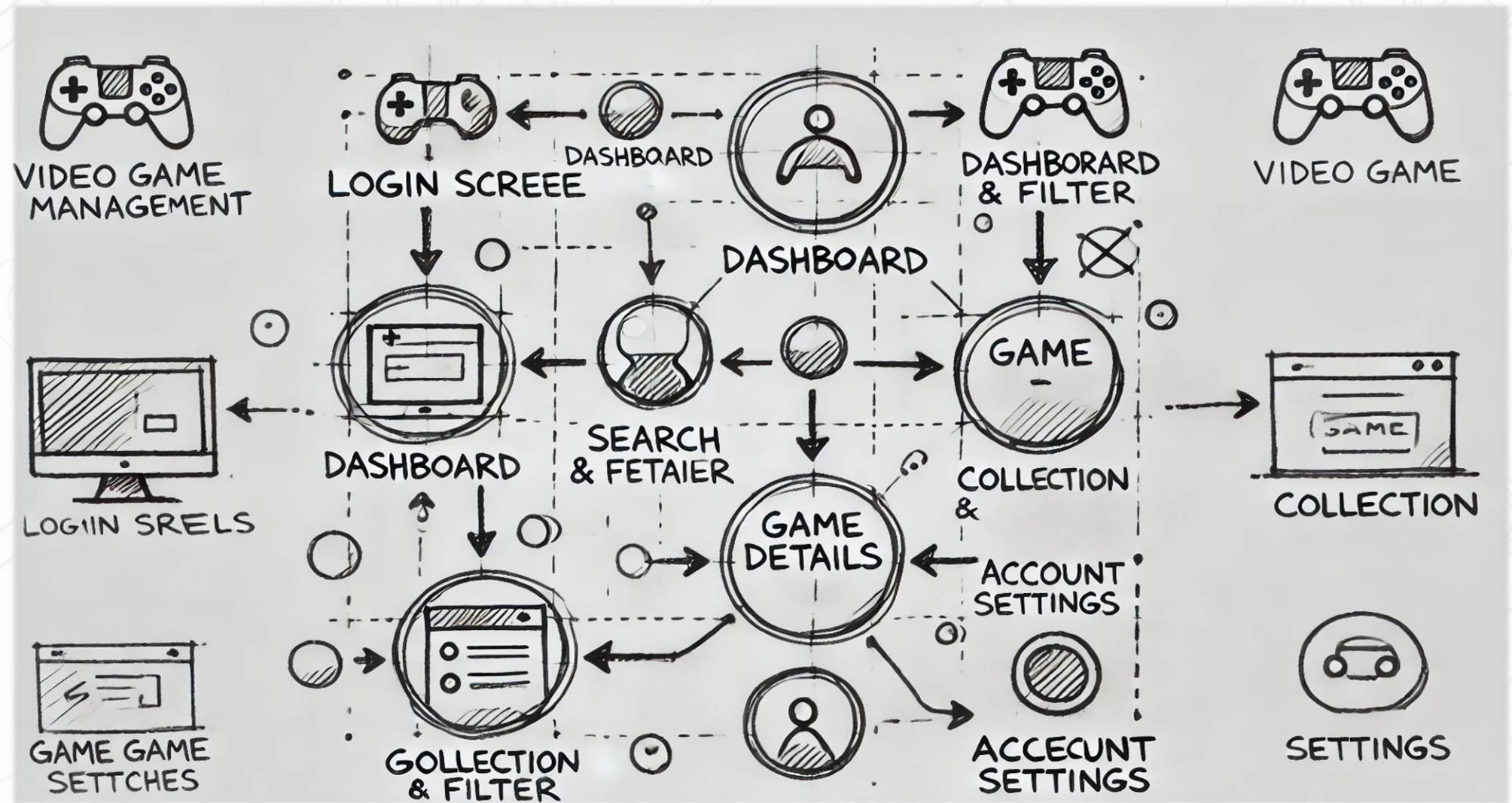
Relación con la Base de Datos:

- Los archivos generados (JSON/CSV) reflejan información de las tablas: un JSON de colección contiene videojuegos (basado en id_videojuego).
- Los logs registran eventos como la creación o modificación de colecciones.

Resumen del flujo:

- El usuario ingresa a la aplicación, se autentica (pantalla de inicio/login).
- Se carga la pantalla principal donde se muestran las colecciones.
- Desde el dashboard, el usuario puede buscar y filtrar videojuegos.
- Al seleccionar un videojuego, el usuario puede añadirlo a una colección existente o crear una nueva.
- El usuario puede consultar y gestionar el progreso de los videojuegos dentro de sus colecciones.

Este flujo de datos y las pantallas definidas permiten una interacción fluida y organizada para la gestión de videojuegos y colecciones en la aplicación.

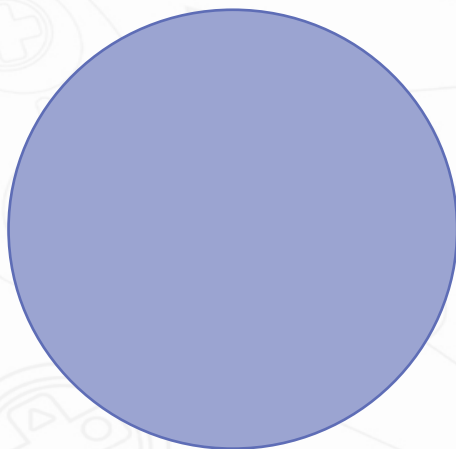


Formatos de logotipo



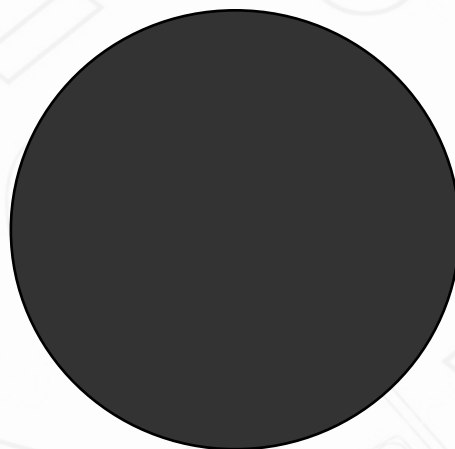
Paleta de colores

Color principal



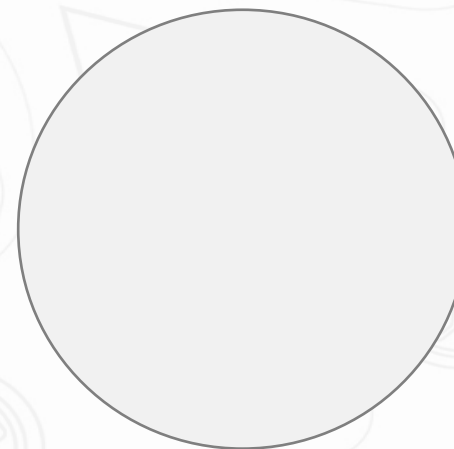
Lila azulado
#9BA4D1

Color secundario



Gris oscuro
#333333

Color secundario



Gris claro
#F1F1F1

Tipografía

CALIBRI

Texto del título

Este texto es una prueba

Este texto es una prueba

Este texto es una prueba

Este texto es una prueba



Playforge

Imágenes



Playforge

Mock Up

