

En este documento se redacta toda la información derivada a la creación e implementación del proyecto de fin de curso. Definición general del mismo, motivación, especificaciones, herramientas y recursos utilizados para la creación de éste, la planificación que se ha llevado, procedimientos que se han de llevar para la instalación del mismo, la arquitectura del sistema y las posibles mejoras que se implementarán en el futuro.

## Proyecto de fin de ciclo.

Manuel Palomino Montealegre

---

## Contenido

1. Definición general. ....	2
2. Motivaciones. ....	3
3. Información general. ....	4
4. Herramientas y recursos. ....	6
5. Planificación, desarrollo, metodología y códigos de estilo. ....	7
6. Procedimientos de instalación y prueba. ....	12
7. Arquitectura del sistema y modelo de datos. ....	17
8. Mejoras futuras. ....	19

# 1. Definición general

## 1.1. ¿Qué es este proyecto?

Este proyecto es una página web donde los usuarios pueden consultar información de videojuegos. Los usuarios también pueden insertar comentarios en la web. La web hace una media de todas las notas que los usuarios ponen en sus comentarios y la muestra.

## 1.2. ¿Con que propósito se ha llevado a cabo?

En la actualidad existe una gran cantidad de videojuegos, esa cantidad aumenta en gran medida con el paso del tiempo, eso lleva a que un montón de ellos pasen desapercibidos.

Con ayuda de mi proyecto se pueden dar a conocer todos los juegos que merezcan la pena, desde el punto de vista de los usuarios, ya que son los usuarios los que evalúan si un juego merece la pena o no.

## 1.3. ¿Para quién ha sido desarrollado este proyecto?

Este proyecto está dirigido a aquellas personas que adquieren una plataforma y no saben que títulos comprar. Mi proyecto ayuda a los usuarios brindándoles un montón de opciones.

Con la opción de escribir comentarios, los usuarios que hayan probado un título pueden dar a conocer su opinión. Esta opinión ayuda a los demás usuarios a elegir qué títulos tienen que adquirir.

## 2.Motivaciones.

### 1.1. Motivaciones como usuario.

Las principales motivaciones como usuario que han llevado a la creación de este proyecto han sido:

- Como persona a la que le apasiona el mundo de los videojuegos, conozco bastante sobre el tema y eso me ha permitido mostrar información coherente en la web.
- Aun conociendo una gran cantidad de títulos, en ocasiones me ha surgido la duda de qué títulos debería probar, la principal barrera para conocer nuevos títulos es la falta de sitios como el de mi proyecto donde poder conocer los existentes y los que van saliendo al mercado.
- La posibilidad de recomendarle esta página a usuarios que no conozcan tanto el mundo de los videojuegos les permite conocer títulos importantes en la industria del videojuego con facilidad.

### 1.2. Motivaciones a nivel profesional.

Las principales motivaciones a nivel profesional que han llevado a la creación de este proyecto han sido:

- La utilización del framework Angular como cliente me ha llevado a profundizar mis conocimientos.
- El hecho de realizar y adaptar el proyecto para cualquier tipo de público. Ha hecho darme cuenta de lo importante que es tener una web accesible.
- Con este proyecto he intentado abarcar todos los conocimientos impartidos en el ciclo y profundizarlos.

# 3. Información general.

## 3.1. Especificaciones.

### 2.1.1. Cliente

He creado los siguientes componentes:

- Funciones necesarias para mandar la información que el usuario redacta o crea al servidor.
- Pasos en las funciones para verificar la integridad y la correcta estructura de la información mandada al servidor.
- Componentes para obtener información del servidor, manipular y adaptar la información recibida del servidor con el fin de darle una coherencia y una mejor legibilidad a la información.
- Controles de zonas para comprobar a qué zonas pueden acceder los usuarios según su rol.
- Un diseño adaptado a cualquier pantalla y accesible para cualquier tipo de usuario.

### 2.1.2. Servidor.

He creado los siguientes componentes:

- Índice donde llamo a todas las funciones del servidor divididas por usuarios, juegos y comentarios
- He comprobado que se produce la correcta manipulación y recogida de los datos provenientes del cliente.
- Peticiones anidadas a la base de datos.
- He comprobado que al eliminar un elemento físico de la base de datos este se elimina de la memoria del servidor
- Salidas en caso de que dé fallo o el resultado sea positivo.
- Sistemas para garantizar la seguridad en operaciones donde es necesario.

### 2.1.3. Base de datos.

He creado las siguientes funciones:

- Tablas para cada tipo de dato.
- Claves principales como id (Tabla) siendo este campo único.
- Campos de fechas.
- Claves foráneas.
- Relaciones para cada tabla.

## 3.2.Requisitos.

Los requisitos del proyecto son los siguientes:

- He creado funciones para el insertado, lectura, modificación y eliminación de datos.
- Las funciones que he creado permiten la comunicación entre el backend, Frontend y la base de datos.
- El desarrollo de un diseño funcional y capaz de ejecutarse en cualquier dispositivo.
- Tablas donde almacenar los datos que se generan.
- Implementación de sistemas de seguridad.
- Verificación de que los datos enviados y recibidos son correctos.

## 3.3.Alcance.

Mi proyecto consigue suplir todas las necesidades de cualquier usuario a nivel de recogida y envío de datos. Estos datos están relacionados con la idea principal del proyecto. Esta idea es la de un sitio web donde informarse de juegos, visualizar opiniones de usuarios y escribir opiniones de forma anónima.

El sitio web también es capaz de mostrar juegos de distintas formas, administrar perfiles de usuario y administrar comentarios.

## 3.4.Limitaciones del proyecto.

Las principales limitaciones del proyecto según mi opinión son:

La poca utilidad que tiene el perfil de usuario en este sitio web. El perfil de usuario solo se utiliza para comprobar si el usuario está registrado y ha iniciado sesión en la web, se podría haber utilizado esa información para funcionalidades que le den un uso más avanzado.

## 4.Herramientas y Recursos.

### 4.1.Herramientas y recursos utilizados a nivel desarrollo.

- **NetBeans:** He utilizado la herramienta NetBeans como entorno de desarrollo para la generación del backend en PHP, esta herramienta facilita generar código por la experiencia que tengo en la utilización de esta herramienta.
- **Visual Studio Code:** He utilizado esta herramienta para desarrollar todo el código del Frontend utilizando el framework Angular, esta herramienta me facilita generar código y la organización del mismo.
- **XAMP:** Esta herramienta la utilizo para la creación de un entorno local donde ejecutar el proyecto y crear bases de datos.

### 4.2.Herramientas y recursos utilizados a nivel de diseño.

- **Bootstrap:** Las librerías de Bootstrap las utilizo para darle diseño a tablas.
- **Free Frontend:** Los botones de mi web los he generado a partir de esta web.

# 5. Planificación, desarrollo, metodología y códigos de estilo.

## 5.1. Planificación.

- Creación de base de datos.
- Desarrollo funcional de las distintas acciones disponibles en el proyecto.
- Prueba del correcto funcionamiento de las acciones desarrolladas.
- Consulta al tutor sobre el funcionamiento de las acciones desarrolladas.
- Creación de diseño.
- Pruebas sobre el diseño
- Depurado de todo el proyecto.

## 5.2. Desarrollo.

Creo los siguientes componentes:

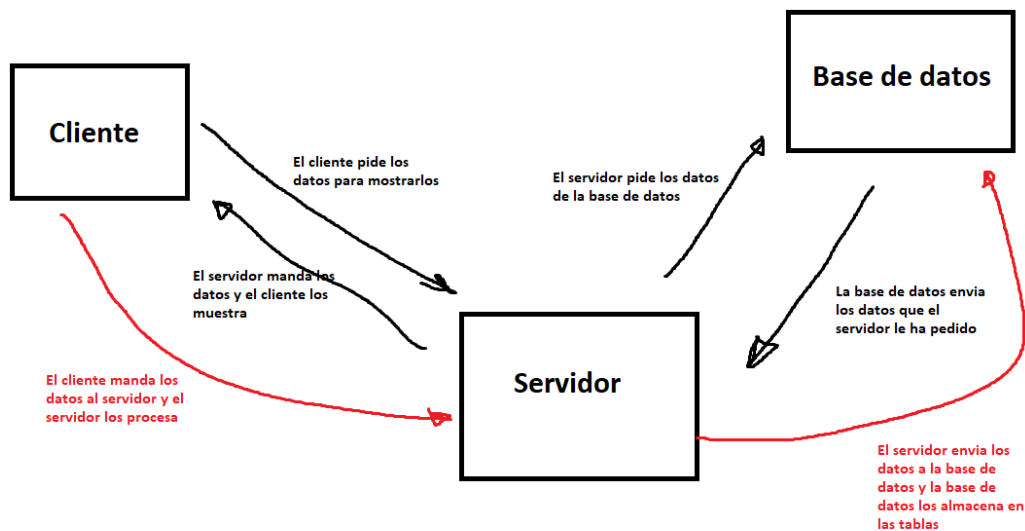
- **Registro de usuarios:** el usuario puede crear una cuenta en la web con sus datos.
- **Inicio de sesión:** el usuario pone sus datos, el servidor comprueba si esos datos son correctos y en caso afirmativo se inicia sesión al usuario.
- **Perfil de usuario:** el usuario puede visualizar los comentarios que ha creado y eliminarlos, también puede editar los datos de su perfil.
- **Recuperar contraseña:** en este componente el usuario puede cambiar su contraseña introduciendo sus datos.
- **Inicio:** aquí se muestran los juegos clasificados por tipos.
- **Inicio plataforma:** aquí se muestran los juegos clasificados por tipos y plataforma elegida.
- **Ver juego:** en esta pestaña se muestran todos los datos sobre el juego elegido (multimedia, comentarios datos generales)
- **Reportar comentario:** el usuario puede reportar un comentario rellenando un formulario.
- **Listado de juegos:** se muestra todos los juegos existentes.
- **Panel de administrador:** se muestra las distintas opciones que tiene un administrador.
- **Listado de juegos administrador:** aquí el administrador puede ver, eliminar o editar todos los juegos.
- **Insertar juego:** el administrador puede insertar un juego.



- **Editar juego:** el administrador puede editar un juego.
- **Listar reportes:** se muestran todos los reportes.
- **Ver reporte:** se muestra los datos del reporte y los datos del comentario, pudiendo eliminar el reporte o el comentario.

## 5.3. Metodología.

En la imagen muestro cómo se han creado las funciones y cómo interactúan las partes entre ellas.



## 5.4. Códigos de estilo.

### 5.4.1. Botones.

Los botones están divididos en 2 grupos los botones genéricos y los botones del menú de ver juego.

- **Botones genéricos:** tienen el mismo diseño a diferencia del color. El color tiene un significado funcional, gris: ver, rojo: eliminar, azul: cambiar menú, rosa: reportar comentario.



```
.botonCambiarMenu{
  display: flex;
  width: 14%;
  height: 100%;
  margin: 0.5%;
  padding: 0.5%;
  font-size: 1vw;
  color: var(--button-colour);
  transition: all 0.3s ease-in-out;
  background: var(--button);
  border: solid 1px var(--button-border);
  box-shadow: inset 0 0 0 2px #ffffff;
  margin-right: 1%;
  font-weight: bold;
  color: white;
  --button: hsl(240, 100%, 49%);
  --button-hover: hsl(240, 100%, 50%);
  --button-active: hsl(231, 76%, 47%);
  --button-visited: hsl(240, 68%, 46%);
  --button-colour: #1e1e1e;
  --button-border: #1e1e1e;
  justify-content: center;
}
```

- Botones de cambio de menú: este tipo de botón es exclusivo del componente ver juego y su funcionalidad es cambiar el menú del juego.



```
.cajaBotones button {
  flex-grow: 1;
  cursor: pointer;
  position: relative;
  padding:
    0.5vw
    0.5vw
    0.5vw;
  border: 0.1vw solid black;
  color: white;
  background-color: gray;
  font-size: 1vw;
  text-transform: lowercase;
  text-shadow: rgba(0, 0, 0, 0.1) 2px 2px;
  transition: flex-grow 250ms cubic-bezier(0.215, 0.61, 0.355, 1);
}

.cajaBotones button + button {
  border-left: v0.1vw solid black;
  margin-left: 0.1vw;
}
```

## 5.4.2. Errores y estado.

Se utiliza el siguiente diseño para mostrar errores a la hora de introducir datos o a la hora de confirmar la inserción de datos.

- Errores:

**Escribir comentario:**

El título del comentario no puede estar vacío

La nota del comentario no puede estar vacía ni superar el 10 ni ser menor de 1

La explicación del comentario no puede estar vacío

```
.error{
  font-size: 1.2vw;
  color: red;
  font-weight: bold;
}
```

- Confirmación:

El comentario ha sido creado con éxito

Escribir comentario:

Enviar comentario

```
.estado{
  font-size: 3vh;
  color: green;
  font-weight: bold;
}
```

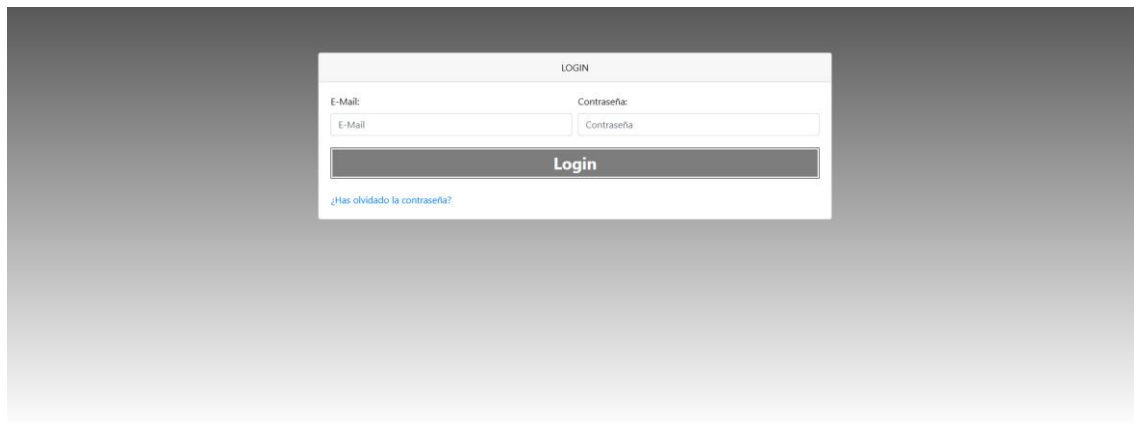
### 5.4.3. Logo.

El logo utilizado es un logo simple pero funcional está formado por el nombre de la página y un símbolo a su vez formado por un fantasma de pixeles y un cuadrado.



## 5.4.4. Fondo de pantalla.

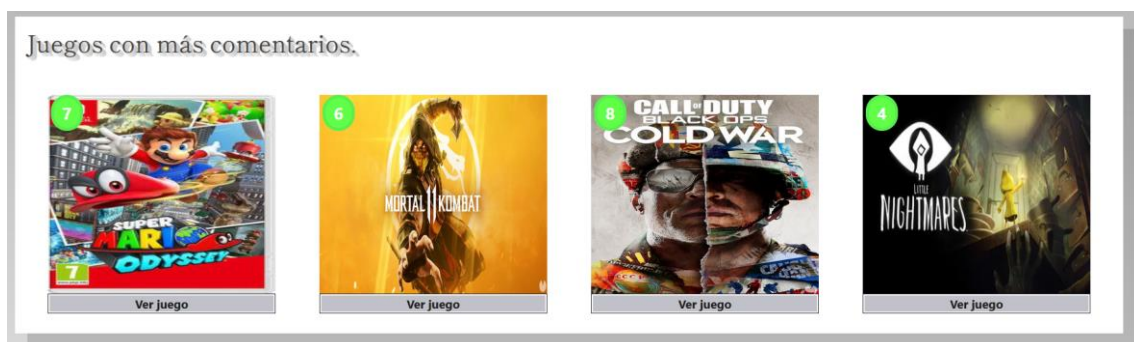
Se utiliza un degradado gris de fondo de pantalla en toda la web.



```
padding: 4%;
background: -moz-linear-gradient(top, rgba(0,0,0,0.65) 0%, rgba(0,0,0,0) 100%); /* FF3.6-15 */
background: -webkit-linear-gradient(top, rgba(0,0,0,0.65) 0%, rgba(0,0,0,0) 100%); /* Chrome10-25,Safari5.1-6 */
background: linear-gradient(to bottom, rgba(0,0,0,0.65) 0%, rgba(0,0,0,0) 100%); /* W3C, IE10+, FF16+, Chrome26+, Opera12+, Safari7+ */
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#a6000000', endColorstr='#00000000', GradientType=0 );
min-width: 100%;
min-height: 77vh;
```

## 5.4.5. Caja para juegos en inicio.

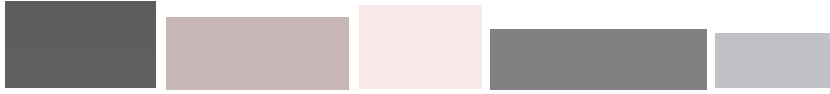
Utilizo una caja con fondo blanco. Esta caja tiene efecto hover para mostrar los juegos en el inicio.



```
.juegosMasComentarios{
margin-top: 1vw;
width: 100%;
height: 55vh;
margin-bottom: 5%;
position: relative;
padding: 1%;
background: white;
border-radius: 0% 0% 0% 0% / 0% 0% 0% 0% ;
color: white;
box-shadow: 20px 20px rgba(0,0,0,.15);
transition: all .4s ease;
}
```

### 5.4.6. Colores.

Utilizo grises, blancos y negros para los elementos de la web. Utilizo estos colores para darle el “toque” retro que quiero que tenga la web.



### 5.4.7. Tipo de letra.

Para los títulos utilizo el siguiente tipo de letra:

```
font-family: 'Shippori Mincho B1', serif;
```

El tráiler del juego mejor valorado por los usuarios.

Y para el título de la página utilizo el siguiente tipo:

```
font-family: neon;
```



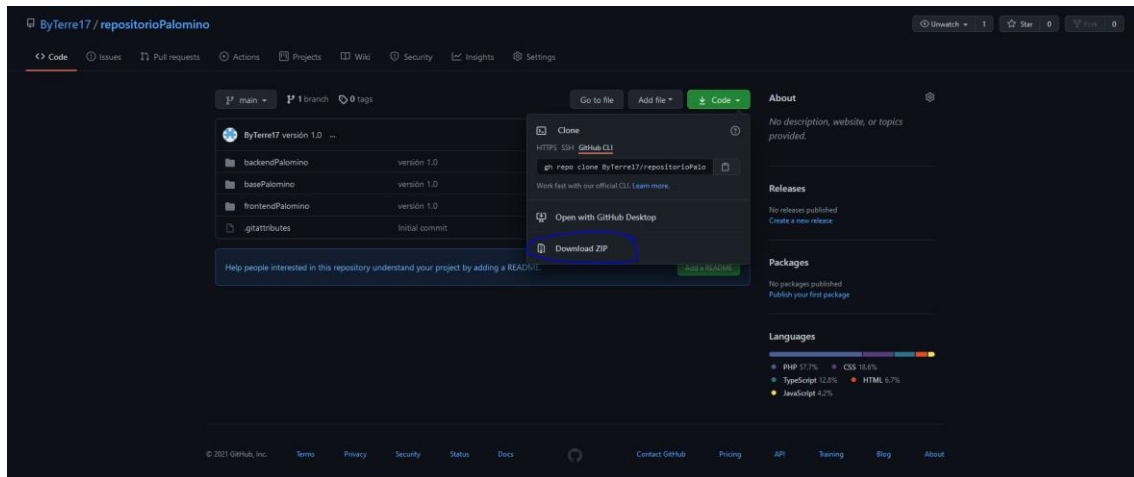
## 6.Procedimientos de instalación y prueba.

A continuación, se redacta un tutorial del proceso de instalación del proyecto, **es importante eliminar cualquier versión antigua eso incluye backend, Frontend y base de datos, en caso de no realizar esta eliminación previa es bastante probable que el proyecto no se ejecute correctamente.**

### 6.1.Primer paso.

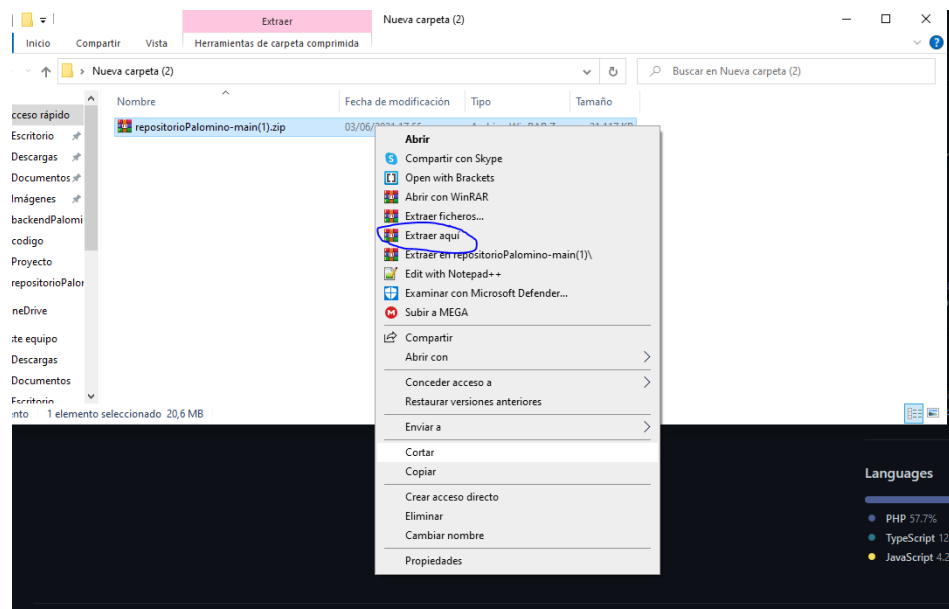
Dirigirse a la siguiente ubicación y descargar todo el proyecto.

<https://github.com/ByTerre17/repositorioPalomino>



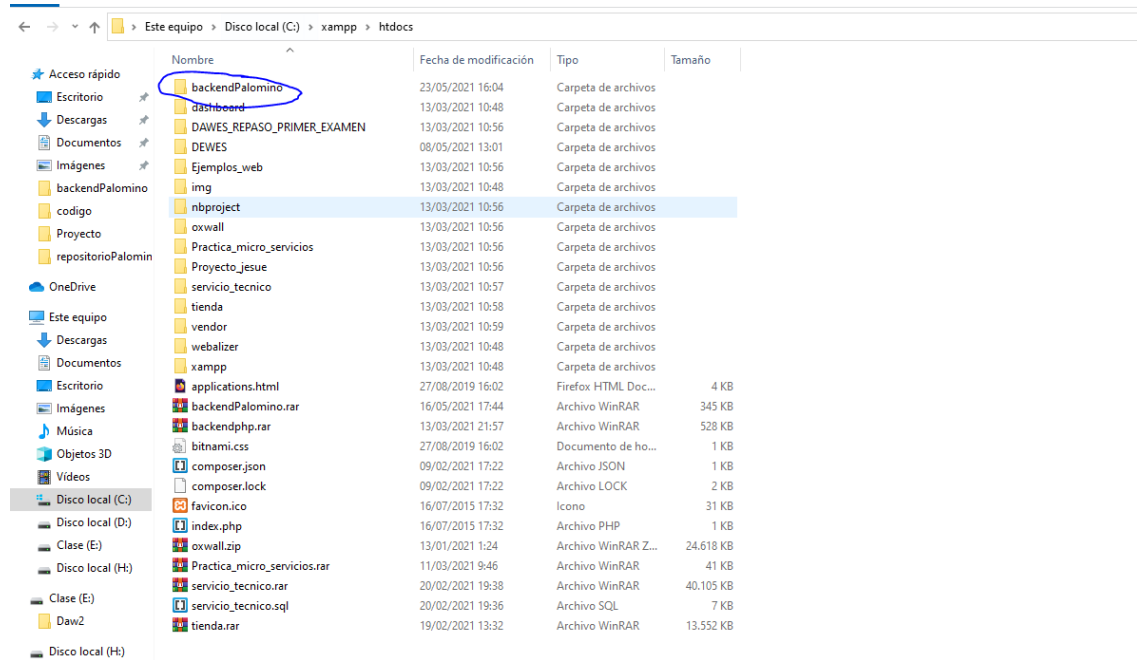
## 6.2.Segundo paso.

Descomprimir el zip descargado en el anterior paso en una carpeta.



## 6.3.Tercer paso.

Del resultado de la extracción del paso anterior copiaremos la carpeta backendPalomino y la pegaremos en la carpeta htdocs.



## 6.4. Cuarto paso.

Nos meteremos en phpmyadmin e importaremos el scriptPalomino que se encuentra en la basePalomino.

Importando al servidor actual

**Archivo a importar:**

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.  
A compressed file's name must end in `[format].[compression]`. Example: `.sql.zip`

Buscar en su ordenador: [Examinar...](#) scriptPalomino.sql (Máximo: 40MB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo: utf8

**Importación parcial:**

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be a good way to import large files, however it can break transactions.)

Omitir esta cantidad de consultas (en SQL) desde la primera: 0

**Otras opciones:**

☒ Habilite la revisión de las claves foráneas

**Formato:**

SQL

**Opciones específicas al formato:**

Modalidad SQL compatible: NONE

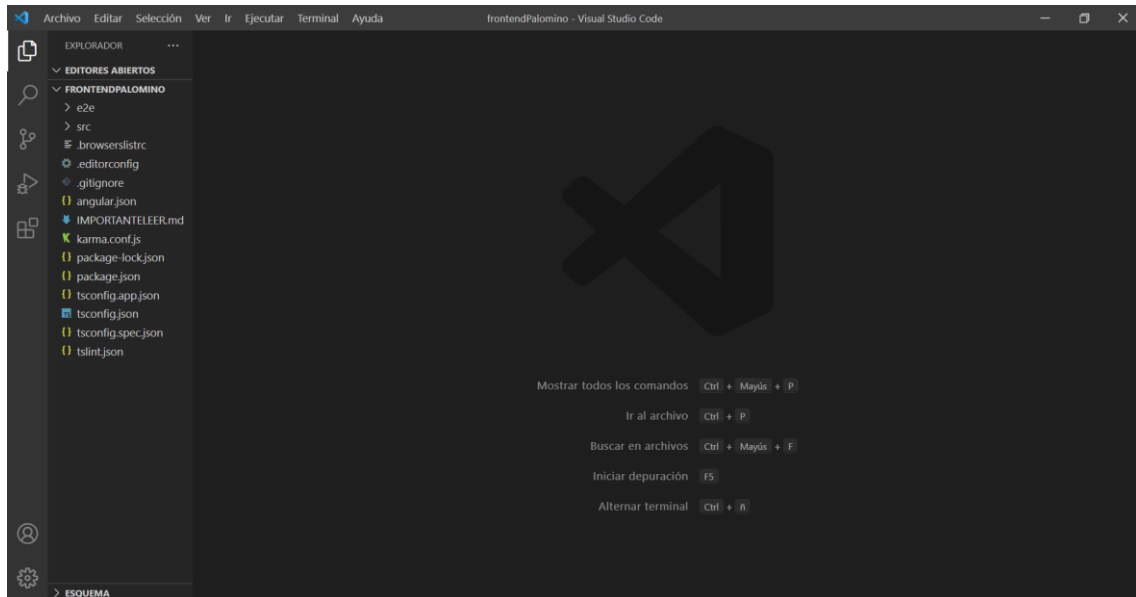
☒ No utilizar AUTO\_INCREMENT con el valor 0

[Continuar](#)

Criticame

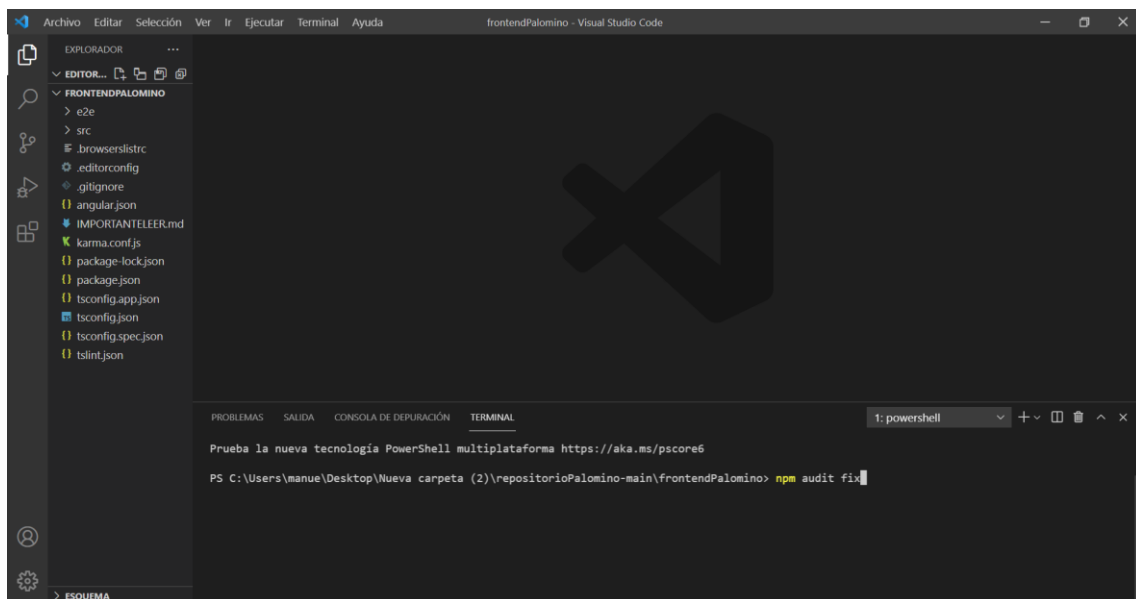
## 6.5. Quinto paso.

Con la herramienta visual estudio abriremos como proyecto la carpeta llamada frontendPalomino.



## 6.6. Sexto paso.

Ejecutaremos el comando “npm Audit fix” en el terminal de visual estudio.





## 6.7.Séptimo paso.

Ejecutaremos el comando “npm run ng serve” en el terminal de visual estudio.

```

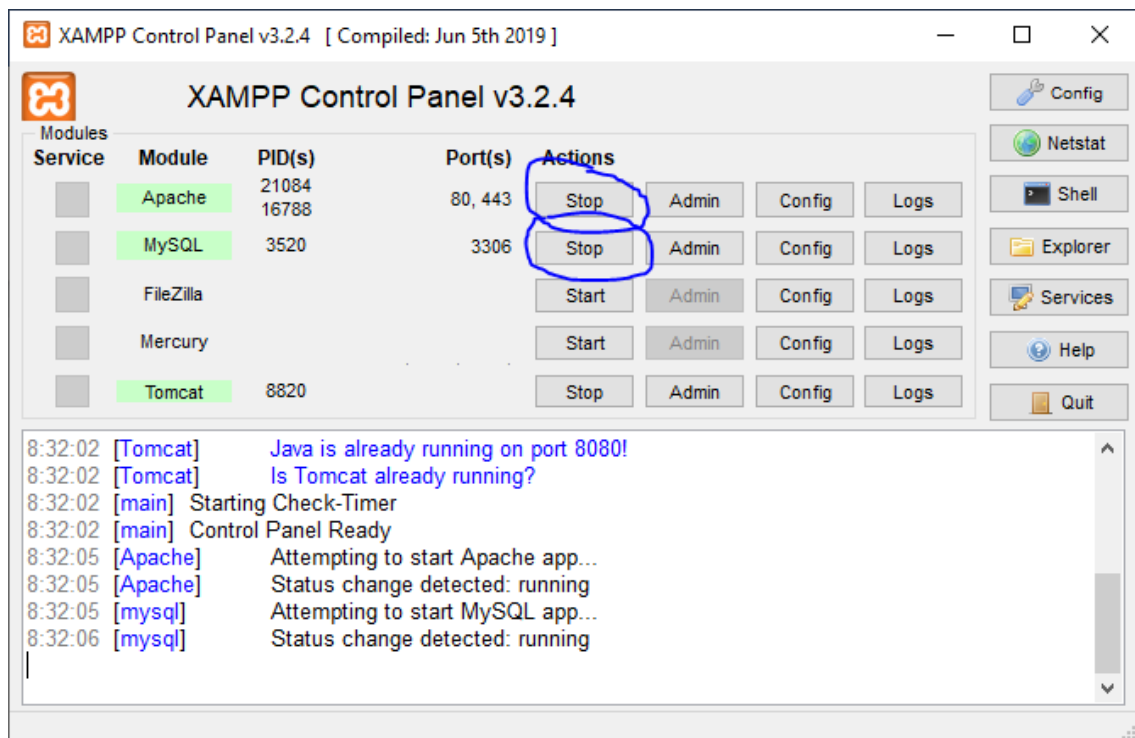
added 1648 packages from 1315 contributors in 48.793s

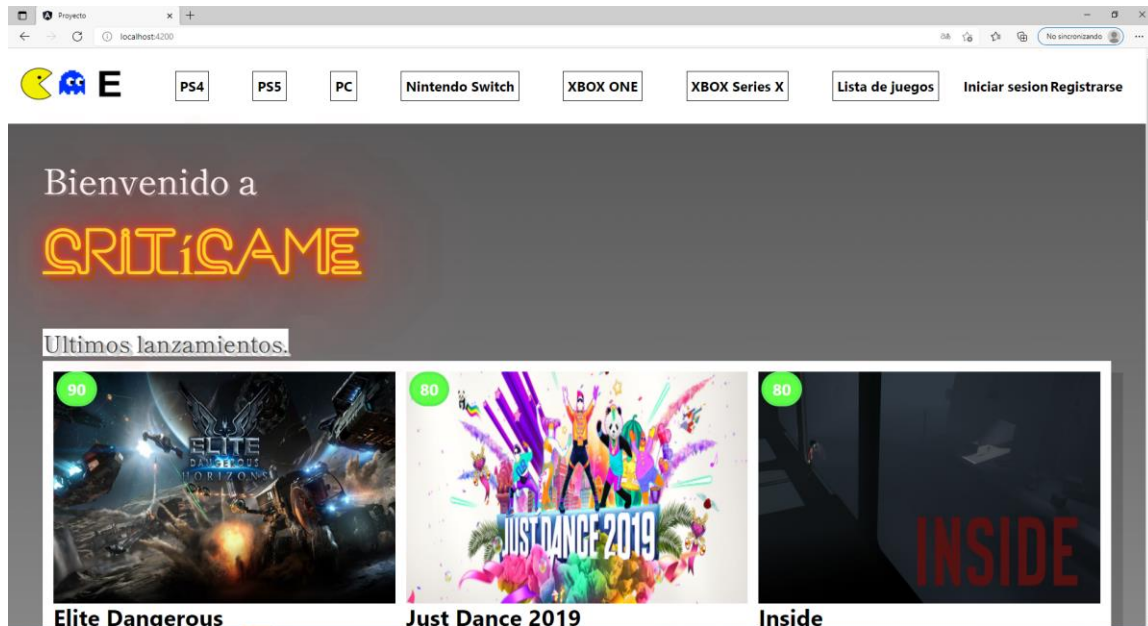
98 packages are looking for funding
  run `npm fund` for details

fixed 16 of 52 vulnerabilities in 1644 scanned packages
  3 vulnerabilities required manual review and could not be updated
  1 package update for 33 vulnerabilities involved breaking changes
  (use `npm audit fix --force` to install breaking changes; or refer to `npm audit` for steps to fix these manually)
PS C:\Users\manue\Desktop\Nueva carpeta (2)\repositorioPalomino-main\frontendPalomino> npm run ng serve
  
```

## 6.8.Octavo paso.

Ejecutaremos XAMPP y habilitaremos apache y MySQL, acto seguido nos dirigiremos a la siguiente dirección en nuestro navegador <http://localhost:4200/> y si hemos hecho todo correctamente el proyecto debería de estar ejecutándose.





## 7.Arquitectura del sistema y modelo de datos.

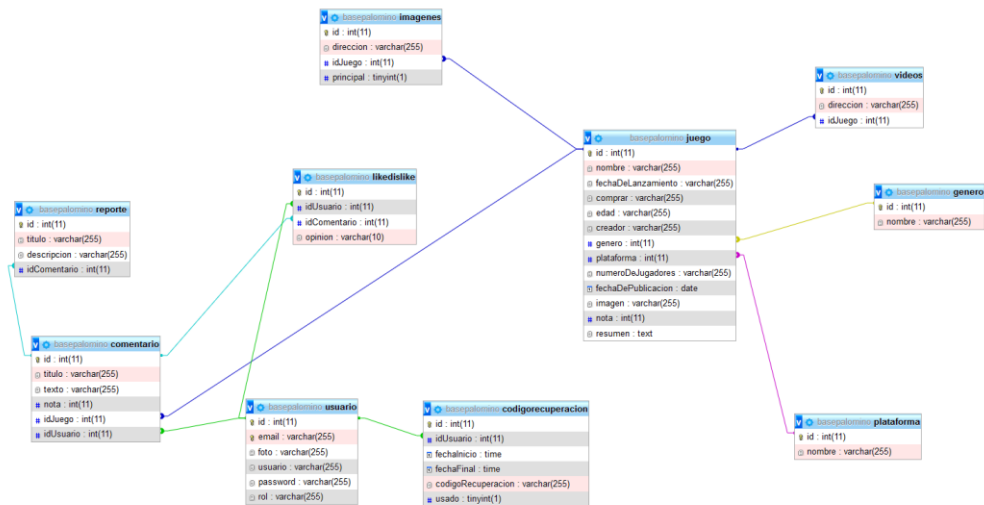
### 7.1.Arquitectura.

La arquitectura utilizada es la **cliente-servidor**:

La **arquitectura cliente-servidor** es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

## 7.2. Modelo de datos.

El modelo de datos utilizado es el siguiente:



En este modelo de datos interactúan los siguientes tipos de datos:

- Juego.
- Código de Recuperación.
- Género.
- Imágenes.
- Comentario.
- Like/dislike.
- Plataforma.
- Reporte.
- Usuario.
- Videos.

## 8. Mejoras futuras.

En este apartado hablaré sobre las mejoras que se realizarán en un futuro las cuales no se han podido implementar actualmente.

- Una mejor utilización de los datos de usuario, implementar una forma de que un usuario pueda visualizar los datos de otros usuarios y visualizar sus comentarios. El sistema avisará de que esos usuarios han subido un comentario nuevo. (Sistema de follow).
- Una forma más fácil de cambiar el rol de un usuario.
- Un sistema de baneos que aplique distintas restricciones a los usuarios que no cumplan las normas de la web.
- Más cantidad de filtros en la lista de juegos.