

My Things:

Aplicación para gestionar colecciones

Autora: Angela Conde López

Fecha: 14/06/2021

Tutor: Santiago Domínguez Zapico



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting a hierarchical or multi-layered structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

Vista previa

Convirtiendo esto:



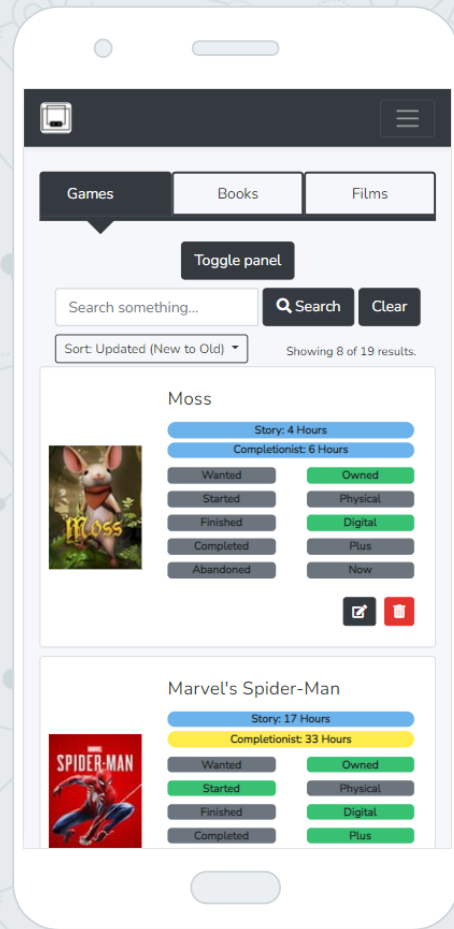
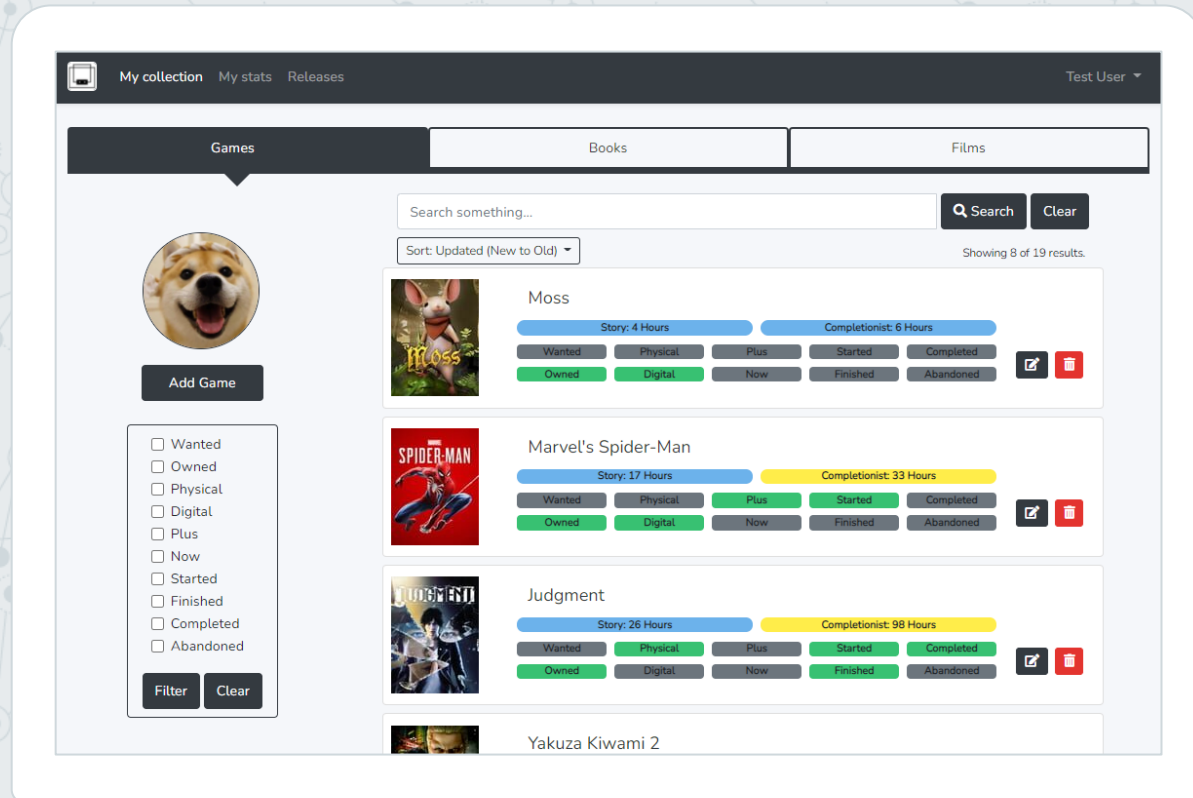
Esto:



O esto:



En esto:



Calendario con petición asíncrona:

June 2021

today



Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1 Necromunda: Hired Gun Operation: Tango Stonefly Virtua Fighter 5: Ultimate Showdown	2	3 Tour de France 2021	4 Sniper Ghost Warrior Contracts 2	5
6	7	8 Chivalry 2	9	10 Chicory: A Colorful Tale	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25 Scarlet Nexus	26
27	28	29	30			

JSON

Datos sin procesar

Cabeceras

Guardar

Copiar

Contraer todo

Expandir todo

Filtrar JSON

▼ 0:

```
event_id: 2
title: "Operation: Tango"
start: "2021-06-01"
end: "2021-06-02"
▼ url: "https://ieslamarisma.net/proyectos/2daw/angela/mythings/public/details/2"
```

▼ 1:

```
event_id: 4
title: "Necromunda: Hired Gun"
start: "2021-06-01"
end: "2021-06-02"
▼ url: "https://ieslamarisma.net/proyectos/2daw/angela/mythings/public/details/4"
```

▼ 2:

```
event_id: 6
title: "Stonefly"
start: "2021-06-01"
end: "2021-06-02"
▼ url: "https://ieslamarisma.net/proyectos/2daw/angela/mythings/public/details/6"
```

Enlaces con Geolocalización mediante IP:



My Things



Test User 1's wishlist

Games

Books

Films

Showing 12 of 32 results.



Horizon Forbidden West

Buy now on
amazon >



It Takes Two

Buy now on
amazon >



Judgment

Buy now on
amazon >



Kentucky Route Zero: TV





Maquette




Mass Effect Legendary

Formulario de contacto con HoneyPot:

 Contact us




Name



your@email.com

Your message



Submit

```
▼ <div class="form-group d-none">
  ▼ <div class="input-group mb-2"> flex
    <input id="trap" class="form-control "
      type="checkbox" name="trap" hidden="">
    </div>
  </div>
```

We have received a message

My Things <mythingscollectionapp@gmail.com>
para mí ▼

These are the details:

Name: Pepe

Email: dm@ksfms.com

Message: Hola, estoy enviando un mensaje.

Your message has been received Recibidos x

My Things <mythingscollectionapp@gmail.com>
para mí ▼

We will answer as soon as possible.

If you didn't request this email, someone used your email address at My Things.

You can let us know or you can simply ignore this message. Thank you.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some solid and some hollow, connected by thin lines. The overall structure is dense and organic, resembling a molecular or biological network.

1.

Introducción

Descripción del proyecto.

Desarrollo de una aplicación web que permite al usuario añadir artículos a una colección guardada en una base de datos y gestionar dicha colección:

- ⦿ El usuario puede añadir artículos, modificar su información sobre posesión y uso del artículo o eliminar el artículo de su colección.
- ⦿ El usuario podrá ver una lista detallada de sus artículos con información sobre estos. Esta lista podrá ser filtrada y ordenada por varios parámetros, así como ejecutar una búsqueda sobre ella.

Descripción del proyecto.

La información de los artículos se obtiene de APIs y/o sitios webs proporcionados por terceros.

Para la demostración práctica de esta aplicación se ha implementado únicamente la colección de videojuegos de Playstation 4.

Cualquier otra colección implementada en un futuro seguirá exactamente el mismo patrón que ésta y únicamente será necesario cambiar la API de la cual se extrae la información y la consulta a realizar sobre ella.



Objetivos generales.

- ◎ Aplicar y ampliar los conocimientos adquiridos durante el ciclo formativo.
- ◎ Aprender a consumir datos proporcionados por servicios de terceros.

Motivaciones personales.

- © Trabajar con APIs para ampliar mis conocimientos sobre ellas.

¿Por qué APIs?

- ⦿ Están cobrando más y más fuerza en las grandes empresas, sobre todo en el sector de la banca tras la normativa europea obligatoria de servicios de pago (PSD2).
- ⦿ Son una herramienta muy eficiente para el transporte de datos en un mercado en el que el auge del big data está incrementando el valor de los datos de forma exponencial.
- ⦿ Estos factores convierten a las APIs en una gran fuente de ingresos para las empresas.

¿En qué centrar el proyecto?

- ◎ La mayoría de buenas APIs gratuitas que pude encontrar proporcionan datos sobre productos de ocio, entretenimiento y cultura.
- ◎ Esto me llevó a decidir desarrollar una biblioteca de productos de ocio para uso personal.

¿Existe ya?

Sí, existen varios servicios webs para catalogar tus colecciones.

Inconveniente:

- ⦿ Están orientados al aspecto social (compartir, comparar, comentar).

¿Hay alternativas para uso privado?

Sí, existe:

- ⦿ Software instalable en un ordenador.
- ⦿ Apps para dispositivos móviles.

Inconvenientes:

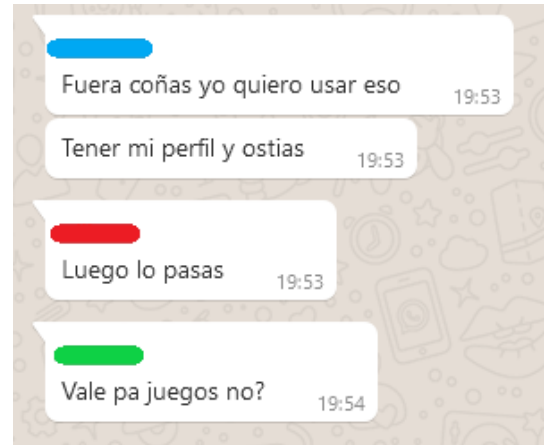
- ⦿ No son multiplataforma.
 - ⦿ No se puede acceder sin instalar.
 - ⦿ La información se guarda en un fichero local con riesgo de pérdida.
 - ⦿ El dispositivo tiene que ser compatible.
- Posible incompatibilidad en futuras actualizaciones del software o el dispositivo.

Alternativas empleadas por los usuarios:

- ◎ Hojas de cálculo en servicios SaaS
(por ejemplo Google Docs).

¿Interesa el proyecto?

Los usuarios han reaccionado positivamente ante la idea de mi proyecto:



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some solid and some hollow, connected by thin lines. The overall structure is a dense, branching network.

2.

Estudio de viabilidad

Requisitos del sistema.

- ⦿ La aplicación consume servicios externos proporcionados por terceros para obtener la información de los artículos.
- ⦿ El uso de la aplicación requiere que el usuario se registre en el sistema.
- ⦿ La aplicación está enfocada a usuarios que posean al menos un manejo básico de aplicaciones.

Requisitos del sistema

- ⦿ La interfaz debe ser diseñada para un usuario con conocimientos básicos de aplicaciones web, de forma intuitiva y fácil de usar.
- ⦿ La interfaz debe ser responsive para facilitar su uso en distintas plataformas.

Requisitos del sistema

- ⦿ La aplicación debe permitir el registro y autenticación de usuarios.
- ⦿ La aplicación debe permitir al usuario modificar sus datos personales o eliminar su cuenta.

Requisitos del sistema

- ⦿ El sistema debe poder recibir la información requerida de la fuente externa y almacenarla en la base de datos propia.
- ⦿ La aplicación debe poder extraer los datos almacenados en la base de datos propia y enviarlos a la vista.

Requisitos del sistema

- ◎ La aplicación debe permitir al usuario:
 - Añadir nuevos objetos a su colección
 - Modificarlos
 - Eliminarlos

Requisitos del sistema

- ⦿ La aplicación debe permitir filtrar los objetos visibles por distintos parámetros.
- ⦿ La aplicación debe permitir ordenar la lista de objetos visibles por distintos parámetros.
- ⦿ La aplicación debe permitir ejecutar una búsqueda sobre la lista de objetos visibles.

Catalogación y priorización de los requisitos

Utilizando la técnica de priorización de requisitos MoSCoW se han catalogado en:

- ◎ M -> “Must have” (Imprescindible)
- ◎ S -> “Should have” (Importante)
- ◎ C -> “Could have” (Posible)
- ◎ W -> “Would like but won’t get” (Me gustaría pero no)

Catalogación y priorización de los requisitos

Must have:

- ⦿ Registro y autenticación de usuarios.
- ⦿ Recibir información de fuente externa.
- ⦿ Almacenar la información externa en la base de datos propia.
- ⦿ Añadir objetos nuevos a la colección del usuario.

Catalogación y priorización de los requisitos

Should have:

- ⦿ Modificar o eliminar la cuenta de usuario.
- ⦿ Editar los objetos añadidos.
- ⦿ Eliminar los objetos añadidos.
- ⦿ Filtrar los objetos visibles.
- ⦿ Interfaz responsive.

Catalogación y priorización de los requisitos

Could have:

- ⦿ Registro y autenticación mediante OAuth.
- ⦿ Ver estadísticas propias.
- ⦿ Ver calendario de lanzamientos próximos.
- ⦿ Permitir al usuario compartir toda o partes de su colección mediante una sección accesible de forma pública.
- ⦿ Añadir enlaces monetizados para la compra de los objetos marcados como deseados.

Catalogación y priorización de los requisitos

Would like but won't get:

- ⦿ Integración de servicios de terceros, como la API de afiliados de Amazon, para obtener el precio actual del objeto de la colección.

Restricciones

- ⦿ Para acceder a la aplicación será necesario disponer de un dispositivo con acceso a Internet, ya sea un ordenador o un teléfono móvil.
- ⦿ Para poder hacer uso de la aplicación será necesario registrarse y/o identificarse.
- ⦿ La aplicación tendrá que ser accesible desde los navegadores más utilizados, como Mozilla Firefox, Google Chrome o Microsoft Edge.

Restricciones

- ⦿ Las contraseñas de los usuarios deberán almacenarse de forma encriptada.
- ⦿ Las modificaciones sobre el perfil del usuario o los objetos de su colección solamente deberán poderse realizar por el dueño de los mismos debidamente autenticado.

Evaluación de riesgos

- ⦿ El mayor riesgo es el cese de prestación del servicio de la API externa, que inutilizaría la funcionalidad de añadir nuevos objetos que no se encuentren ya en la base de datos propia.

La posibilidad de ocurrir esto es muy baja, ya que se ha seleccionado una API profesional proporcionada por una de las mayores empresas de su sector.

Plan de contingencia

- ⦿ Ante el posible cese de prestación del servicio de la API externa se ha seleccionado otra API externa de características similares y con muchos años de servicio, que puede proporcionar los datos necesarios.
- ⦿ El cambio se llevaría a cabo de forma rápida y sencilla y únicamente quedaría inutilizada de forma temporal la funcionalidad de añadir nuevos objetos.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting a hierarchical or central structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

3. **Análisis**

Diagrama de casos de uso

☉ Usuario no registrado:



Diagrama de casos de uso

© Usuario registrado no identificado:

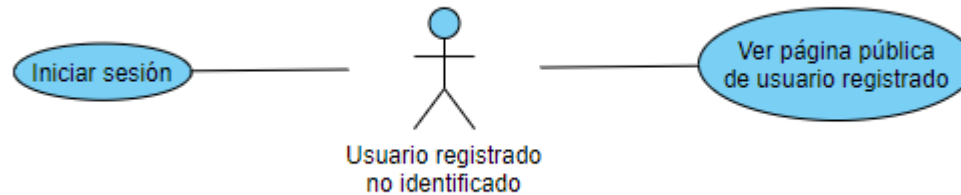
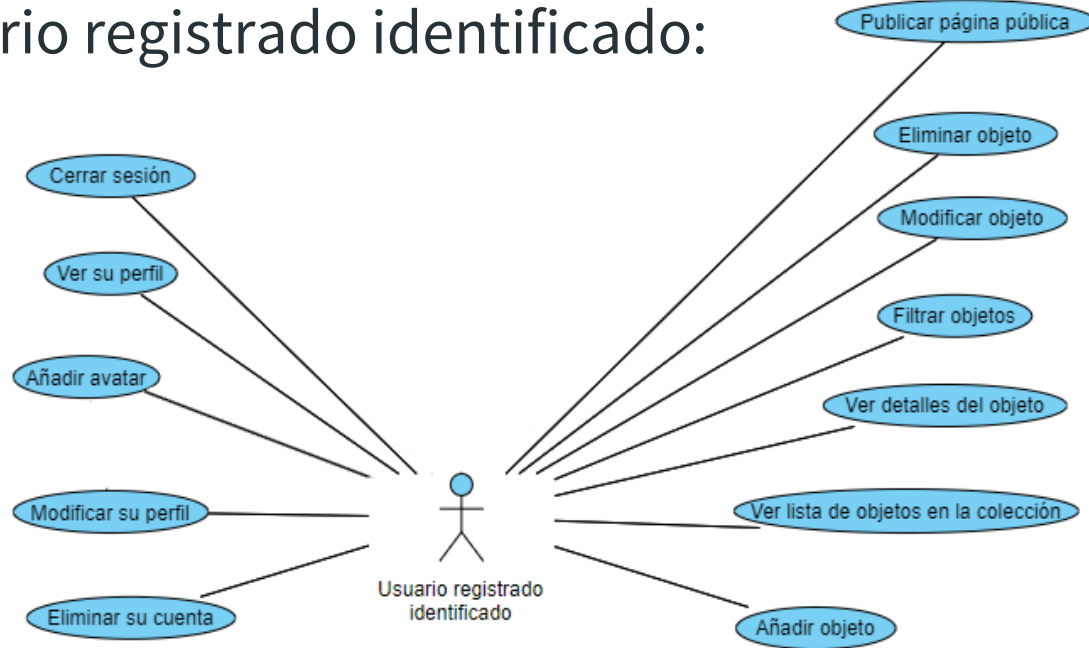


Diagrama de casos de uso

⦿ Usuario registrado identificado:



Caso de Uso como ejemplo

Caso de uso	Añadir objeto desde la API
Descripción	Permite al usuario añadir un objeto a su colección.
Actores	Usuario registrado.
Precondiciones	Iniciar sesión.
Flujo normal	<ol style="list-style-type: none">1. El usuario rellena y envía el formulario de añadir un juego con el título del juego y su información de uso.2. Se valida el formulario en el cliente.3. Se valida el formulario en el servidor.
Postcondiciones	<ol style="list-style-type: none">1. Se solicita el juego a la API.2. Se crea objeto Game con la información recibida de la API.3. Se parsea la web HowLongToBeat para extraer la información de duración.4. Se convierte la información de duración extraída a minutos.5. Se añade la información de duración al objeto Game.6. Se guarda el objeto Game en la base de datos.7. Se guardan las imágenes en las carpetas de imágenes.8. Se crea objeto UserGame con los datos de uso recibidos en el formulario.9. Se devuelve mensaje de éxito.
Excepciones	<p>Si el formulario tiene error de validación se informa al usuario.</p> <p>Si el juego no se encuentra en la API se informa al usuario.</p>

Diagrama de clases

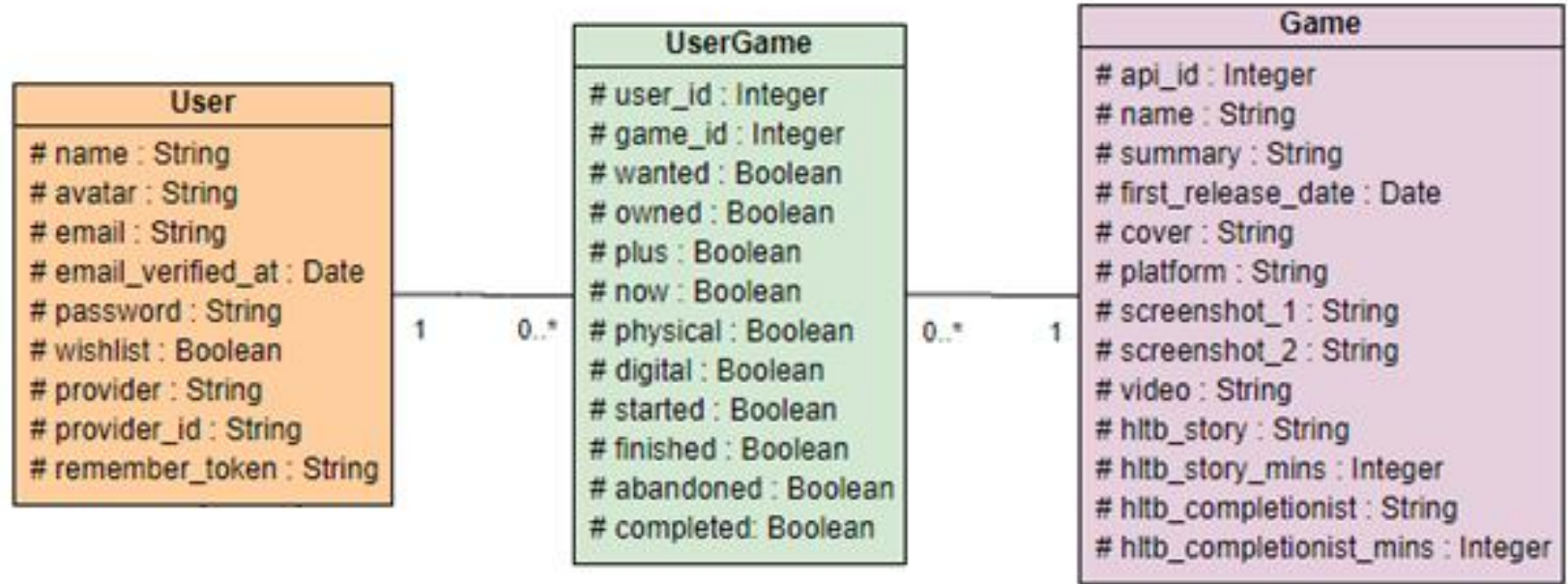
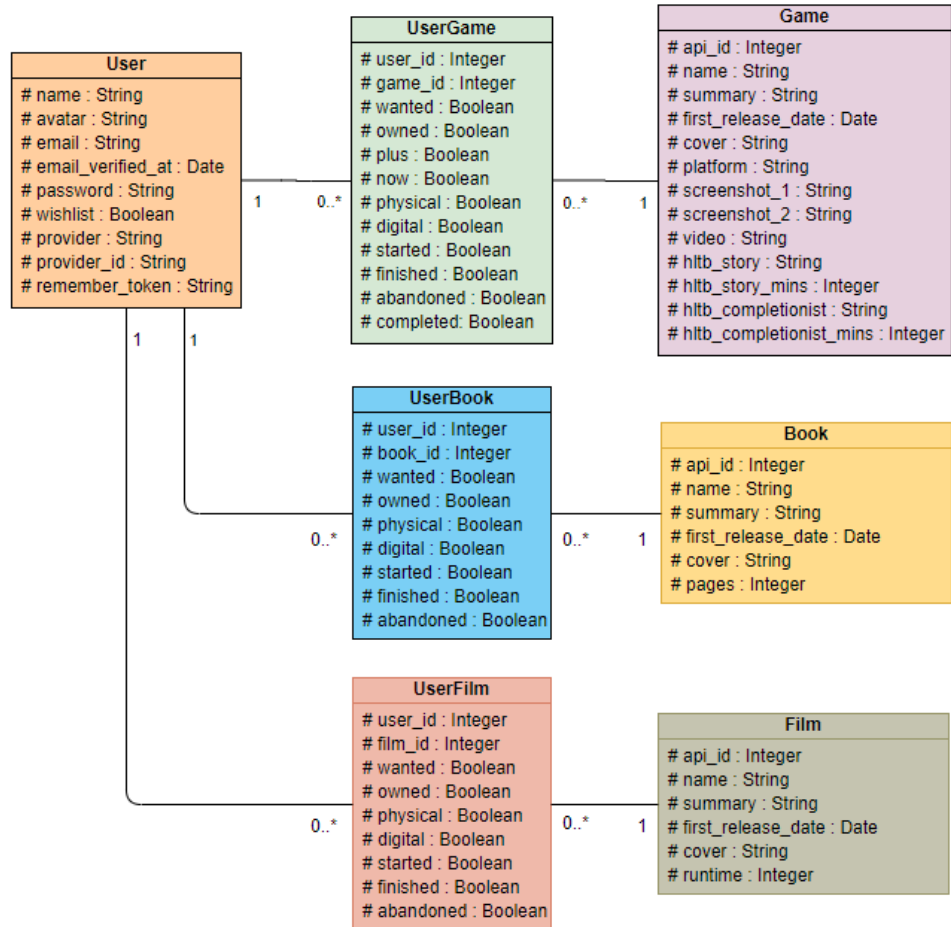
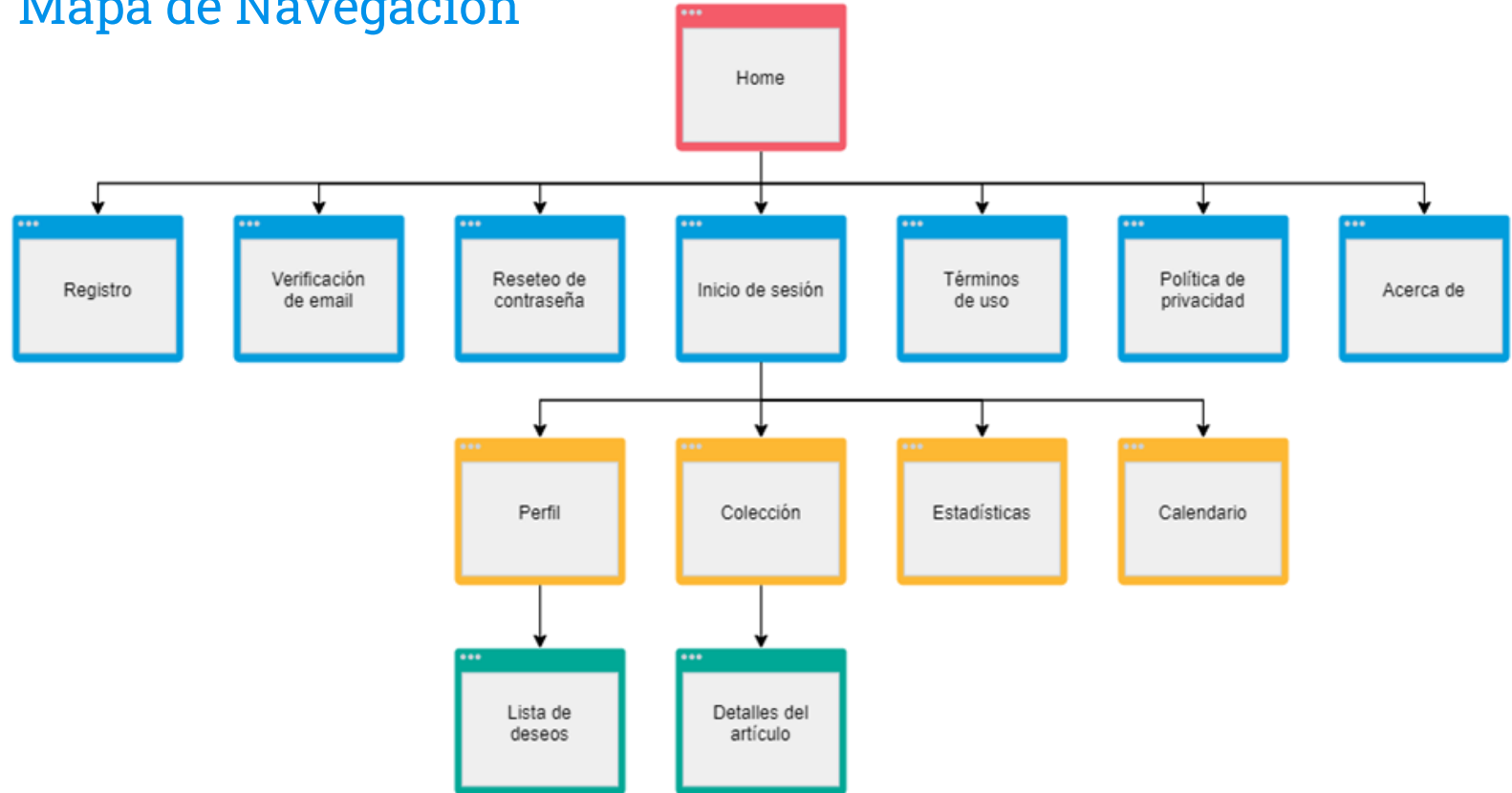


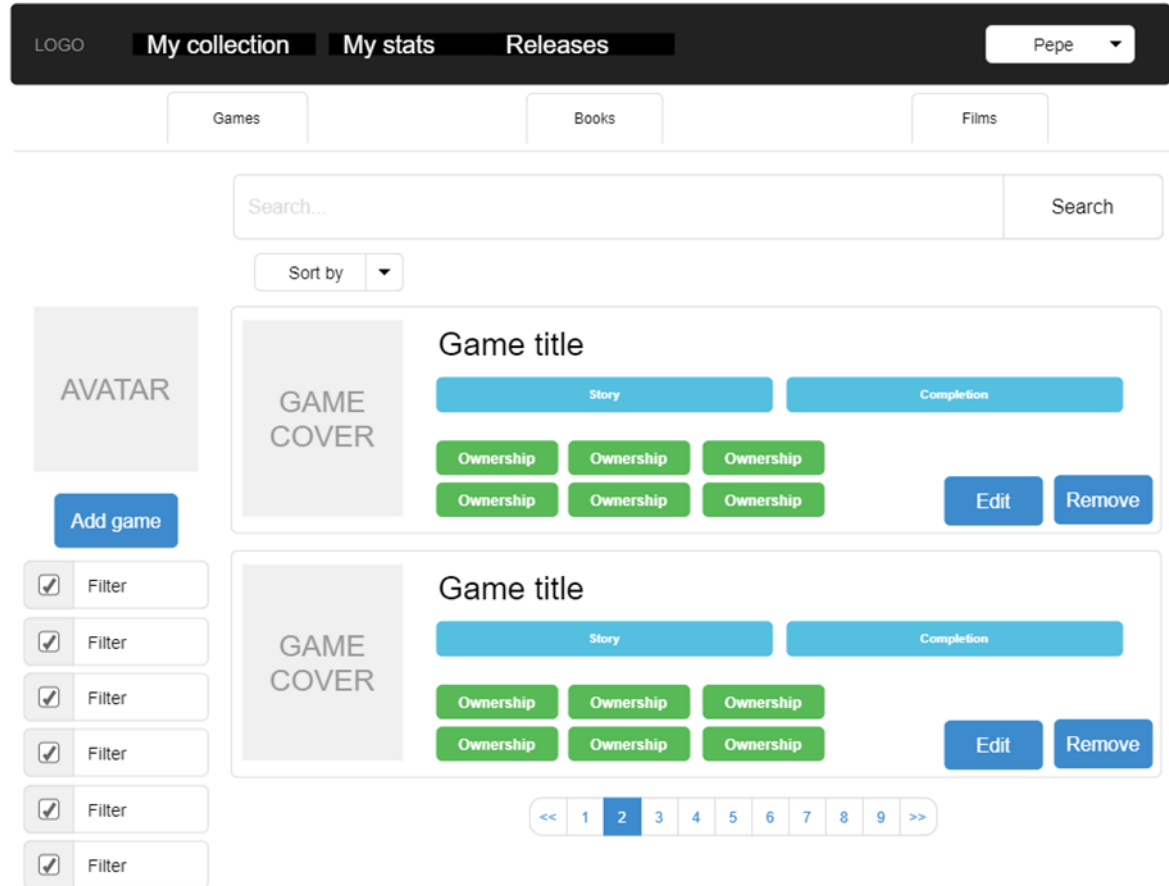
Diagrama de clases



Mapa de Navegación




Mockups



Mockups

LOGO



Games

Books

Films

Search...

Search

Sort by ▼

GAME COVER

Game title

Story

Completion

Ownership

Ownership

Ownership

Ownership

Ownership

Edit

Remove

GAME COVER

Game title

Story

Completion

Ownership

Ownership

Ownership

Ownership

Ownership

Edit

Remove

<< 1 2 3 4 5 6 7 8 9 >>

UX – Navegación por la interfaz

- ◎ Con la finalidad de ofrecer al usuario facilidad de uso de la aplicación, se ha diseñado con una interfaz simple, minimalista e intuitiva.
- ◎ Se hace uso de una barra de navegación para acceder a las distintas secciones de la aplicación.
- ◎ En la sección de la colección, que consta de varios apartados para cada tipo de coleccionable, se hace uso de pestañas para separar y facilitar la navegación entre tipos.

UX – Paleta de colores

- ◎ En cuanto a la paleta de colores se ha optado por un diseño minimalista bicolor en blanco y gris marengo para la interfaz en general, sobre la que destacan los tonos coloridos empleados para mostrar la información de los artículos de la colección del usuario o sus estadísticas.

UX – Orden de los botones

- ◎ Otro aspecto importante es el orden de los botones de Aceptar y Cancelar en modales y formularios.

UX – Orden de los botones

- © Hay distintas pautas a seguir según distintas compañías. Así, Microsoft indica que se debe colocar en primer lugar el botón de Aceptar y en segundo lugar el de Cancelar. Por su parte, Apple y Google indican que se debe colocar el botón que inicia una acción a la derecha del todo y el que la cancela justo a su izquierda.

UX – Orden de los botones

- © Al ser una aplicación diseñada para su uso en distintas plataformas y teniendo en cuenta el rápido crecimiento de usuarios que utilizan como plataforma principal un móvil Android o iPhone, se ha optado por seguir las instrucciones de Apple y Google.

UX – Colores de los botones

- ◎ Se ha optado por utilizar el color gris marengo principal de la interfaz para los botones en general.
- ◎ A los botones que inician o confirman una acción destructiva se les ha dado el color rojo como señal visual de peligro.
- ◎ A los botones que simplemente cierran un modal se les ha dado color gris claro para restarles importancia respecto al botón principal.



Accesibilidad

Para facilitar la accesibilidad a usuarios con discapacidades visuales, motoras o cognitivas se han tenido en cuenta los factores que afectan a la accesibilidad de la aplicación web, entre ellos:



Accesibilidad – Compatibilidad con el teclado

- ⦿ La aplicación debe ser completamente navegable con el teclado. Se debe permitir saltar entre secciones o elementos, marcar, pulsar...
- ⦿ Además debe hacerse uso de indicadores de posición que permitan identificar el elemento en el que se encuentra el teclado.
- ⦿ También se debe respetar un orden lógico de navegación, por ejemplo de arriba abajo y de izquierda a derecha.

Accesibilidad – Puntos de referencia ARIA

- ⦿ Estas son etiquetas que se agregan al contenido para definirlo claramente en la página, esto permite a los lectores de pantalla y dispositivos similares comprender el contenido y es especialmente útil con contenido dinámico.
- ⦿ ARIA también es útil para hacer que la navegación sea más sencilla, ya que permite a los usuarios saltar directamente a contenido específico.



Accesibilidad – Texto alternativo para las imágenes

- © Esto permite a los lectores de pantalla "leer" la imagen.

Accesibilidad – Contraste de colores

- ◎ Se deben utilizar colores para los elementos que contrasten con el fondo, evitando crear una paleta de colores donde los tonos sean demasiado similares en matiz y saturación.

Accesibilidad – Redimensionamiento

- ◎ La mayoría de los dispositivos y navegadores permiten a los usuarios cambiar el tamaño del texto, lo que puede ser útil para las personas con discapacidad visual.
- ◎ La aplicación web debe estar diseñada de forma que aumentar el nivel de zoom no dañe las funcionalidades.

Accesibilidad – Evitar reproducción automática

- ⦿ Se debe evitar incluir elementos multimedia que comiencen sin que el usuario los solicite primero.
- ⦿ También es mejor evitar la navegación automática, como carruseles y controles deslizantes.

Accesibilidad – Evitar CAPTCHAs

- ◎ Los captchas visuales suponen un gran problema para las personas invidentes, con baja visión o con una discapacidad de aprendizaje como la dislexia.
- ◎ Por eso es preferible emplear sistemas como preguntas de lógica, textuales o que no requieran interacción por parte del usuario (como el método “Honeypot” empleado en este proyecto, con un checkbox oculto para atrapar bots).

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting a hierarchical or central structure. The lines are thin and gray, connecting the nodes in a non-linear fashion.

4. Diseño

Entorno de desarrollo

El entorno de desarrollo está compuesto por:

- ⦿ Windows
- ⦿ Apache
- ⦿ MariaDB
- ⦿ PHP

Ha sido elegido por su extendido uso y soporte y en especial por su alta compatibilidad a la hora de ser desplegado.

API externa

Como API externa a la que solicitar los datos de videojuegos se ha seleccionado la proporcionada por IGDB (<https://api-docs.igdb.com>).

Limitaciones:

- ⦿ 4 peticiones por segundo.
- ⦿ Uso no comercial.
- ⦿ Requiere cuenta de desarrollador de Twitch.

API externa para plan de contingencia

Como API externa a la que solicitar los datos de videojuegos en caso de que la principal cese su actividad se ha seleccionado la proporcionada por Giant Bomb (<https://www.giantbomb.com/api/documentation>).

Limitaciones:

- ◎ 200 peticiones por recurso por hora.
- ◎ Uso no comercial.

Link a su web.

Petición a la API de IGDB

IGDB utiliza una petición POST con consultas avanzadas en el cuerpo de la petición, lo que:

- ⦿ Permite obtener toda la información necesaria con una sola petición.
- ⦿ Ofrece muchas más facilidades a la hora de escalar la aplicación en el futuro y requerir solicitar información más compleja.

Petición a la API de IGDB

- ◎ Se realiza mediante POST.
- ◎ En la cabecera hay que indicar la ID de cliente y el Token de autorización proporcionados por el servicio de desarrolladores de Twitch.
- ◎ En el cuerpo de la petición hay que escribir la consulta a realizar, en lenguaje [Apicalypse](#).
- ◎ Además, se pueden realizar subconsultas desde esa misma petición en vez de realizar consultas sucesivas.

Ejemplo de petición a la API de IGDB

Para buscar la ID de un juego de PS4 por su título:

- Endpoint: <https://api.igdb.com/v4/games>

- Cuerpo:

 - fields id;

 - search "{TITULO}";

 - where platforms = (48);

 - limit 1;

Ejemplo de petición a la API de IGDB

Si queremos además añadir información sobre el juego:

◎ Cuerpo:

```
fields id, cover, first_release_date, name,  
platforms, screenshots, summary, videos;  
search "{TITULO}";  
where platforms = (48);  
limit 1;
```

Ejemplo de petición a la API de IGDB

Si queremos además recibir las imágenes y vídeos:

◎ Cuerpo:

```
fields id, cover.image_id, first_release_date, name,  
platforms, screenshots.image_id, summary,  
videos.video_id;
```

```
search "{TITULO}";
```

```
where platforms = (48);
```

```
limit 1;
```

Respuesta de la petición a la API de IGDB

```
{
  "id": 7334,
  "cover": {
    "id": 82054,
    "image_id": "co1rba"
  },
  "first_release_date": 1427155200,
  "name": "{TITULO}",
  "platforms": [
    48
  ],
  "screenshots": [
    {
      "id": 9530,
      "image_id": "adssvwsfbaxcrucjolv9"
    },
    {
      "id": 9531,
      "image_id": "bklmdxphzoflbgtushg0"
    }
  ],
  "summary": "{RESUMEN DE LA HISTORIA DEL JUEGO}",
  "videos": [
    {
      "id": 2830,
      "video_id": "rMxEpPCSH5A"
    }
  ]
}
```

Respuesta de la petición a la API de IGDB - Rutas

- Las imágenes van en la ruta:

https://images.igdb.com/igdb/image/upload/t_{tamaño}/{image_id}.jpg

- Donde el tamaño es uno de estos (se pueden escalar al doble añadiendo _2x):

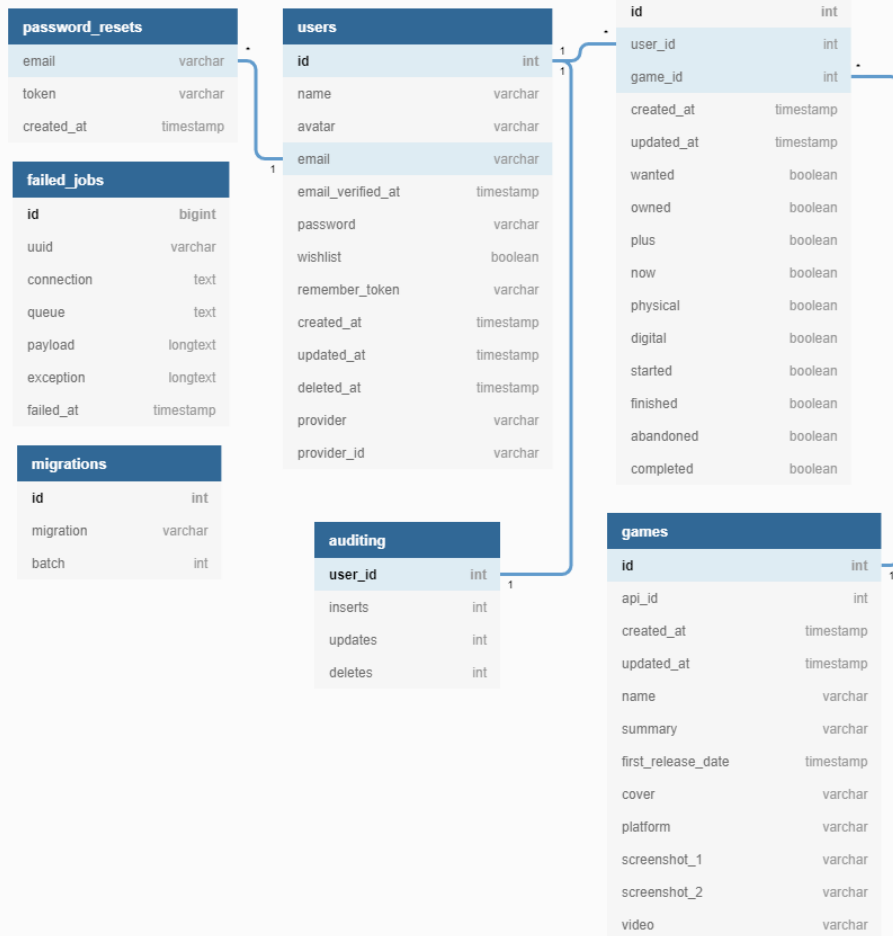
cover_small	90 x 128	Fit
screenshot_med	569 x 320	Lfill, Center gravity
cover_big	264 x 374	Fit
logo_med	284 x 160	Fit
screenshot_big	889 x 500	Lfill, Center gravity
screenshot_huge	1280 x 720	Lfill, Center gravity
thumb	90 x 90	Thumb, Center gravity
micro	35 x 35	Thumb, Center gravity
720p	1280 x 720	Fit, Center gravity
1080p	1920 x 1080	Fit, Center gravity

Respuesta de la petición a la API de IGDB - Rutas

© Los vídeos van en la ruta:

https://www.youtube.com/watch?v={video_id}

Estructura de la Base de Datos



Estructura de la Base de Datos: Migraciones

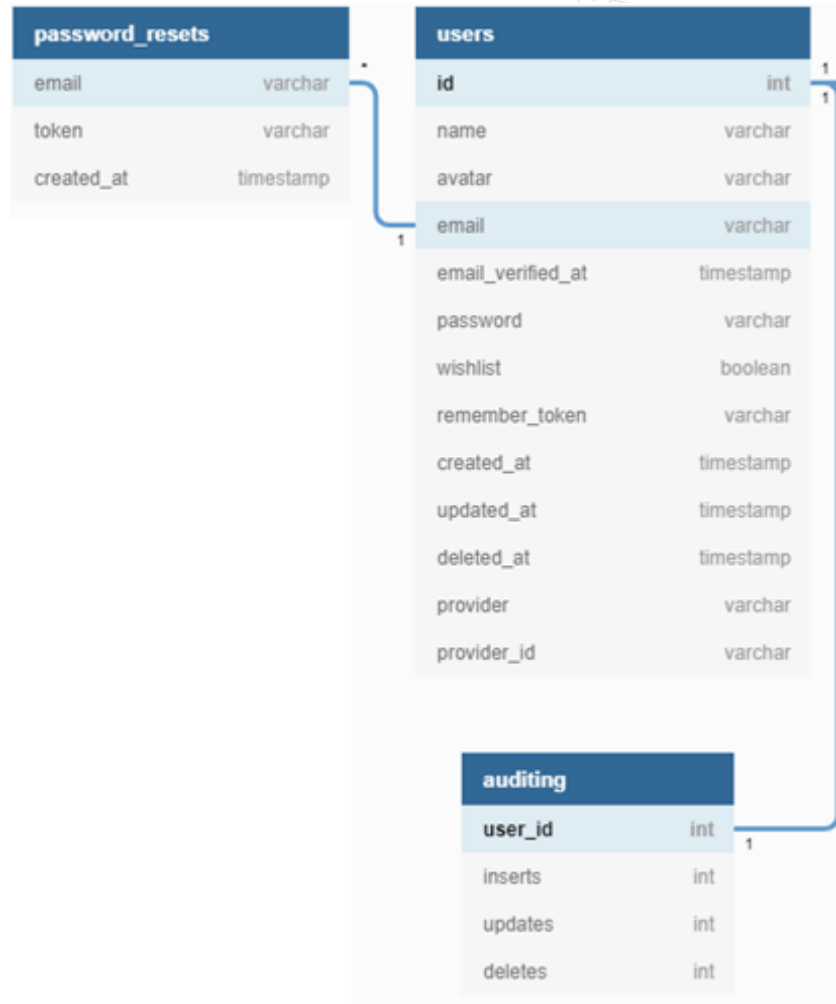
failed_jobs

id	bigint
uuid	varchar
connection	text
queue	text
payload	longtext
exception	longtext
failed_at	timestamp

migrations

id	int
migration	varchar
batch	int

Estructura de la Base de Datos: Usuarios



Estructura de la Base de Datos: Coleccionables

games	
id	int
api_id	int
created_at	timestamp
updated_at	timestamp
name	varchar
summary	varchar
first_release_date	timestamp
cover	varchar
platform	varchar
screenshot_1	varchar
screenshot_2	varchar
video	varchar

Estructura de la Base de Datos: Colección del Usuario

user_games	
id	int
user_id	int
game_id	int
created_at	timestamp
updated_at	timestamp
wanted	boolean
owned	boolean
plus	boolean
now	boolean
physical	boolean
digital	boolean
started	boolean
finished	boolean
abandoned	boolean
completed	boolean

Disparadores para auditoría

Para llevar un control de la actividad de los usuarios se utiliza una serie de disparadores de inserto, modificación y eliminación para incrementar el total de operaciones de cada tipo cuando el usuario las realiza.

Se ejecutan cada vez que el usuario:

Disparadores para auditoría

© Se registra:

```
CREATE TRIGGER `add_new_audit_user`  
AFTER INSERT  
ON `users`  
FOR EACH ROW BEGIN  
  INSERT INTO auditing  
  VALUES (null, NEW.id, 0, 0, 0);  
END
```


Disparadores para auditoría

© Añade un juego:

```
CREATE TRIGGER `increment_insert_ug`  
AFTER INSERT  
ON `user_games`  
FOR EACH ROW BEGIN  
  UPDATE auditing  
    SET auditing.inserts = auditing.inserts+1  
    WHERE auditing.user_id = NEW.user_id;  
END
```

Disparadores para auditoría

© Modifica un juego:

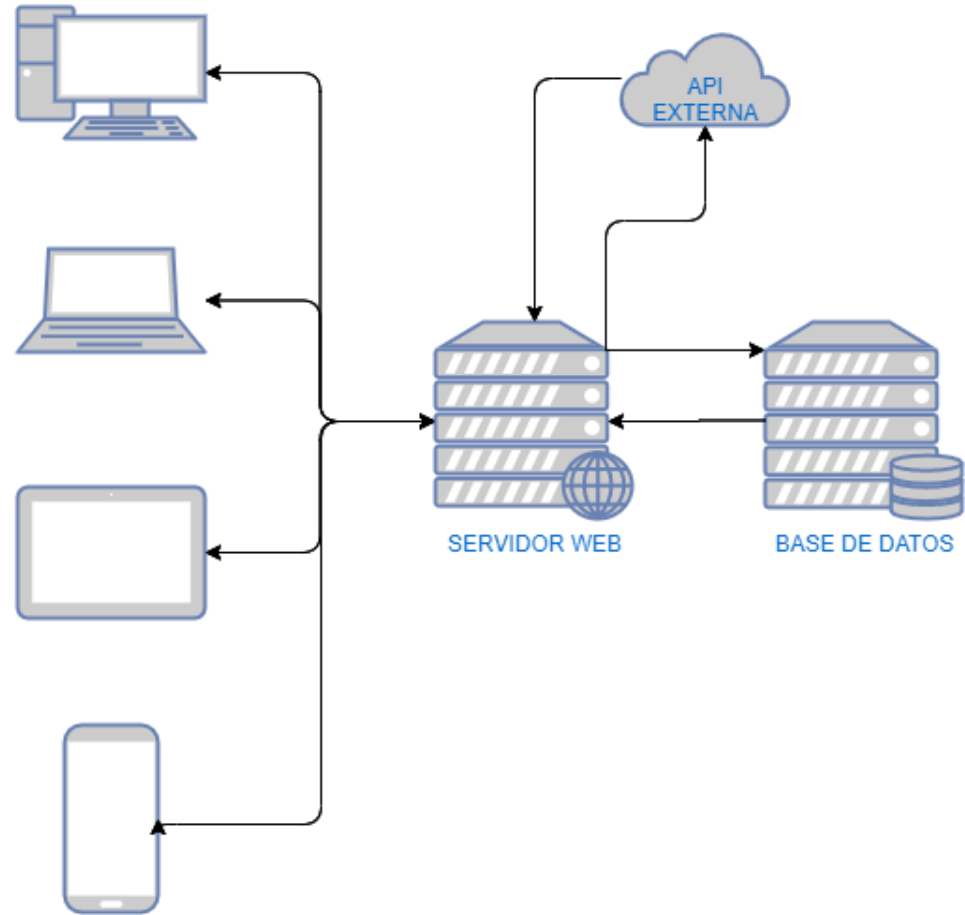
```
CREATE TRIGGER `increment_update_ug`  
AFTER UPDATE ON `user_games`  
FOR EACH ROW BEGIN  
  UPDATE auditing  
    SET auditing.updates = auditing.updates+1  
    WHERE auditing.user_id = NEW.user_id;  
END
```

Disparadores para auditoría

© Elimina un juego:

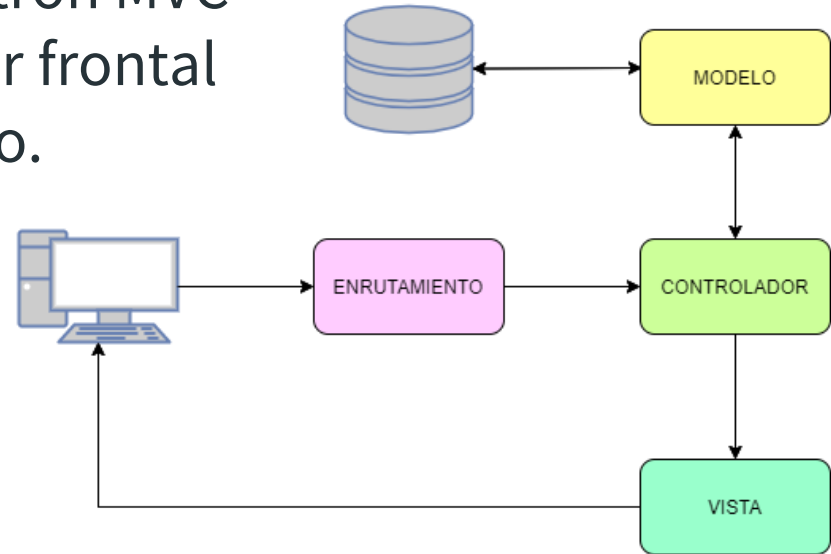
```
CREATE TRIGGER `increment_delete_ug`  
AFTER DELETE ON `user_games`  
FOR EACH ROW BEGIN  
  UPDATE auditing  
    SET auditing.deletes = auditing.deletes+1  
    WHERE auditing.user_id = OLD.user_id;  
END
```

Arquitectura física



Arquitectura de software

La aplicación está desarrollada siguiendo los principios básicos del patrón MVC e implementa un controlador frontal que gestiona el enrutamiento.



Capas

- ◎ Capa de persistencia.
- ◎ Capa de presentación.
- ◎ Capa de lógica del negocio.

Capa de persistencia

En esta capa se representan y almacenan los datos de la aplicación y permite realizar las operaciones de creación, lectura, actualización y borrado (CRUD).

- ⦿ Modelo.

Capa de persistencia - Modelo

Es el componente que representa la información en la base de datos mediante un conjunto de clases para ser gestionada por la aplicación.

Para la gestión de la información de la base de datos y la representación de esta mediante clases la aplicación utiliza el ORM (Mapeo Objeto-Relacional) Eloquent que integra el framework Laravel para hacer un mapeo de datos relacional.

Capa de presentación

Esta capa es la que permite al usuario interactuar con el sistema, se encarga de recoger la información del usuario y comunicarse con la capa de lógica.

Una vez la capa de lógica ha procesado la información esta capa se encarga de generar la presentación mediante vistas.

◎ Vista.

Capa de presentación - Vista

La vista se encarga de presentar toda la información del modelo que proporciona el controlador.

La aplicación hace uso del motor de plantillas Blade integrado en el framework Laravel para generar las diferentes vistas. Esto aporta a la aplicación una gran modularidad gracias a las secciones y a la herencia de plantillas.

Blade además permite el uso de estructuras de control, permitiendo que el código sea limpio y conciso.



Capa de lógica

Esta capa se encarga de procesar las interacciones del usuario con la aplicación, conectando la capa de presentación con la capa de persistencia.

- ⦿ Controladores.
- ⦿ Enrutamiento.

Capa de lógica - Controladores

El controlador es el que se encarga de responder a las acciones que se solicitan en la aplicación.

Es además el encargado de solicitar las operaciones de creación, lectura, actualización y borrado (CRUD) de los datos de la aplicación a la capa de persistencia.

Capa de lógica - Enrutamiento

El enrutamiento implementado mediante Laravel permite determinar cuál es la acción que se ejecuta ante una petición HTTP del usuario de la aplicación y solicitarla al controlador adecuado.

Además ofrece herramientas para trabajar con funcionalidades como redireccionamiento, parámetros, restricciones, middleware o espacios de nombre.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some solid and some hollow, connected by thin lines. The overall structure is a dense, branching network.

5.

Implementación

Tecnologías

Para esta aplicación se ha optado por tecnologías:

- ◎ De uso muy extendido.
- ◎ Que a pesar de haber sido utilizadas durante años o incluso décadas se mantienen actualizadas y permiten desarrollar con un enfoque actual.

Tecnologías - Backend

◎ PHP 7.4.11:

Lenguaje de código abierto especialmente adecuado para el desarrollo web.

◎ LARAVEL 8.22:

Framework para PHP ampliamente utilizado cuya filosofía es desarrollar código de forma elegante y simple basado en un modelo MVC.

◎ MARIADB 10.4.14:

Sistema de gestión de bases de datos derivado de MySQL, uno de los más populares para el desarrollo web.

◎ BLADE:

Potente motor de plantillas integrado en el Framework Laravel.

Tecnologías - Frontend

◎ HTML 5:

Lenguaje de marcado para la elaboración de páginas web.

◎ CSS 3:

Lenguaje para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

◎ BOOTSTRAP 4:

Conjunto de herramientas de código abierto para desarrollos web responsive con HTML, CSS y JavaScript.

◎ JAVASCRIPT ES6:

Lenguaje de programación interpretado, se utiliza principalmente del lado del cliente, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Herramientas

◎ VISUAL STUDIO CODE:

Editor de código fuente desarrollado por Microsoft.

◎ PHPMYADMIN:

Herramienta para manejar la administración de bases de datos MySQL a través de una interfaz web, utilizando un navegador.

◎ COMPOSER:

Sistema de gestión de dependencias de paquetes y librerías para PHP.

◎ NPM:

El sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.

Herramientas

◎ GIT:

Sistema de control de versiones para llevar el registro de los cambios en los archivos.

◎ GITHUB:

Plataforma de desarrollo colaborativo que permite tener el código fuente en la nube.

◎ ARTISAN:

Interfaz de línea de comandos incluida en Laravel.

Herramientas

◎ PHPUNIT:

Framework de código abierto para realizar pruebas unitarias en código PHP.

◎ LARAVEL DUSK:

Herramienta para realizar pruebas automatizadas de navegador.

◎ SONARQUBE:

Plataforma para evaluar código fuente, que usa diversas herramientas de análisis estático de código.

A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are solid grey and others are hollow with a grey outline. The lines are thin and grey, creating a mesh-like structure.

6.

Pruebas

Tipos de pruebas

- ◎ Pruebas automatizadas
- ◎ Pruebas manuales
- ◎ Análisis estático

Pruebas automatizadas

- ◎ Pruebas de características
- ◎ Pruebas unitarias
- ◎ Pruebas de navegador

Pruebas automatizadas - Características

- ⦿ Mediante estos tests se prueba la funcionalidad de la aplicación, tanto de funcionalidades independientes como de las que necesitan interactuar entre ellas o con agentes externos.
- ⦿ Estos tests tienen acceso a todo lo necesario para el funcionamiento de la aplicación, incluyendo la base de datos o peticiones a servicios externos.

Pruebas automatizadas - Características

- © Se han usado para comprobar funcionalidades como el enrutamiento:

```
public function testRouteLogin()  
{  
    $response = $this->get('login');  
    $response->assertStatus(200);  
}
```

Pruebas automatizadas - Características

- © O para comprobar la autenticación, como por ejemplo si un usuario tiene acceso a su propia lista de deseos marcada como privada:

```
public function testRoutePrivateWishlistWithAuthenticatedUserThatIsTheOwner()
{
    $userWithPrivateWishlist = User::where('wishlist', 0)->first();
    if ($userWithPrivateWishlist) {
        $response = $this->actingAs($userWithPrivateWishlist)->get('wishlist/' . $userWithPrivateWishlist->id);
        $response->assertStatus(200);
        $response->assertSee('Your wishlist is currently');
    } else {
        $this->assertNull($userWithPrivateWishlist);
    }
}
```

Pruebas automatizadas - Unitarias

- ⦿ Mediante los tests unitarios se prueba la funcionalidad de un solo método.
- ⦿ Estos tests son totalmente independiente unos de otros y no tienen acceso más que al método probado.
- ⦿ No pueden acceder a la base de datos o servicios externos.

Pruebas automatizadas - Unitarias

- © Se han usado estos tests para comprobar funciones concretas, como por ejemplo la de convertir una cadena de texto en minutos:

```
public function testConvertStringToMinutesFunction($expectedResult, $input)
{
    $gameController = new GameController();
    $results = $gameController->convertStringToMinutes($input);
    $this->assertEquals($expectedResult, $results);
}
```

Pruebas automatizadas - Unitarias

Para comprobar los distintos parámetros que podría recibir la función, se ha hecho uso de los data provider.

Los data provider permiten generar series de testeo con parámetros sobre un mismo método.

De este modo se evita repetir código.

```
public function provideStringsToMinutesData()
{
    return [
        'one hour' => [
            60,
            '1 Hour',
        ],
        'one hour and a half' => [
            90,
            '1½ Hour',
        ],
        'some hours' => [
            180,
            '3 Hours',
        ],
        'some hours and a half' => [
            210,
            '3½ Hours',
        ],
        'one minute' => [
            1,
            '1 Minute',
        ],
        'some minutes' => [
            45,
            '45 Minutes',
        ],
        'only seconds' => [
            0,
            '30 Seconds',
        ],
        'invalid string' => [
            0,
            'this is an invalid string to convert'
        ],
    ];
}
```

Pruebas automatizadas - De navegador

- © Simulan ser un usuario navegando por la aplicación y se utilizan para comprobar que determinadas acciones del usuario ofrecen el resultado esperado.
- © Para realizar estas pruebas se ha hecho uso de Laravel Dusk.
- © Se han utilizado estas pruebas para comprobar la correcta usabilidad de la aplicación evaluando el resultado de las posibles interacciones del usuario.

Pruebas automatizadas - De navegador

- ◎ Por ejemplo, para comprobar que se muestran errores de validación:

```
public function testRegisterWithWrongPasswordFormat()
{
    $this->browse(function (Browser $browser) {
        $browser->visit('/register')
            ->assertSee('Confirm Password')
            ->type('name', 'Dusk Test User')
            ->type('email', 'dusktestuser@email.com')
            ->type('password', 'password')
            ->type('password_confirmation', 'password')
            ->check('terms')
            ->press('Register')
            ->pause(1000)
            ->assertSee('The password must contain');
    });
}
```

Pruebas automatizadas - De navegador

- © O que los enlaces que abren modales funcionan correctamente y muestran el contenido esperado:

```
public function testLandingLinkToTerms()
{
    $this->browse(function (Browser $browser) {
        $browser->visit('/')
            ->assertSee('Manage your collections')
            ->press('Terms and conditions')
            ->whenAvailable('#termsModal', function ($modal) {
                $modal->waitForText('Terms and conditions of use')
                    ->assertSee('Terms and conditions of use');
            });
    });
}
```


Pruebas manuales

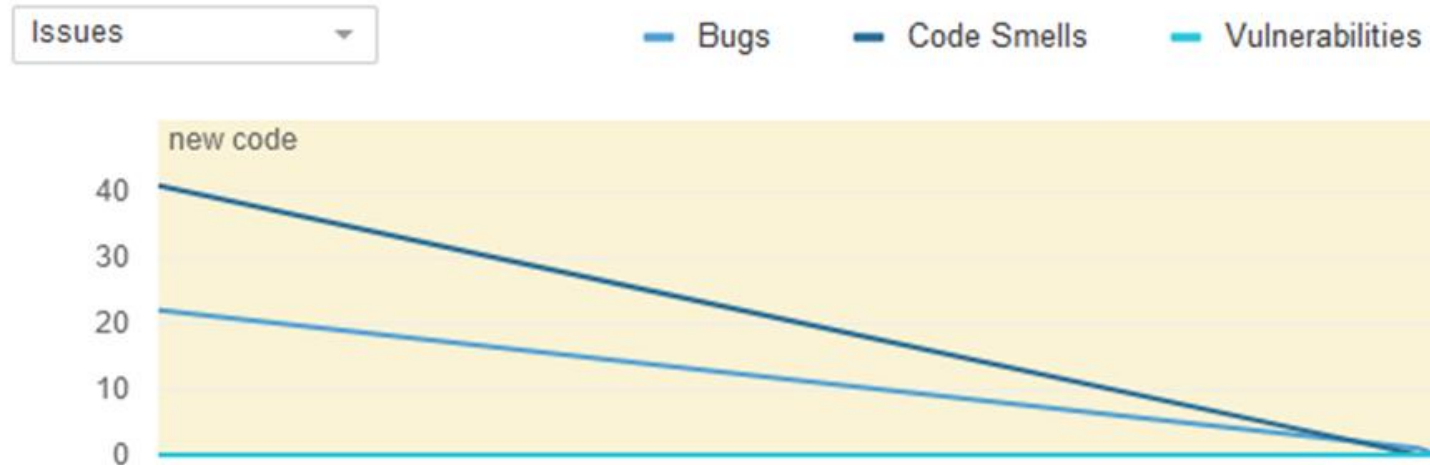
- ◎ Visualización en distintos dispositivos:
 - Ordenadores a distinta resolución
 - Tablet (horizontal y vertical)
 - Móvil (horizontal y vertical)
- ◎ Visualización en distintos navegadores:
 - Chrome
 - Firefox
 - Edge

Análisis estático

- ⦿ Este análisis se realiza sobre el código fuente de la aplicación y proporciona distintas métricas para medir la calidad del código.
- ⦿ Para la realización de este análisis se ha utilizado la herramienta Sonarqube.

Análisis estático

Se han ido resolviendo los distintos problemas con el código:



Análisis estático

Hasta obtener un resultado final de calidad:

0 🐛 Bugs

Reliability **A**

0 🔒 Vulnerabilities

Security **A**

0 🔒 Security Hotspots ⓘ

100% Reviewed

Security Review **A**

0 Debt

0 ☹️ Code Smells

Maintainability **A**

Resultados obtenidos

- ⦿ Todos los tipos de tests realizados han devuelto en un principio varios errores de diversa gravedad.
- ⦿ Estos errores se han ido solucionando y vuelto a comprobar hasta obtener resultados satisfactorios en todas las pruebas.

PASS Tests\Unit\GameFunctionTest

PASS Tests\Feature\RoutesTest

Tests: 27 passed
Time: 1.31s

Time: 01:16.404, Memory: 24.00 MB

OK (16 tests, 41 assertions)



Bugs

0



Vulnerabilities

0



Hotspots Reviewed

100%



Code Smells

0



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting different levels of connectivity or importance. The lines are thin and gray, creating a mesh-like structure.

7.

Conclusiones

Posibles ampliaciones

- ⦿ Utilizar la fecha de añadido en la tabla de objetos del usuario para implementar estadísticas por años o periodos de tiempo personalizables.
- ⦿ Permitir guardar un comentario personal del usuario en cada objeto.
- ⦿ Versión demo con usuario de prueba para usuarios sin registrar.
- ⦿ Elección del tema (otra paleta de colores, modo oscuro, etc.).
- ⦿ Panel de administración.

Licencia

- © La licencia elegida para la aplicación es la GNU General Public License v3.0.
- © Es compatible con las licencias de los paquetes utilizados en el proyecto.
- © Se han utilizado las herramientas proporcionadas por Composer y npm para obtener un listado de las licencias de las dependencias.

Licencia

```
PS C:\Users\Angela\Desktop\LARAVEL\mythings> composer licenses --format=summary
```

License	Number of dependencies
MIT	78
BSD-3-Clause	31
Apache-2.0	2

```
PS C:\Users\Angela\Desktop\LARAVEL\mythings> npx license-checker --summary
```

```
npx: instaló 57 en 6.339s
```

```
├─ MIT: 796
├─ ISC: 53
├─ BSD-2-Clause: 24
├─ BSD-3-Clause: 11
├─ Apache-2.0: 7
├─ Public Domain: 2
├─ CC0-1.0: 2
├─ 0BSD: 2
├─ (CC-BY-4.0 AND OFL-1.1 AND MIT): 1
├─ (MIT OR Apache-2.0): 1
├─ CC-BY-4.0: 1
├─ MIT OR GPL-2.0-or-later: 1
├─ (BSD-3-Clause OR GPL-2.0): 1
├─ (MIT AND Zlib): 1
├─ MIT*: 1
├─ (MIT AND BSD-3-Clause): 1
├─ (MIT OR CC0-1.0): 1
└─ BSD: 1
```

GNU General Public License v3.0

- © Licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto que garantiza a los usuarios finales la libertad de:
 - Usar
 - Estudiar
 - Compartir
 - Modificar

GNU General Public License v3.0

- © La GPL se distingue del dominio público o de otras licencias de software libre conocidas como permisivas por hacer hincapié en el copyleft.
- © El copyleft es una práctica legal que consiste en propiciar el libre uso y distribución de una obra, exigiendo que las copias y derivados perpetúen la misma licencia.

GNU General Public License v3.0

◎ Permite:

- Uso comercial
- Modificación
- Distribución
- Patente
- Uso privado

◎ Obliga a:

- Publicar el código
- Mantener la misma licencia



GRACIAS
POR SU ATENCIÓN