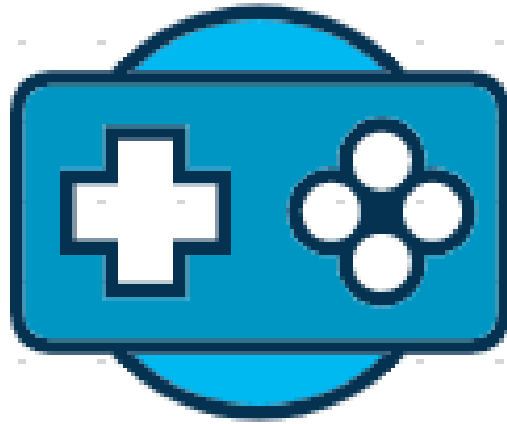


PROYECTO FIN DE GRADO



GameInfo

Alumno: Alejandro Boix Perez

Tutor: David Escrich

Game Info

Introducción y Justificación	2
Objetivos y Metodología	3
Análisis del contexto	5
Javascript:	5
JSON:	5
HTML:	5
BOOTSTRAP:	5
Node JS:	5
XAMPP:	6
MYSQL:	6
VISUAL STUDIO CODE:	6
POSTMAN:	6
DESARROLLO DEL CONTENIDO	7
Explicación de la app:	7
Ejemplo de uso de la aplicación:	8
Imágenes de la aplicación:	9
Búsqueda de un juego:	11
Apartado Listas:	13
BASE DE DATOS:	14
Api Externa IGDB:	16
Problemas encontrados:	17
Problemas y errores relacionados con código:	17
Explicación de fragmentos de código interesantes:	18
El funcionamiento de POST:	18
La generación de código html dinámico mediante archivos JSON:	18
El funcionamiento de bootstrap:	19
Conclusión y valoración personal:	21
Recursos utilizados:	22

Introducción y Justificación

Este es el proyecto final de grado del ciclo de Desarrollo de Aplicaciones Multiplataforma 2020-2021.

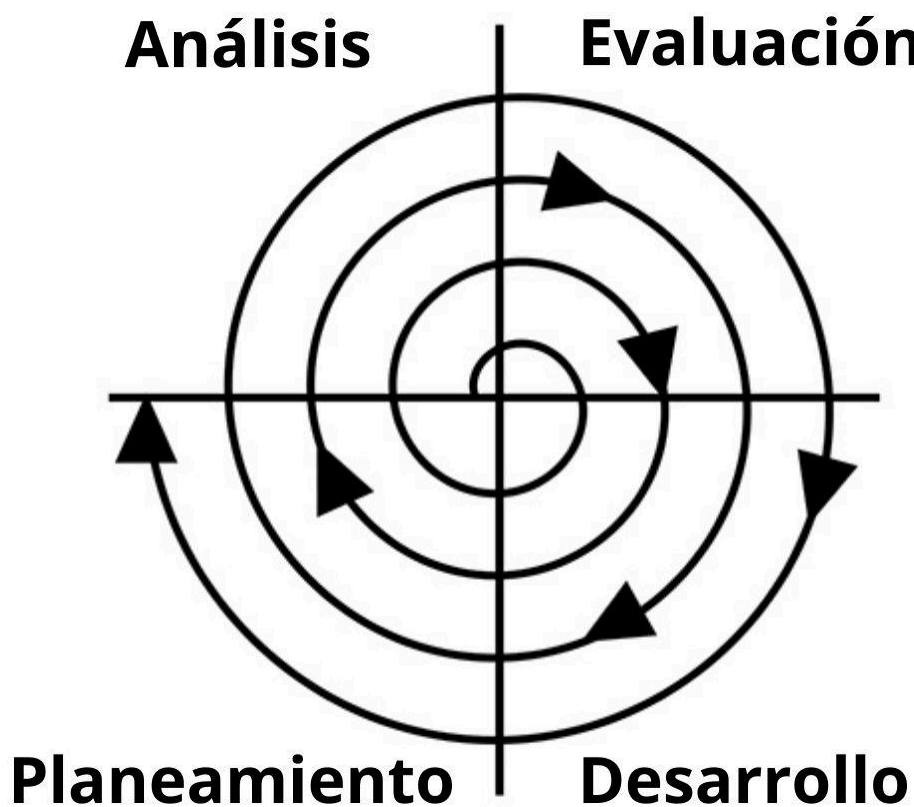
Se ha desarrollado una aplicación web con el objetivo de ser una fuente de información y datos de videojuegos, optimizada para funcionar tanto en dispositivos PC/MAC como en plataformas móviles Android/iOS.

Contendrá:

- Diseño Responsive para poder funcionar en cualquier tipo de dispositivo
- Diseño intuitivo y usable para poder navegar por ella fácilmente
- Usando las últimas tecnologías tanto en front-End como en back-End

Objetivos y Metodología

Se ha usado la metodología de diseño en cascada es decir una evaluación continua de la propia aplicación para un testing continuo y un fácil acceso para añadir mejoras.



La aplicación contiene las siguientes características:

- Un Login y Registro de usuarios
- Se podrá crear listas de videojuegos
- Se podrá poner nota a los videojuegos

Game Info

-Un apartado de búsqueda en el cual se enseñaran los resultados de la siguiente manera.

- Título
- Portada
- Historia resumida(150 caracteres)

-Un apartado de ficha de los videojuegos que contendrán.

- Título
- Portada del título
- Desarrolladores y estudios asistentes, con su título e imagen.
- Un listado de 3 videojuegos similares con hipervínculos a sus propias fichas técnicas.
- Un apartado de 5 temas de los que trata el juego
- Un texto en el cual se encontrará la historia del título
- Un trailer del propio videojuego
- Un apartado con 5 capturas del propio videojuego
- Un listado con las plataformas en las que está disponible

Análisis del contexto

Aquí se van a detallar todas las herramientas y lenguajes utilizadas en la creación de la aplicación.

Javascript:

Se ha usado javascript tanto como para el backend como el front-end, sobre todo para hacer consultas al servidor node js al querer recibir datos como a la propia generación automática de contenido html mediante código javascript al recibir archivos JSON.

JSON:

Se ha usado este tipo de archivos para el momento de comunicar información entre la API-SERVIDOR y SERVIDOR-APP, estos archivos son muy útiles para pasar cantidades de información altas dado que usan un sistema LLAVE-DATO con el cual podemos encontrar la información rápidamente.

HTML:

Se ha usado esta tecnología para dar estructura a la aplicación web.

BOOTSTRAP:

Se ha usado esta tecnología para poder hacer la aplicación “responsive” para dar soporte a dispositivos con pantallas muy pequeñas.

Bootstrap es una librería de javascript y css la cual incorpora métodos predefinidos para hacer cualquier tipo de aplicación responsive.

Node JS:

Servidor BACK-END asíncrono con el cual podemos hacer llamadas a la api externa de igdb para poder recibir archivos JSON con los parámetros que le hemos proporcionado.

Aparte también se ha usado para hacer de “middleware” entre la aplicación web y la propia api, dado que la api te obliga a usar algún tipo de servidor para realizar llamadas para así no exponer los credenciales de acceso a esta.

Este servidor usa el lenguaje javascript.

Game Info

XAMPP:

Se ha utilizado XAMPP como motor de bases de datos, con un plugin de node js para poder darle soporte.

MYSQL:

Se ha utilizado una base de datos MYSQL relacional para poder crear los usuarios y listas de videojuegos.

VISUAL STUDIO CODE:

He utilizado este editor de código porque tiene soporte para todos los tipos de archivos que iba a utilizar con plugins para la programación, además de que no se recomienda usar VISUAL STUDIO CODE o el terminal para realizar la programación.

POSTMAN:

Una aplicación que usa para poder hacer tests a API's, dispone de aplicación web y de pc, y no hace falta tener que hacer código para poder hacer tests de las queries de las llamadas a la API

DESARROLLO DEL CONTENIDO

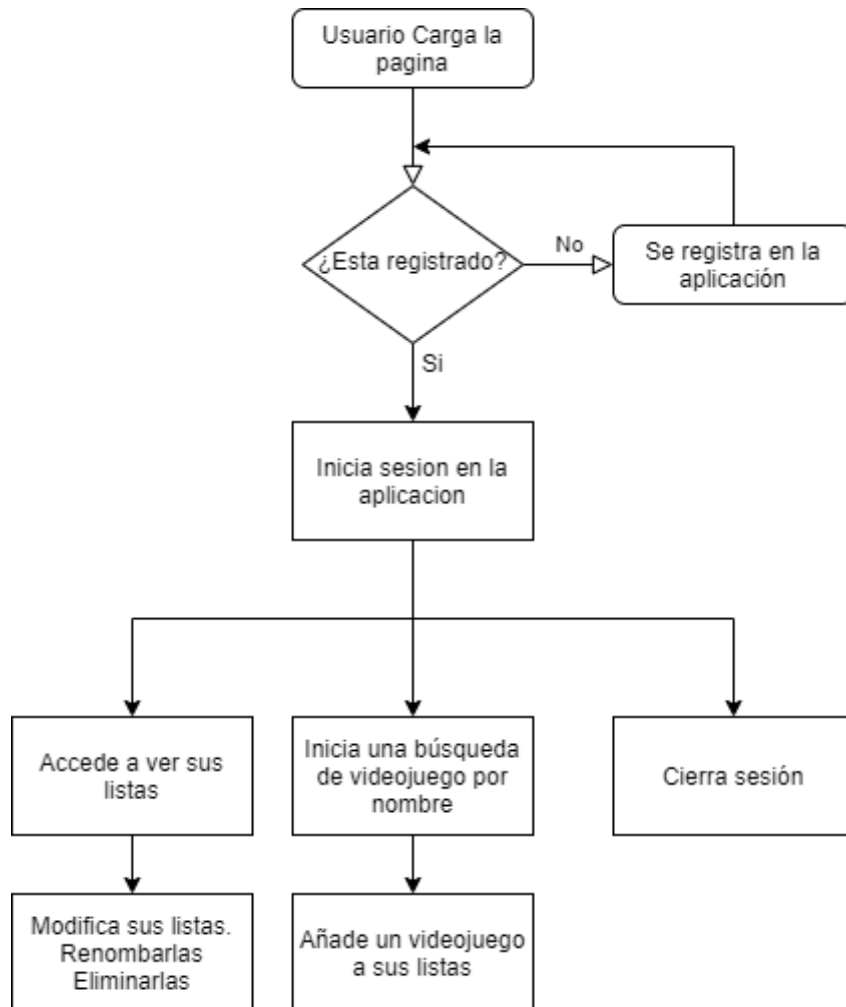
Aquí se va a explicar el uso de los mecanismos de la propia aplicación, y el propio funcionamiento externo e interno.

Explicación de la app:

Una aplicación WEB

La aplicación sería una especie de imdb de los videojuegos, es decir escribir el nombre del título y obtener información inmediatamente de este, aparte el usuario podrá registrarse en la aplicación y crear listas de sus juegos favoritos, juegos que haya jugado y juegos que le interesen pero no hayan salido aún.

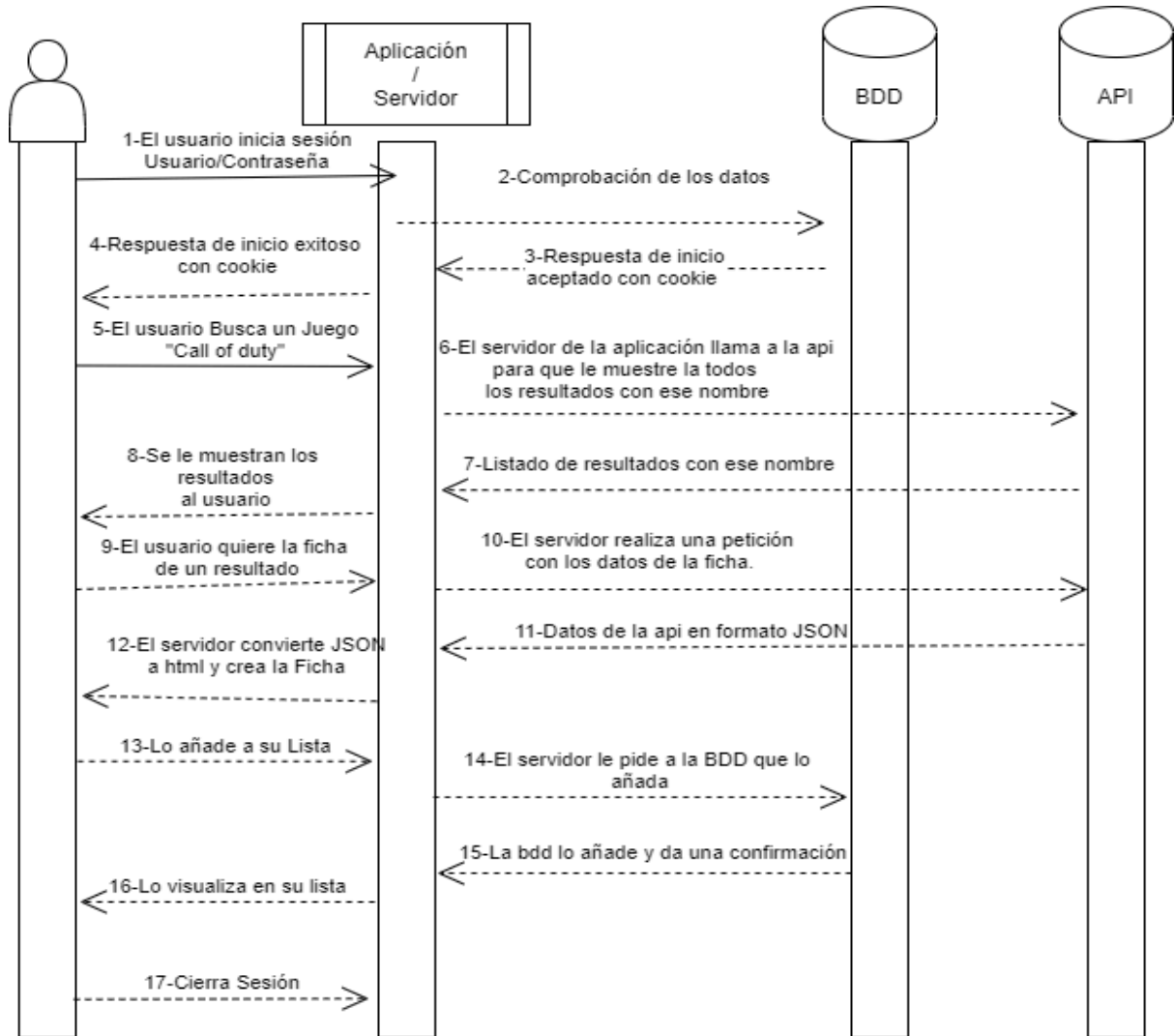
Diagrama de la aplicación de flujo:



Ejemplo de uso de la aplicación:

Caso de uso de la aplicación donde un usuario realizará las siguientes acciones:

1. Inicio de sesión
2. Búsqueda de un videojuego
3. Selección de uno de la lista que obtiene
4. Visualización de la ficha del juego
5. Añadirlo a sus listas.
6. Cerrar sesión

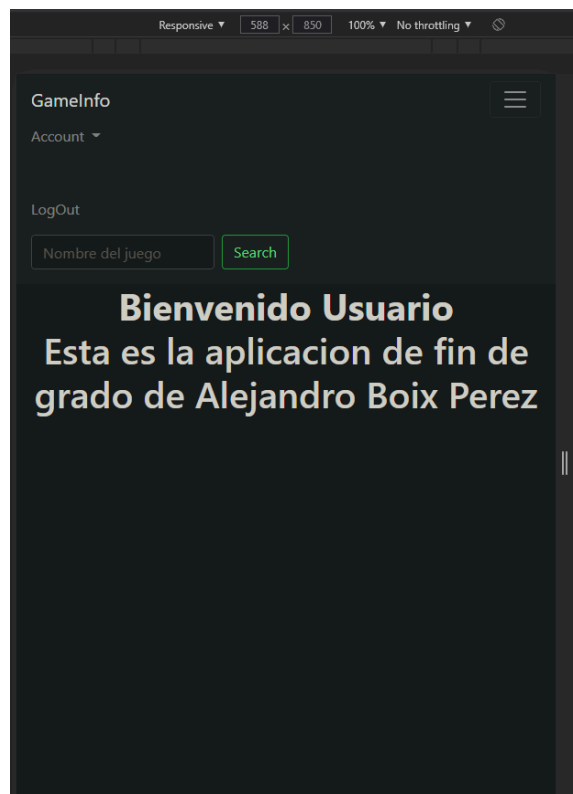
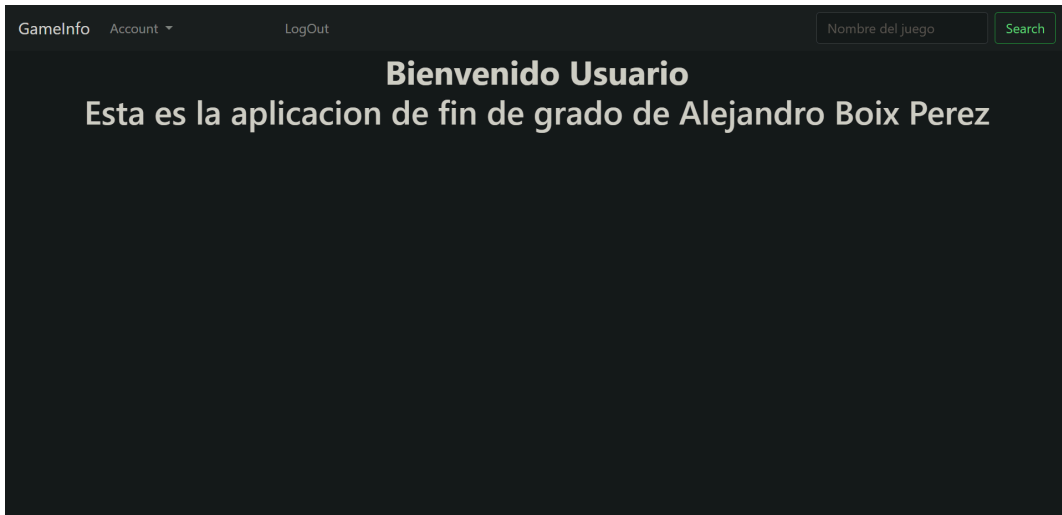


Game Info

Imágenes de la aplicación:

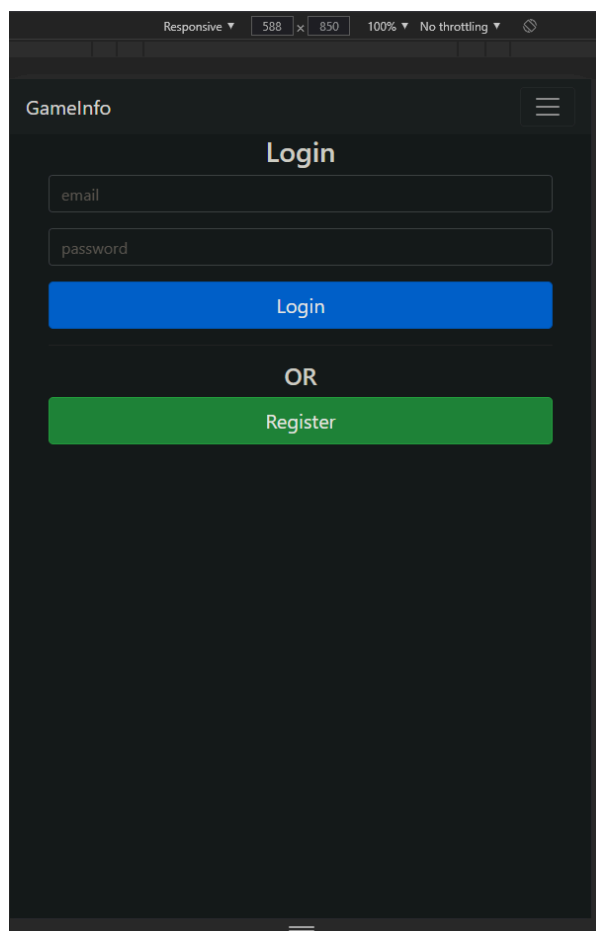
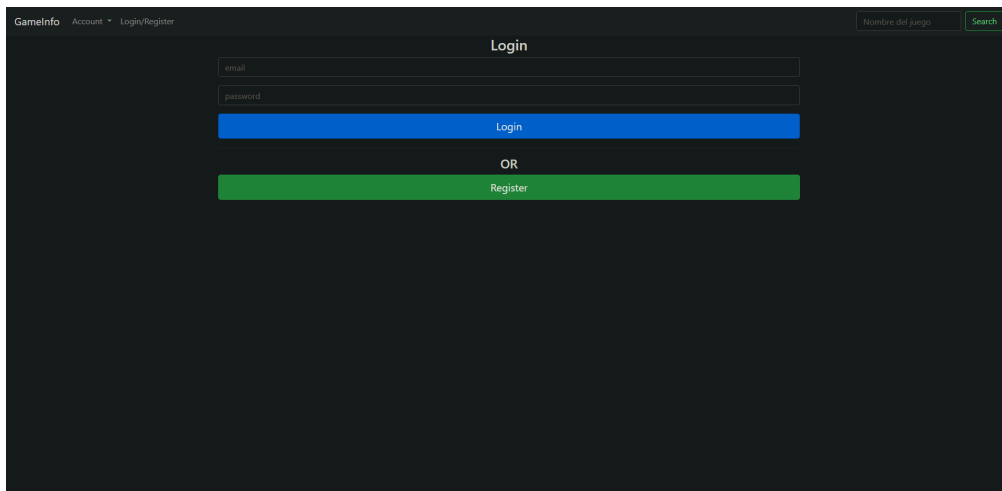
Las de cada parte de la aplicación habran 2 imagenes 1 en formato 16:9(pc) y otra en formato móvil con la ayuda del device toolbar de google, que me permite cambiar la relación aspecto y tamaño de pantalla de lo que se está renderizando.

Página principal:



Game Info

Inicio de sesión:



Game Info

Búsqueda de un juego:



Game Info

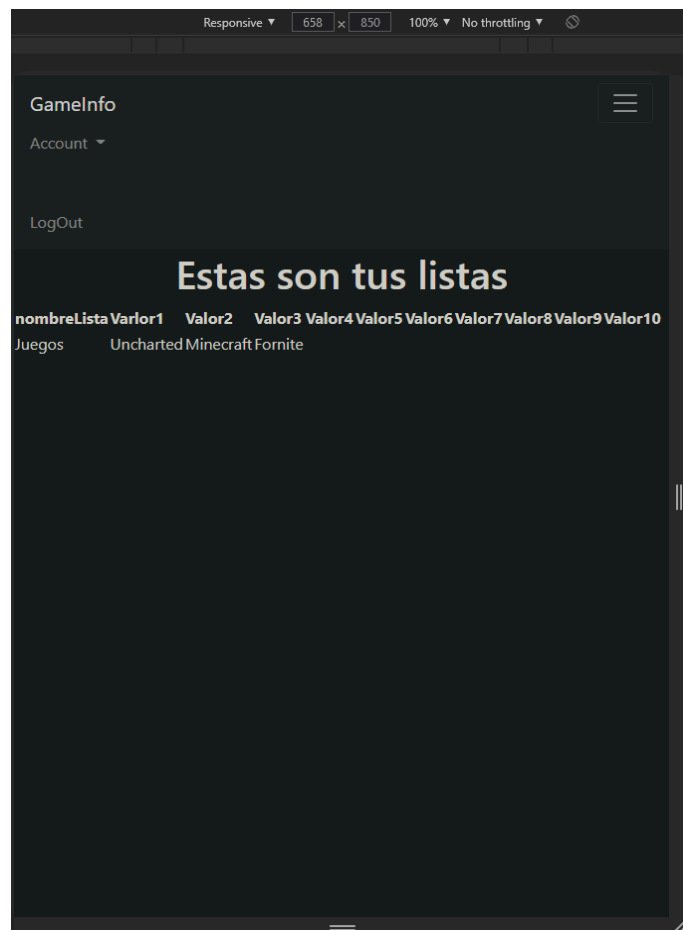
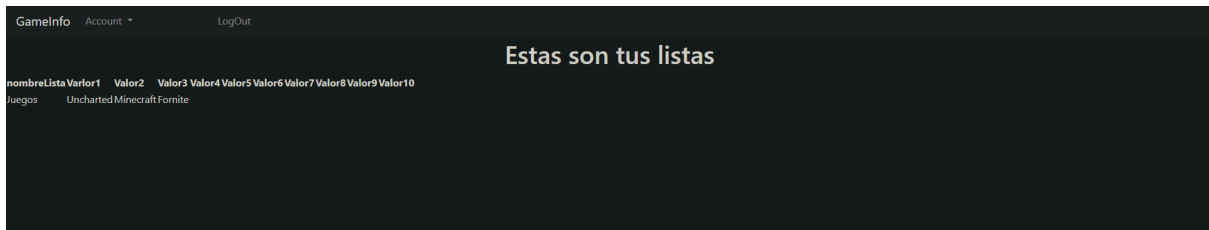
Ficha de un juego:

The desktop view of the Battlefield 4 game page on GamelInfo. At the top left, there are links for 'GamelInfo', 'Account', and 'LogOut'. The main title 'Battlefield 4' is centered. Below it, the 'Summary' section contains a paragraph: 'Battlefield 4™ is the genre-defining action blockbuster created by DICE, and made from moments that blur the line between game and glory – moments found only in Battlefield. Go to Boot Camp to learn the basics of Battlefield 4's engaging single player and intense multiplayer mode. Here you'll also find intel on Levolution, Commander Mode, Naval Warfare and many more exciting game features.' To the left of the summary is a 'Cover' image of a soldier in a combat environment, with the text '29/10/2013 1:00:00' and 'Developed by : EA Digital Illusions CE'. Below the cover is a 'Trailer' section featuring a YouTube video player with the title 'Battlefield 4: Official 17 Minutes "Fishing in Baku" Gamepl...'. At the bottom, there is a 'ScreenShots' section with a preview image of a soldier and a helicopter.

The mobile view of the Battlefield 4 game page on GamelInfo. At the top, there are browser controls: 'Responsive', '600 x 850', '100%', 'No throttling', and a refresh icon. The 'GamelInfo' logo is in the top left, and a menu icon is in the top right. The main title 'Battlefield 4' is centered. Below it, the 'Summary' section contains the same paragraph as the desktop view. To the left is the 'Cover' image with the text '29/10/2013 1:00:00' and 'Developed by : EA Digital Illusions CE'. Below the cover is the 'EA' logo and the text 'Electronic Arts'. At the bottom, there is a 'ScreenShots' section with a preview image of a soldier and a helicopter. The page is designed to be responsive, with elements stacked vertically.

Game Info

Apartado Listas:



BASE DE DATOS:

El diseño de la base de datos es el siguiente:

- Tabla Usuarios:
 - ID
 - Clave Primaria,INT de 10 caracteres, Autoincremental
 - Nombre
 - Varchar de 15 caracteres
 - Nombre del usuario
 - Contraseña
 - Varchar de 15 caracteres
 - Contraseña del usuario
 - Cookie
 - Varchar de 10 caracteres
 - Cookie que contendrá un código que hará referencia al usuario para poder mantener la sesión abierta aunque cierre la página.
- Tabla Listas:
 - ID Lista
 - Clave Primaria,INT de 10 caracteres,Autoincremental
 - ID Usuario Lista
 - Este campo está relacionado con ID de la tabla usuarios
 - INT de 10 caracteres
 - NombreLista
 - Varchar de 50 caracteres
 - Nombre que le pondrá el usuario a su propia lista
 - Valor 1
 - Valor 2
 - Valor 3
 - Valor 4
 - Valor 5
 - Valor 6
 - Valor 7
 - Valor 8
 - Valor 9
 - Valor 10
 - Todos los campos Valor tienen un Varchar de 50 caracteres, y su intención es para que vaya el nombre del videojuego

Game Info

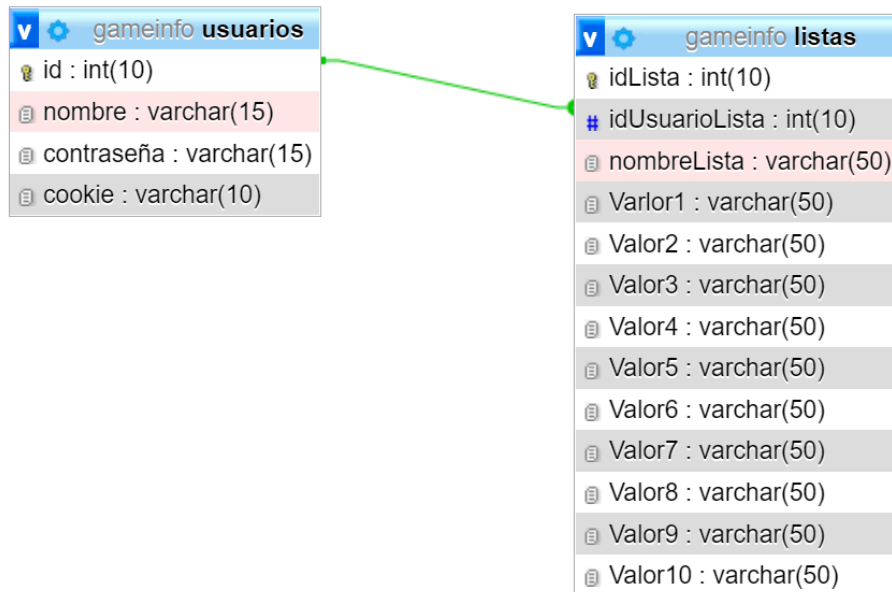


Imagen de la base de datos dentro de PHPMYADMIN.

id	nombre	contraseña	cookie
1	alex	admin1	0000

idLista	idUsuarioLista	nombreLista	Varlor1	Valor2	Valor3
1	1	Juegos de aventuras	Uncharted	Minecraft	Fornite

Ejemplo de los campos rellenos de la base de datos

Game Info

Api Externa IGDB:

IGDB es una plataforma de recopilación de videojuegos la cual pone a disposición de sus usuarios registrados un acceso a su API para poder realizar peticiones mediante una licencia de uso abierto mientras no se sobrepase los límites ofrecidos de peticiones por segundo y peticiones máximas mensuales.

La api usa una metodología de POST para poder realizar las peticiones, mandando dentro del POST un archivo JSON con la información que quieres solicitar.

Ejemplo de una consulta a la api de IGDB:

```
const getQuery = async function (nombre){
var data = 'limit 20;search "' + nombre + '";fields name,rating,summary,cover.url,storyline,themes.name;where category = 0 & cover != null;';
var config = {
  method: 'post',
  url: 'https://api.igdb.com/v4/games',
  headers: {
    'Client-ID': 't2lt4vbl4vxv3viitz2t1y0mz64kqv',
    'Authorization': 'Bearer vkf2utprb0bxbgaiukaa0dd8hpn0ob0',
    'Content-Type': 'text/plain'
  },
  data : data
}
try {
  return resPOST = await axios(config);
} catch (error){
  console.log(error);
}
}
```

En el campo data contiene una string que se enviará en el cuerpo del JSON la cual la API usará como query cuando nos quiera devolver los resultados.

Problemas encontrados:

El problema principal que he tenido yo con mi proyecto es no conocer mis propias habilidades bien, y no prever problemas, este proyecto lo he acabado el último día antes de la presentación.

Tener que aprender prácticamente un lenguaje desde 0, es el caso de javascript, aunque al ya saber otros lenguajes se ha hecho más ameno, pero sigue siendo un lenguaje diferente que no se parece a ninguno de los que hayamos dado, como python o java.

Problemas y errores relacionados con código:

- Más de una vez la API ha dejado de funcionar porque la estaban actualizando o haciéndole mantenimiento (aunque ellos avisaban por su discord oficial).
- Cargar imágenes con formatos extraños y que la página web no las pueda renderizar bien, aunque bootstrap sea responsive muchas veces no puede cambiar la relación aspecto de alguna imagen sin destrozarla y que no se vea nada.
- Muchos errores al realizar peticiones lo cual me frustraba dado que en la aplicación postman funcionaban perfectamente pero al pasarlo a node js con axios no funcionaba.
- Muchos cuelgues del servidor NODE JS por crear demasiadas peticiones y sobrecargando la memoria, dado que el servidor hasta que no recibe la información pedida y la envía y recibe una confirmación de que ha sido recibida no libera ese espacio en memoria.

La mayoría de problemas he podido solucionarlos o tomar otra vía para poder sobrepasarlos

Explicación de fragmentos de código interesantes:

El funcionamiento de POST:

```
app.post("/login", (request, response) => {
  var datos = request.header("datos")
  datos = datos.split(",")
  console.log(datos)
  var nombre = datos[0]
  console.log(nombre)
  var contraseña = datos[1]
  var accion = datos[2]
  var archivoJson = ""
  if(accion = "Login"){
    try{
      var conexion = mysql.createConnection({
        host: 'localhost',
        database: 'gameinfo',
        user: 'root',
        password: ''
      })
      conexion.connect(function(error){
        if(error){
          throw error;
        }else{
          console.log('Database Connection')
        }
      });
      conexion.query('SELECT nombre, contraseña from usuarios where nombre=?', [nombre], function(error, results, fields){
        if(error){
          throw error;
        }
      })
    }
  }
})
```

El tema de las solicitudes POST para poder obtener datos, es una de las claves de hacer páginas dinámicas, en este proyecto yo los he usado para hacer peticiones a mi base de datos y a la propia API de IGDB.

La generación de código html dinámico mediante archivos JSON:

```
for(i=0; i<contarJSON(json[0].involved_companies); i++){
  DeveloperString += `<div ><img style="max-height: 250px" src=${imagen1080p(json[0].involved_companies[i].company.logo.url)} class="card-img-top">
`
}
for(i=0; i<5; i++){
  gameCategory += `<li class="list-group-item">${json[0].keywords[i].name}</li>`
}
```

En este fragmento de código, que es el que se usa en el apartado ficha para poder poner todos los estudios de desarrollo que han participado en un juego, por que este dato varía mucho, puede que haya solo 1 estudio, o en juegos grandes pueden llegar a participar 5 o 6, y en la ficha de videojuegos yo quería poder poner la imagen de su compañía y el nombre de esta, así que lo que he hecho a sido iterar sobre el archivo JSON sumandole el código html, que contiene la imagen de la compañía y su

Game Info

nombre dentro de un apartado DIV, para que así tenga límites y sigue siendo responsive al estar hecho con bootstrap.

El funcionamiento de bootstrap:

Bootstrap es una librería de css y javascript que aplica estilos a prácticamente todos los elementos existentes en html, usa un sistema de GRIDS o tablas el cual divide la pagina web en 12 columnas iguales de espacio, estas 12 columnas se pueden repartir entre todos los elementos que necesites como en el siguiente ejemplo.

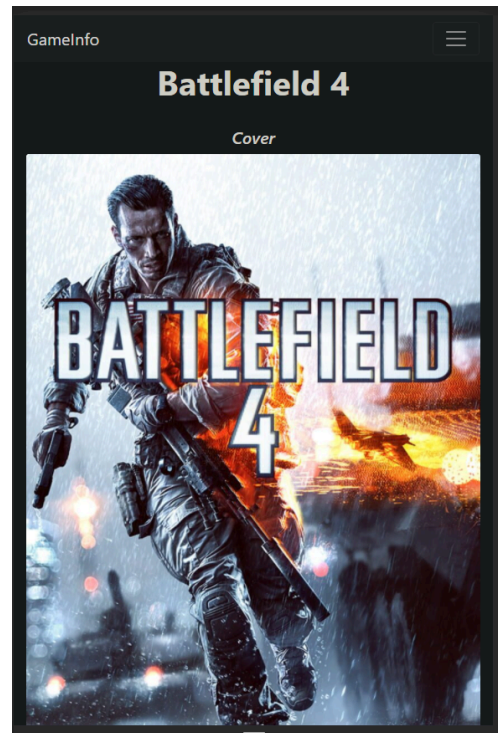


The screenshot shows a dark-themed web page layout for 'Battlefield 4'. At the top left is the text 'GameInfo' and at the top right is a hamburger menu icon. The main title 'Battlefield 4' is centered in a large, bold, white font. Below the title, the layout is split into two columns. The left column, labeled 'Cover', contains a game cover image, the release date '29/10/2013', and a duration '1:00:00'. The right column, labeled 'Summary', contains a paragraph of text describing the game.

En este ejemplo las 12 columnas se dividen de la siguiente manera:

-Cover que es un div que ocupa la columna entera, esta ocupando 4 de las 12 columnas que hay en pantalla, aunque no se han visibles están ahí, el div contiene el siguiente elemento en su clase `<div class="col-sm-4">` el cual le indica que el div mientras que la pantalla sea mayor a tamaño sm(768 px de ancho) ocupa 4 columnas pero en el momento que el tamaño sea inferior.El div ocupará todo el ancho y se sobrepondrá sobre el div de la derecha(Summary)

Game Info



tanks

Summary

Battlefield 4™ is the genre-defining action blockbuster created by DICE, and made from moments that blur the line between game and glory – moments found only in Battlefield. Go to Boot Camp to learn the basics of Battlefield 4's engaging single player and intense multiplayer mode. Here you'll also find intel on Levolution, Commander Mode, Naval Warfare and many more exciting game features.

Trailer:



ScreenShots

Summary está debajo de categorías, que es el último elemento del div de la izquierda.

Conclusión y valoración personal:

Gracias a este proyecto y las prácticas he aprendido bastante sobre desarrollo web y backend, me parece un tema muy interesante el como funciona por detrás un servidor, al cargar html o hacer peticiones a servidores externos,.

Programación asíncrona es una de las cosas que he aprendido al tener que tratar con peticiones, se trata de plantear las funciones de una manera a destiempo, dado que los servidores externos como la api no responden instantáneamente, así que hay que usar funciones asíncronas y callbacks, para que cuando lleguen las respuestas, se siga ejecutando ese segmento de código.

Una cosa sobre lo que he aprendido bastante a sido sobre el uso de ficheros JSON y lo potentes que son para pasar información en grandes cantidades y luego poder crear una web dinámica (ficha del producto) la cual dependiendo de cuantos datos lleve el archivo JSON generará un archivo html distinto.

Las cookies también es un tema que me ha parecido muy interesante, el cómo lo usan las aplicaciones para recabar información, o como lo uso yo en mi aplicación como una manera de recordar quién es el cliente de la aplicación sin tener que tener guardada su contraseña/usuario dentro de una variable.

En general este proyecto me va a servir como portfolio para un futuro, aparte he aprendido varias habilidades que están muy demandadas en el mercado como el manejo de servidores node JS y creación de aplicaciones web responsive con bootstrap.

Recursos utilizados:

Para información sobre bootstrap y diseños:

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

Para dudas generales de código:

<https://www.google.com/>

<https://stackoverflow.com/>

<https://www.w3schools.com/>

<https://www.youtube.com/>

Para aprender el funcionamiento de node js y las Apis::

https://www.youtube.com/playlist?list=PLRqwX-V7Uu6YxDKpFzf_2D84p0cyk4T7X

<https://nodejs.org/en/docs/>

<https://www.w3schools.com/nodejs/>

Para aprender a cómo se hacen las consultas de la Api IGDB:

<https://api-docs.igdb.com/#about> (Api oficial)

<https://discord.gg/FrvfwQg> (Foro donde estan los administradores para dar soporte)