

Cumplimentación proyecto 1

USUARIOS PARA ACCEDER EN EL LOGIN DE LA PAGINA:

Github.com/tonimanuel/proyect

Administrador CONTRA: Alumno2021.

antonio CONTRA: Alumno2021.

• Antecedentes

Mi proyecto final trata de de una página web sobre una tienda online en el que tendremos stock, precio, carrito (solo registrados), añadir comentarios (solo registrados), darnos de alta, recuperar contraseña ver productos. Esta página web va dirigida a usuarios que estén interesados en comprar ya sea un videojuego o videoconsola o simplemente informarse.

Se han utilizado distintos lenguajes de programación, como pueden ser PHP y JavaScript para su desarrollo, y para su diseño se ha hecho uso de visual estudio y de estilos CSS.

• Requisitos.

- La aplicación tendrá una parte visible para todo el público que acceda a esta. Aunque no esté registrado, necesitaremos controlar el acceso a diferentes páginas según el rol del usuario.
- Carrito para guardar juegos y consolas, pero solo a registrados.
- Comentarios a usuarios logueados en cada producto ya sea videojuego o consola. Y el único que podrá eliminarlos será el ADMIN.
- Niveles de usuario: tendremos el usuario Administrador que solo podre acceder a él solamente yo, y usuario normal que podrá tanto darse de alta como recuperar su contraseña.
- Administrador: El administrador puede añadir juegos, videoconsolas modificar descripción, id, precio, eliminar un comentario de un usuario para hacer esto debemos de acceder al panel de administración que nuestro más arriba y dentro tenemos cuatro opciones: videoconsolas, videojuegos, panel de usuarios y productos enel que podemos acceder a cada uno de ellos ya sea para añadir, modificar, eliminar.
- Diseño responsive.

- Los usuarios no registrados podrán ver artículos pero no añadir a la cesta ni hacer un comentario.
- Validación de campos en formularios.
 - DNI
 - Teléfono
 - Mail
 -

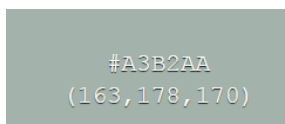
• Guía de estilos



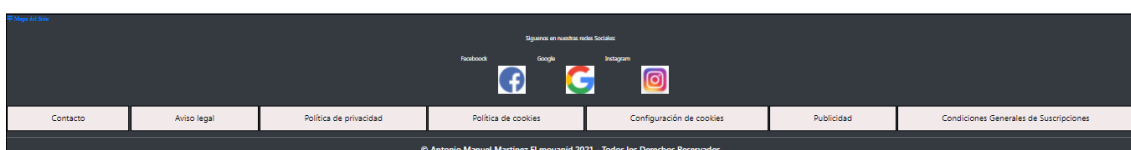
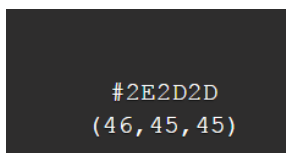
El logotipo viene de un equipo diseñado por mí de deportes electrónicos. Representa un ninja japonés que destaca por sus símbolos de guerra y que va asociado a los botones en cuanto a estilo.

Está compuesto de dos colores:

Este es el primario que destaca por ser un color claro y para mi gusto transmite elegancia.



Este sería mi color secundario que sería un color más oscuro y menos llamativo por eso es menos empleado a mi gusto. Uso también colores de bootstrap para el header y el footer.



FOOTER

- el header y el footer le doy una tonalidad gris oscura empleada con bootstrap. También en cuanto dejamos el cursor sobre un enlace cambia el color de este y si pulsamos podemos acceder al mismo. Tenemos dos

carruseles que son recuadros que van cambiando de imagen cada cierto tiempo automáticamente.

Plataformas Disponibles

- En cuanto a la tipografía pretendo destacar más los títulos poniendo el color blanco y todo lo que vaya fuera del header y footer ira en blanco.



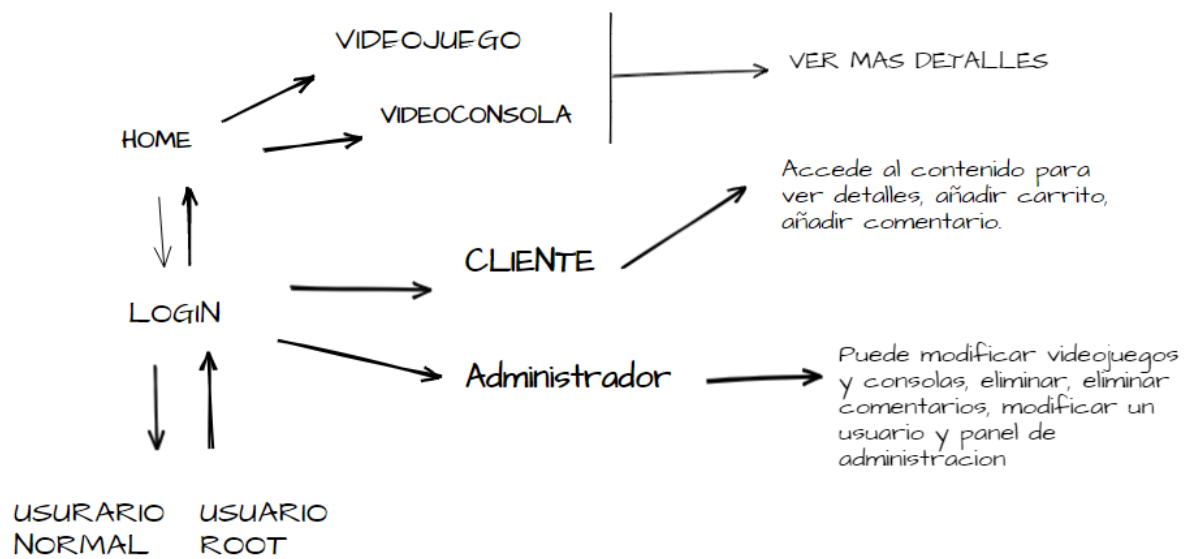
Estos serian los botones de loguearse y registrarse que lo tengo asimilado a los botones de nueva consola y videojuego y los botones de modificar y eliminar del mismo.



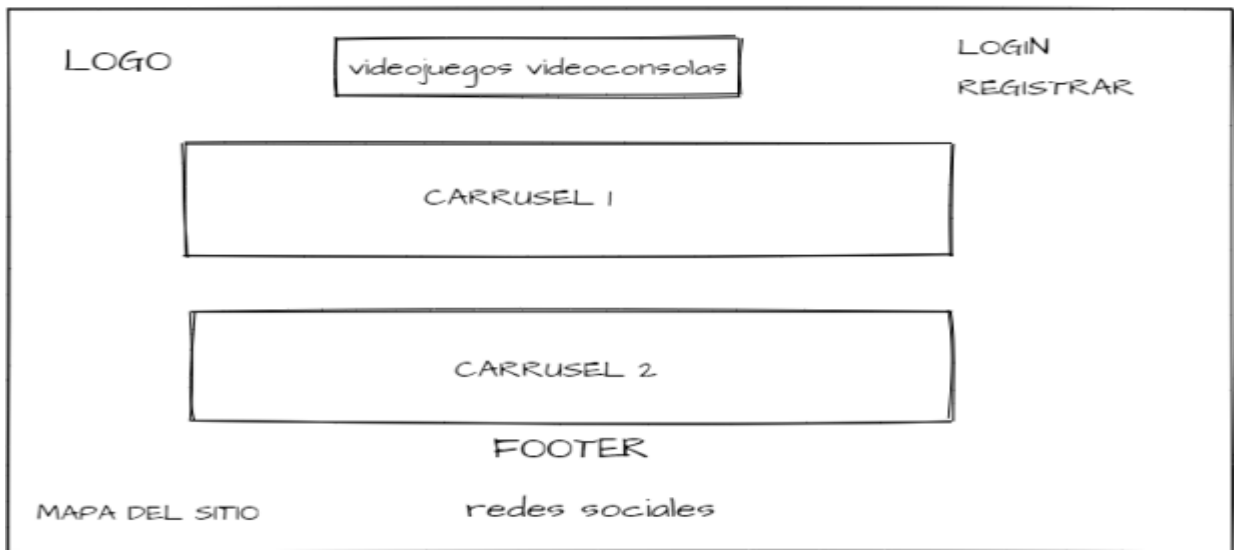
Para los botones de añadir consola, eliminar y modificar le he dado este estilo que se asimila al logo y tiene temática de guerra con un fondo grisáceo militar.

• ANÁLISIS Y DISEÑO WEB

MAPA WEB: Según el tipo de usuario (ROOT O USUARIONORMAL), tendrá acceso a unas páginas u otras controlando el acceso y cambiando la barra de navegación.

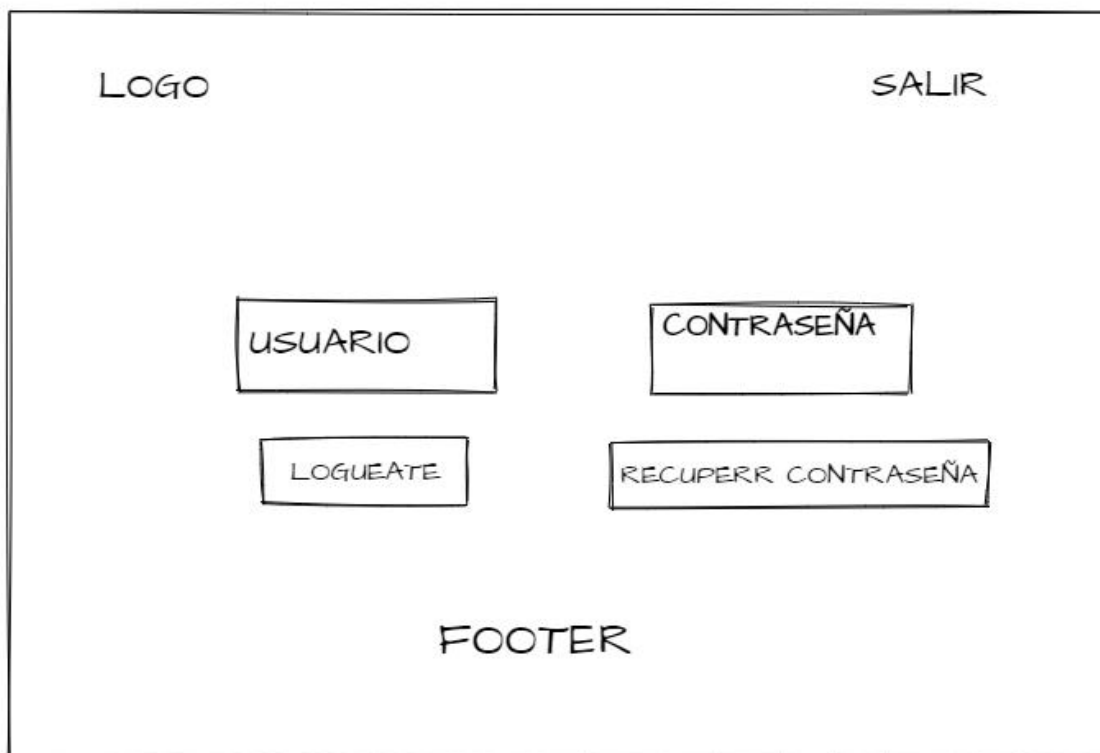


Prototipo web y boceto de la estructura



Este sería el home. Con los dos carruseles y el login y registrarse.

Este sería el login



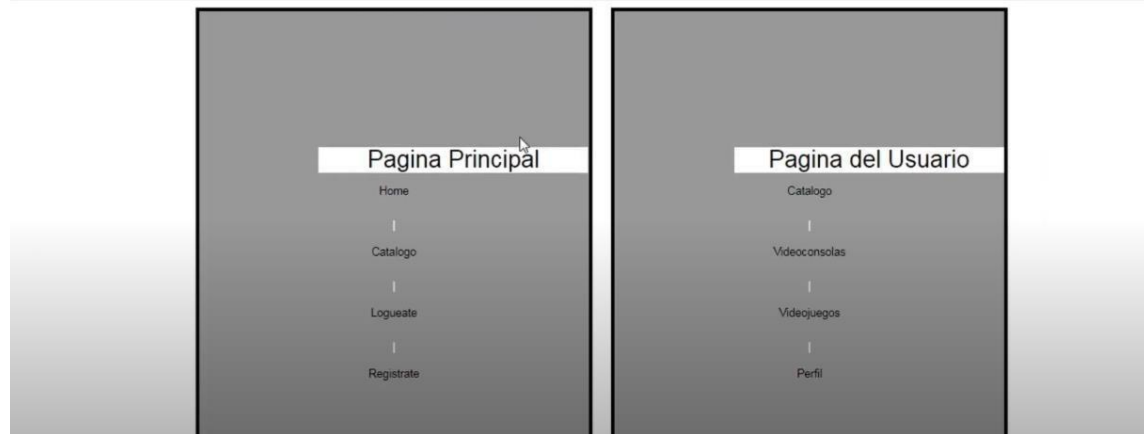
Esta seria la interfaz del apartado videojuegos o videoconsolas sin loguearse.

A hand-drawn wireframe of a web interface for a video games section. At the top, there is a navigation bar with five buttons: "LOGO", "VIDEOJUEGOS" (highlighted in black), "VIDEOCONSOLA", "LOGIN", and "REGISTRAR". Below the navigation bar, there is a grid of six empty square boxes arranged in two rows of three. At the bottom of the page, the word "FOOTER" is centered.

A hand-drawn wireframe of a new user registration form. At the top left is a "LOGO" button, and at the top right is a "salir" button. In the center, the text "Registro usuario nuevo" is displayed. Below this text, there are six input fields arranged in a 3x2 grid, labeled "campo 1", "campo 2", "campo 3", "campo 4", "campo 5", and "campo 6". At the bottom of the page, the word "FOOTER" is centered.

Formulario de registro para un nuevo usuario.

Este sería mi mapa del sitio que lo tengo añadido en el home de mi página web.



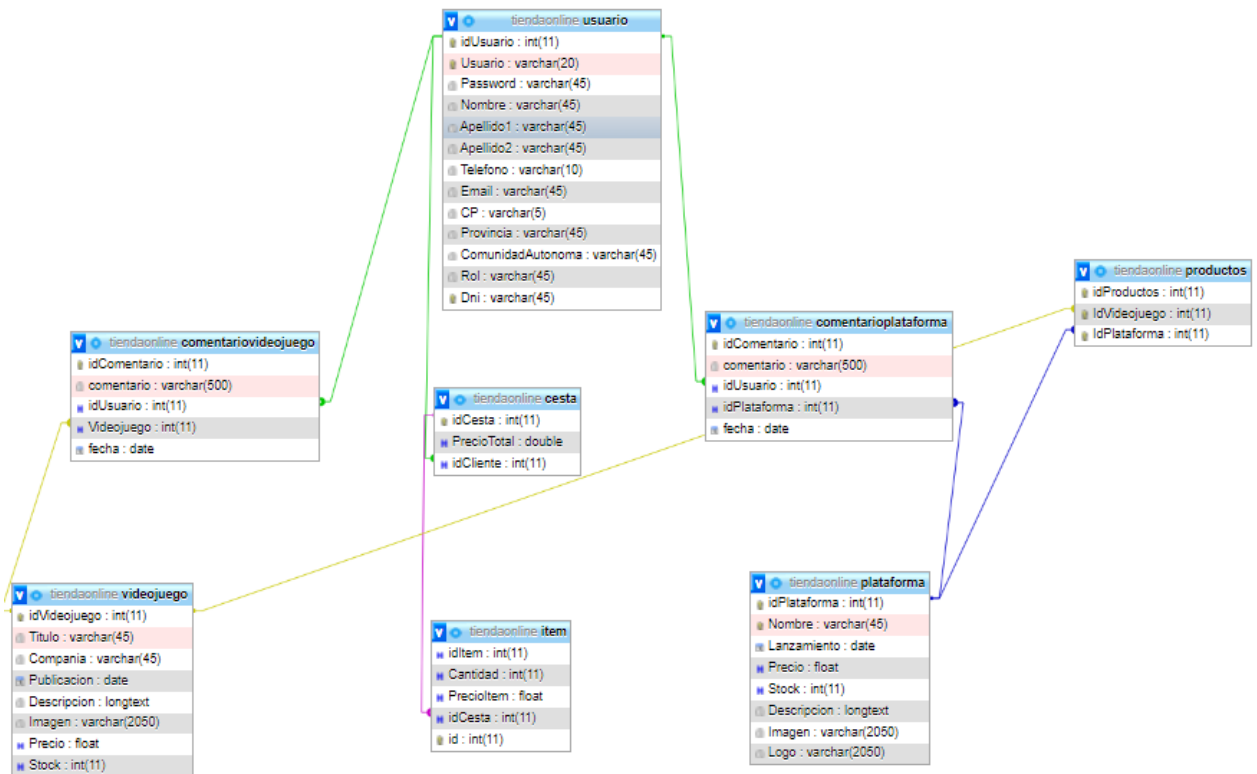
Planificación de tareas

TÍTULO DE LA TAREA	FECHA
Análisis e inicio del proyecto	
Investigación	02/10/21
configuración base de datos	29/09/21
Comienzo a escribir el script SQL	22/09/21
Conexión a la BD	01/10/21
creación Login	02/10/21
creación Login	03/10/21
creación login	04/10/21
creación olvidar contraseña	05/10/21
creación olvidar contraseña	06/10/21
creación olvidar contraseña	07/10/21
Comprobar usuario	15/10/21
Comprobar usuario	16/10/21
Comprobar usuario	17/10/21
validado formulario	18/10/21
validado formulario	19/10/21
validado formulario	20/10/21
Comprobar usuario	21/10/21
ingresar Usuario nuevo	22/10/21
ingresar usuario nuevo	23/10/21
Administrador usuario	24/10/21
cesta	25/10/21

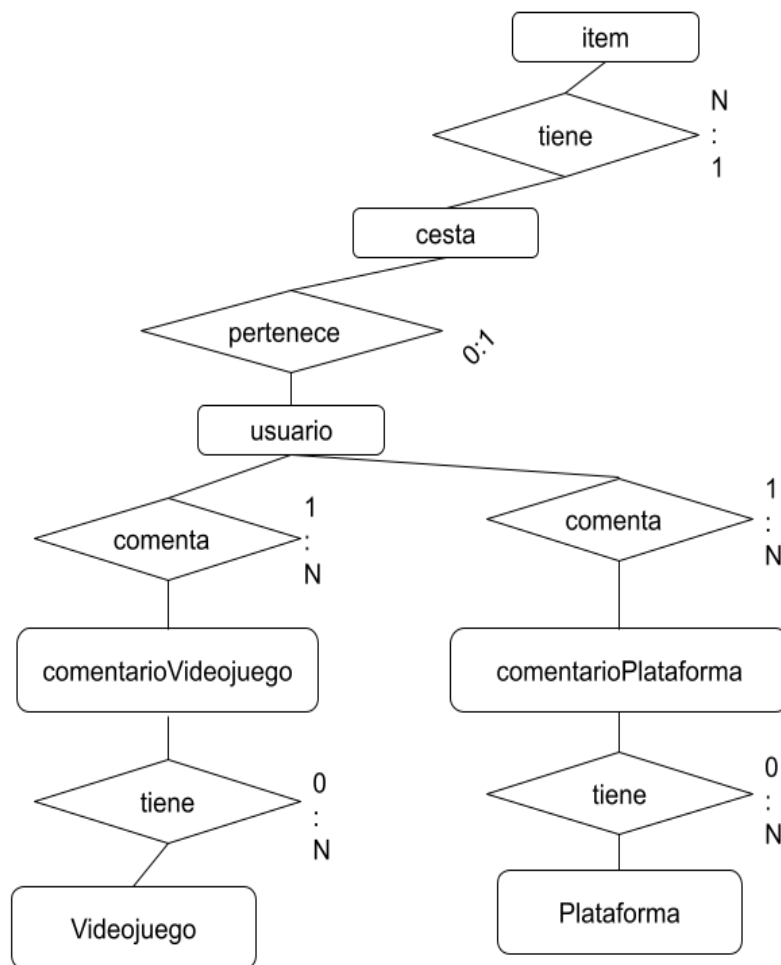
	26/10/21
Administrador usuario	27/10/21
Administrador producto	28/10/21
Modificar usuario	29/10/21
Modificar usuario	30/10/21
Modificar usuario	31/10/21
Administrador producto	01/11/21
Administrador producto	02/11/21
footer con estilos	03/11/21
footer con estilos	04/11/21

Diagrama relacional

Este sería el modelo relacional.



Este sería el diagrama entidad- relación.



ANÁLISIS Y DISEÑO WEB. Script creación BBDD

Tabla CESTA: La llave primaria sería el idCesta, y la llave foránea IDcliente y el campo preciototal es de tipo double que lo utilizamos para sacar al usuario el precio total del carrito que `pasa por el item antes.

```
CREATE TABLE `cesta` (  
  `idCesta` int(11) NOT NULL,  
  `PrecioTotal` double DEFAULT '0',  
  `idCliente` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

El alter table sería para añadir la llave foránea y primaria.

```
ALTER TABLE `cesta`  
  ADD PRIMARY KEY (`idCesta`),  
  ADD KEY `Cesta` (`idCliente`);
```

Tabla COMENTARIOPLATAFORMA: La llave primaria seria el idComentario, y tendríamos las dos llaves foráneas idUsuario que nos viene de la tabla usuario e idPlataforma que sale de la tabla plataforma.

```
CREATE TABLE `comentarioplataforma` (  
  `idComentario` int(11) NOT NULL,  
  `comentario` varchar(500) NOT NULL,  
  `idUsuario` int(11) NOT NULL,  
  `idPlataforma` int(11) NOT NULL,  
  `fecha` date DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
ALTER TABLE `comentarioplataforma`  
  ADD PRIMARY KEY (`idComentario`),  
  ADD KEY `idUsuario` (`idUsuario`),  
  ADD KEY `idPlataforma` (`idPlataforma`);
```

Tabla COMENTARIOVIDEOJUEGO: La estructura seria la misma que la anterior solo que una se usa para las consolas y otra para los videojuegos.

```
CREATE TABLE `comentariovideojuego` (  
  `idComentario` int(11) NOT NULL,  
  `comentario` varchar(500) NOT NULL,  
  `idUsuario` int(11) NOT NULL,  
  `Videojuego` int(11) NOT NULL,  
  `fecha` date DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

La llave primaria seria idComentario y las dos foráneas serian idUsuario y videojuego como en el anterior caso.

```
ALTER TABLE `comentariovideojuego`  
  ADD PRIMARY KEY (`idComentario`),  
  ADD KEY `idUsuario` (`idUsuario`),  
  ADD KEY `Videojuego` (`Videojuego`);
```

Tabla ITEM: Aquí tendríamos estos campos ITEM tabla nos serviría para clasificar ya sea videoconsola o videojuego con 0 o 1, para diferenciar entre uno u otro y ya de esta tabla pasa a la siguiente que es la de cesta y ahí es donde se mostraría.

```
CREATE TABLE `item` (  
  `idItem` int(11) NOT NULL,  
  `Cantidad` int(11) NOT NULL,  
  `PrecioItem` float DEFAULT NULL,  
  `idCesta` int(11) NOT NULL,  
  `id` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

La llave primaria sería el ID y La llave foránea el idCESTA.

```
ALTER TABLE `item`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `ItemPerteneceACesta` (`idCesta`);
```

Tabla PLATAFORMA: Esta sería la tabla para añadir las consolas con interfaz gráfica(imagen).

```
CREATE TABLE `plataforma` (  
  `idPlataforma` int(11) NOT NULL,  
  `Nombre` varchar(45) COLLATE utf8_bin NOT NULL,  
  `Lanzamiento` date NOT NULL,  
  `Precio` float NOT NULL,  
  `Stock` int(11) NOT NULL,  
  `Descripcion` longtext COLLATE utf8_bin NOT NULL,  
  `Imagen` varchar(2050) COLLATE utf8_bin DEFAULT NULL COMMENT 'Imagen de la plataforma',  
  `Logo` varchar(2050) COLLATE utf8_bin NOT NULL COMMENT 'Logo de la consola'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

La llave primaria sería idPlataforma y la llave foránea Nombre.

```
ALTER TABLE `plataforma`  
  ADD PRIMARY KEY (`idPlataforma`),  
  ADD UNIQUE KEY `Nombre_UNIQUE` (`Nombre`);
```

Tabla usuario: Esta sería la tabla de usuario con sus campos que tendría todos estos campos y para registrar un usuario debe de añadir todos los campos.

```
CREATE TABLE `usuario` (  
  `idUsuario` int(11) NOT NULL,
```

```

`Usuario` varchar(20) COLLATE utf8_bin NOT NULL COMMENT 'El usuario debe ser
único en la Base de Datos. Además debe tener una longitud mínima de 6
caracteres.',
`Password` varchar(45) COLLATE utf8_bin NOT NULL COMMENT 'La contraseña no
puede ser nula.\nTiene que tener una longitud mínima de 8 caracteres, contener
números y algún carácter especial de la siguiente lista (!,@,#,$,%,&,?,*,?,'),
`Nombre` varchar(45) COLLATE utf8_bin NOT NULL,
`Apellido1` varchar(45) COLLATE utf8_bin NOT NULL,
`Apellido2` varchar(45) COLLATE utf8_bin DEFAULT NULL,
`Telefono` varchar(10) COLLATE utf8_bin NOT NULL COMMENT 'Se comprobará que
solo puedan ser teléfonos nacionales',
`Email` varchar(45) COLLATE utf8_bin NOT NULL COMMENT 'Se comprobará que es un
email válido',
`CP` varchar(5) COLLATE utf8_bin NOT NULL,
`Provincia` varchar(45) COLLATE utf8_bin NOT NULL COMMENT 'Se comprobará que la
provincia introducida concuerda con el CP introducido',
`ComunidadAutonoma` varchar(45) COLLATE utf8_bin NOT NULL COMMENT 'Se
comprobará que la provincia seleccionada pertenece a la comunidad autónoma
correspondiente',
`Rol` varchar(45) COLLATE utf8_bin NOT NULL,
`Dni` varchar(45) COLLATE utf8_bin DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

La clave primaria sería el idUsuario, y la clave foránea el usuario.

```

ALTER TABLE `usuario`
  ADD PRIMARY KEY (`idUsuario`),
  ADD UNIQUE KEY `idUsuario_UNIQUE` (`idUsuario`),
  ADD UNIQUE KEY `Usuario_UNIQUE` (`Usuario`),
  ADD UNIQUE KEY `Dni` (`Dni`);

```

Tabla VIDEOJUEGO: Esta sería la tabla de videojuego donde añadimos los videojuegos nuevos y lo mostramos con imágenes.

```

CREATE TABLE `videojuego` (
  `idVideojuego` int(11) NOT NULL,
  `Titulo` varchar(45) COLLATE utf8_bin NOT NULL,
  `Compania` varchar(45) COLLATE utf8_bin NOT NULL,
  `Publicacion` date NOT NULL,
  `Descripcion` longtext COLLATE utf8_bin NOT NULL,
  `Imagen` varchar(2050) COLLATE utf8_bin DEFAULT NULL COMMENT 'Url de la imagen
de la carátula del juego',
  `Precio` float DEFAULT NULL,
  `Stock` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

```

Tendríamos la llave primaria que es idVideojuego.

```

ALTER TABLE `videojuego`
  ADD PRIMARY KEY (`idVideojuego`);

```

Consultas

Esta seria una consulta de tipo **inserción** para añadir datos en las tablas.

```
INSERT INTO `comentariovideojuego` (`idComentario`, `comentario`, `idUsuario`,  
`Videojuego`, `fecha`) VALUES
```

Esta seria la consulta de **eliminación** para eliminar determinados datos de una tabla utilizamos el delete from, es una consulta muy simple que le pasamos el parámetro para saber que eliminar.

```
DELETE FROM plataforma WHERE `plataforma`.`idPlataforma` = 12"
```

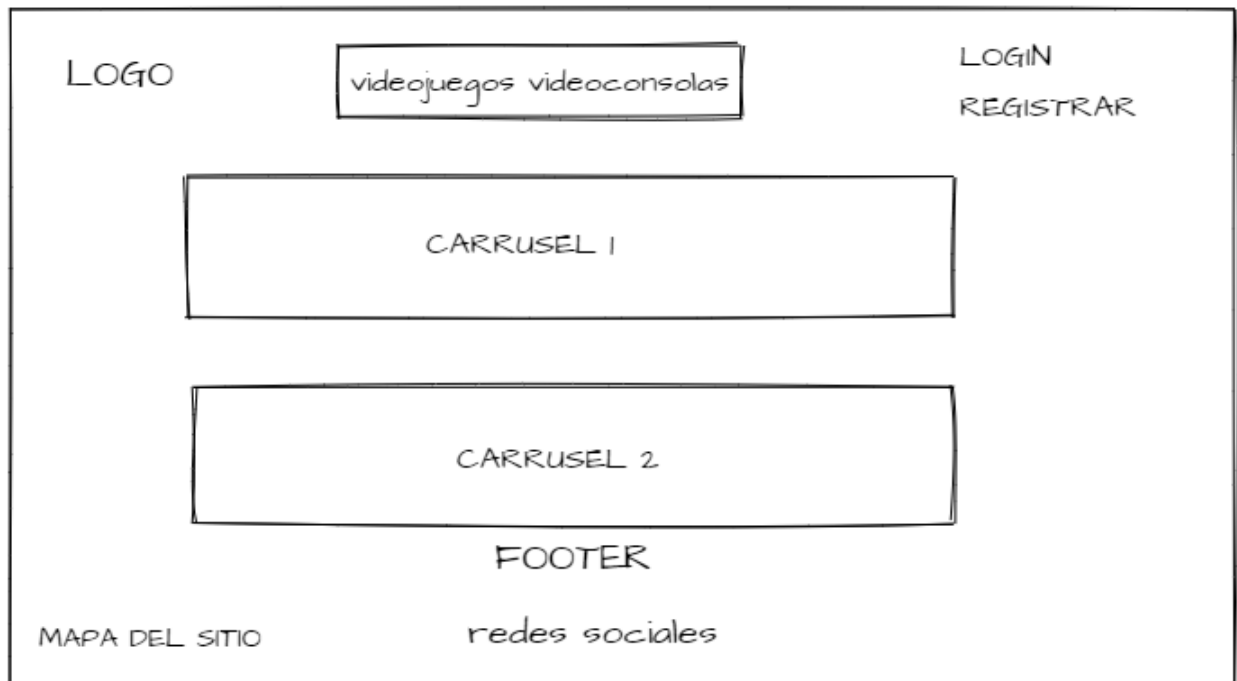
También tenemos consulta del tipo **"UPDATE"** para modificar el estado de una tabla por ejemplo en este caso estamos modificando la tabla usuario.

```
"UPDATE `tiendaonline`.`usuario` SET `Usuario` = '$usuario', `Password` =  
'$password', `Nombre` = '$nombre', `Apellido1` = '$apellido1', `Apellido2` =  
'$apellido2', `Telefono` = '$telefono', `Email` = '$email', `CP` = '$cp',  
`Provincia` = '$provincia', `ComunidadAutonoma` = '$comunidadautonoma', `Dni` =  
'$dni',
```

Consultas más específicas **"select"** pasando por parámetros datos únicos para identificar elementos de una tabla específica.

```
consulta = "select * from item where idCesta='$idCesta';
```

- **Diseño de la interface**



El sitio web consta de cuatro partes fundamentales. Esta estructura es común en todas las páginas.

Cabecera: Contiene el logo de la empresa. En este apartado encontramos, a la derecha, el nombre del usuario que estuviera registrado o la palabra 'login' para los usuarios no logueados y también el botón de registrarse.

Barra de navegación: Contiene los enlaces a las páginas: home (logo), El apartado de videojuegos y videoconsolas login/logout. Dependiendo del rol del usuario logueado, contendrá, además, los enlaces hacia las demás páginas. En el caso de cliente: videojuego y videoconsola .productos. El carrito aparece una vez logueado

Contenido central: Irá variando ya que es un carrusel y va cambiando de producto cada x tiempo.. Es la única parte "dinámica" de la página.

Footer: Se encuentra al final de la página, debajo del contenido central. En él podemos encontrar datos como el contacto, redes sociales, mapa del sitio.

• IMPLEMENTACIÓN

1. **Control de vista según usuario:** Para que haya una página reconocible debe tener distinción entre roles de usuario, se implementa un control de vista para cada de usuario.

Para hacer su correcto funcionamiento he empleado php y HTML.

```
$existeUsuario = consultaLogin($conexion,$usuario,$password);  
  
$existeSoloUsuario=consultaUsuario($conexion, $usuario);  
if(mysqli_num_rows($existeUsuario)==1){
```

2. **Carrito:** El usuario podrá añadir productor y videojuegos de nuestra pagina a su cesta el único requisito es que deberá de estar registrado

Para su funcionamiento hemos empleado php.

```
function sacarId($item){  
    function sacarTipo($item){  
f (isset($_POST['anadirConsola'])) {  
    $long= strlen(strlen($_POST['idConsola']));  
    $idItemIndicador = str_pad($_POST['idConsola'], $lon
```

3. **Añadir comentarios:** es una herramienta para que el usuario normal pueda dejar su opinión acerca del producto o un simple comentario.

Para su correcto funcionamiento he empleado php.

```
if(isset($_SESSION['Usuario']))){  
    ?>  
    <div class="enviar-comentario">  
        <h2>Deja un comentario</h2>  
        <form method="POST">  
            <!-- <input type="text" name="comentario"> -->  
            <div class="form-group">
```

4. **Diseño responsive:** lo utilizamos para que sea posible navegar en dispositivos móviles. La principal necesidad que existe para hacer un diseño «responsive», es porque se pierden bastantes visitantes, al no tener el site adaptable a todos.

Para su correcto funcionamiento he utilizado Bootstrap.

```
<script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scrip  
t
```

5. **Carrito:** lo utilizamos para que el usuario guarde ya sea videojuegos o videoconsolas y sepa cual es el precio total de lo que desea. Tendremos un apartado para el carrito para acceder directamente.

Para su funcionamiento he utilizado lenguaje html y php sobretodo.

```
function sacarId($item){  
    $longitud=strlen(strval($item));  
    $tip=substr(strval($item),0,($longitud-1));  
    $resultado=intval($tip);  
    return $resultado;  
}
```

```
$resul=insertarJuegoCarrito($conexion,$_SESSION['idUsuario'],$_SESSION['idUsuario'  
''],$idItem,$_POST['cantidad'],$_P
```

Para mostrar juego o consola en el carrito.

```
$cesta=consultaPrecioCesta($conexion,$_SESSION['idUsuario']);  
if (mysqli_num_rows($cesta)) {  
    $cestaPre=mysqli_fetch_assoc($cesta);  
?>
```

Consulta para el precio final de la cesta

- **PRUEBAS**
- **Metodología de las pruebas**

Para realizar las pruebas he ido modificando en localhost primero con la base de datos y con el visual studio conforme iba modificando errores iba desplegándolo en el local y seguidamente de la máquina virtual debían, subiéndolo a github y comprobando que el despliegue se hace correctamente ya que en el anterior proyecto me estaba dando fallos bastantes apartados que no terminaban de desplegar y de salir como corresponde.

Durante el desarrollo, se ha comprobado la salida o entrada de datos mediante "echos" para imprimir antes de que llegaran al modelo.

En cuanto a las pruebas de los estilos he estado teniendo problemas con el navegador ya que cambiaba estilos desde el código pero al desplegarlo no se me actualizaba y luego me di cuenta que era problema del navegador que se quedaba guardado la cache y no lo mostraba, la solución es abrir una pestaña privada y actualizar la pagina.

• Validación de formularios

Para la validación de formularios se ha utilizado Javascript. Los formularios de la página son sencillos, por lo que los scripts son muy parecidos entre sí. Se le han asignado a cada uno de los campos una expresión regular determinada (acorde con el tipo de campo y los requisitos).

Después nos aseguramos de que el usuario ha metido un valor,

Llamamiento al id.

```
const usuario = document.getElementById('usuario');
const password = document.getElementById('password');
const nombre = document.getElementById('nombre');
$("#usuario_error").css("visibility", "hidden");
$("#password_error").css("visibility", "hidden");
$("#nombre_error").css("visibility", "hidden");
```

Aquí le asignamos la expresión regular.

```
const expresiones = {
  usuario: /^[a-zA-Z0-9\_\-]{4,16}$/ ,
  password: /^(?=.*{8,})?(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*\W).*$/,
  nombre: /^[a-zA-Z]{2,12}$/ ,
```

Esta seria las funciones para validar el usuario y la password que yo tengo una función por cada campo que tengo que validar

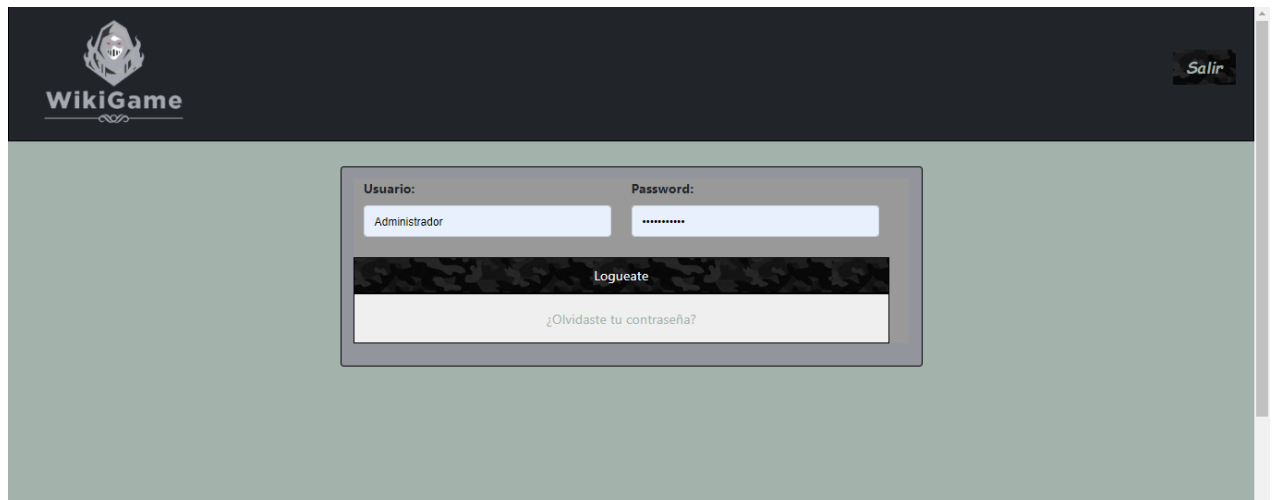
```
function validarUsuario(){
  if(expresiones.usuario.test(usuario.value)){
    document.getElementById('usuario_error').style.visibility = "hidden";
  }else{
    document.getElementById('usuario_error').style.visibility = "visible";
    document.getElementById('usuario_error').style.color = "red";
  }
}

function validarPassword(){
  if(expresiones.password.test(password.value)){
    document.getElementById('password_error').style.visibility = "hidden";
  }else{
    document.getElementById('password_error').style.visibility = "visible";
    document.getElementById('password_error').style.color = "red";
  }
}
```

Añadimos el oyente de eventos para que la validación se lleve a cabo.

```
usuario.addEventListener("keyup", validarUsuario);  
usuario.addEventListener("blur", validarUsuario);  
password.addEventListener("keyup", validarPassword);
```

Formulario Login



The screenshot shows a web application interface for 'WikiGame'. The header is dark grey with the 'WikiGame' logo on the left and a 'Salir' button on the right. The main content area has a light green background. In the center, there is a login form with a grey border. The form contains two input fields: 'Usuario:' with the text 'Administrador' and 'Password:' with masked characters. Below these fields is a dark button labeled 'Logueate'. At the bottom of the form is a link that says '¿Olvidaste tu contraseña?'.

▪ Herramientas

En cuanto a mí he utilizado las siguientes herramientas;

Microsoft Visual Studio: es un entorno de desarrollo integrado para Windows y se utiliza para programar en múltiples lenguajes de programación.

Xampp: software libre que consiste en la gestión de bases de datos Mysql.

Mamp: Se puede usar para ejecutar un concierto servidor web desde una Mac, pero se usa más comúnmente para desarrollo web y propósitos de prueba locales.

Mysql workbench: herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos Mysql.

Wix Logo Maker: web para crear logo.

Mackflow: es una herramienta que permite crear borradores de la estructura de las páginas que compondrán nuestro sitio web, es decir, los prototipos de las distintas páginas.

Bootstrap: es la pantalla de interfaz con el usuario- para desarrollar aplicaciones que se adaptan a cualquier dispositivo.

Lenguajes

HTML: es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Por ejemplo, sus contenidos podrían ser párrafos, una lista con viñetas, o imágenes y tablas de datos. La utilizamos en vistas de nuestra página WEB.

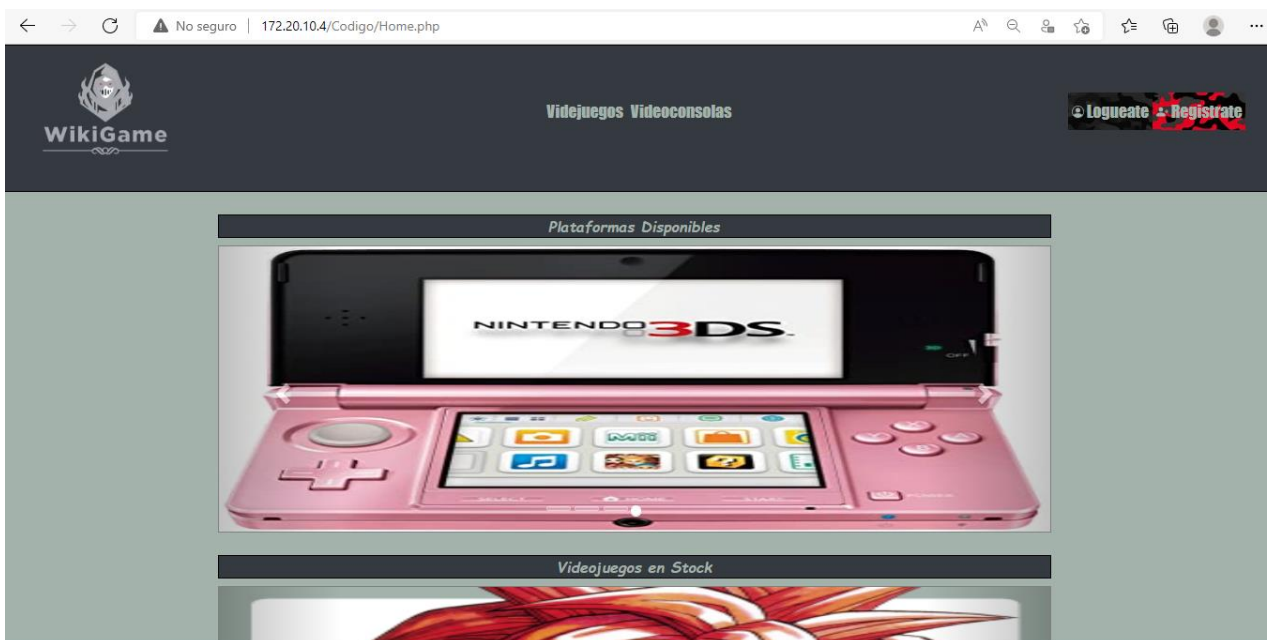
Css: lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un visitante las visita. Funciona junto al lenguaje HTML, que se encarga del contenido básico de las páginas.

PHP: es un lenguaje de programación que permite el desarrollo web o aplicaciones web dinámicas, el cual es apto para incrustar el lenguaje HTML,

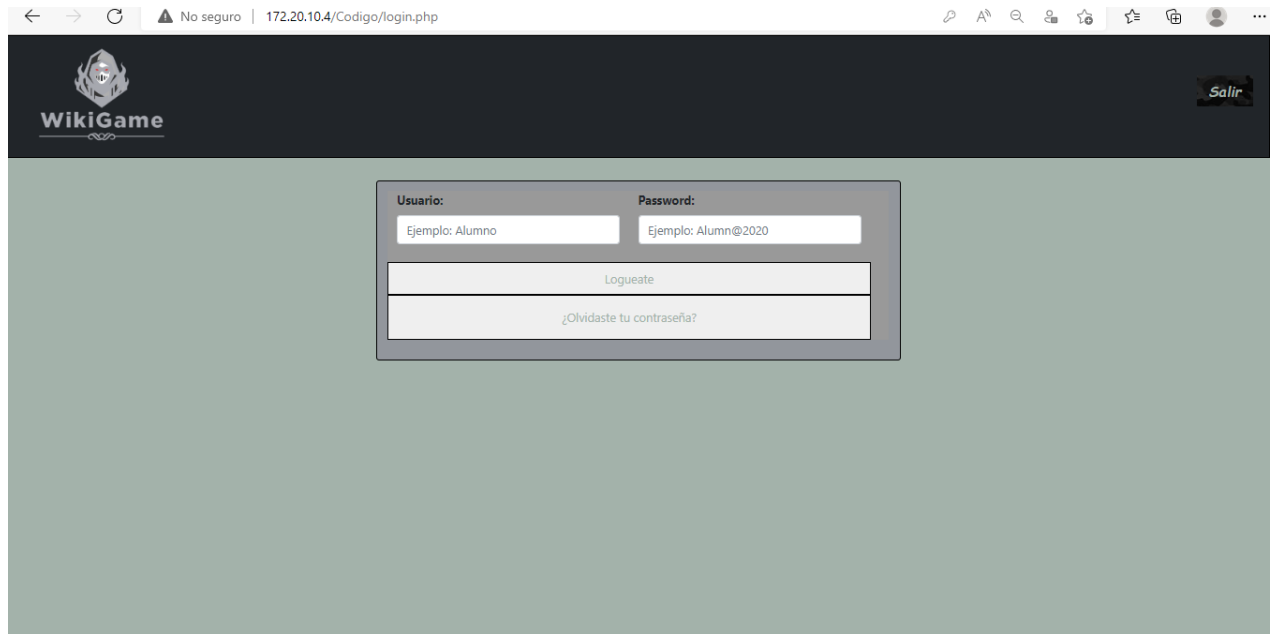
ahora bien, siempre siguiendo algunas reglas establecidas.

JavaScript: es el lenguaje de programación que debes usar para añadir características interactivas a tu sitio web.

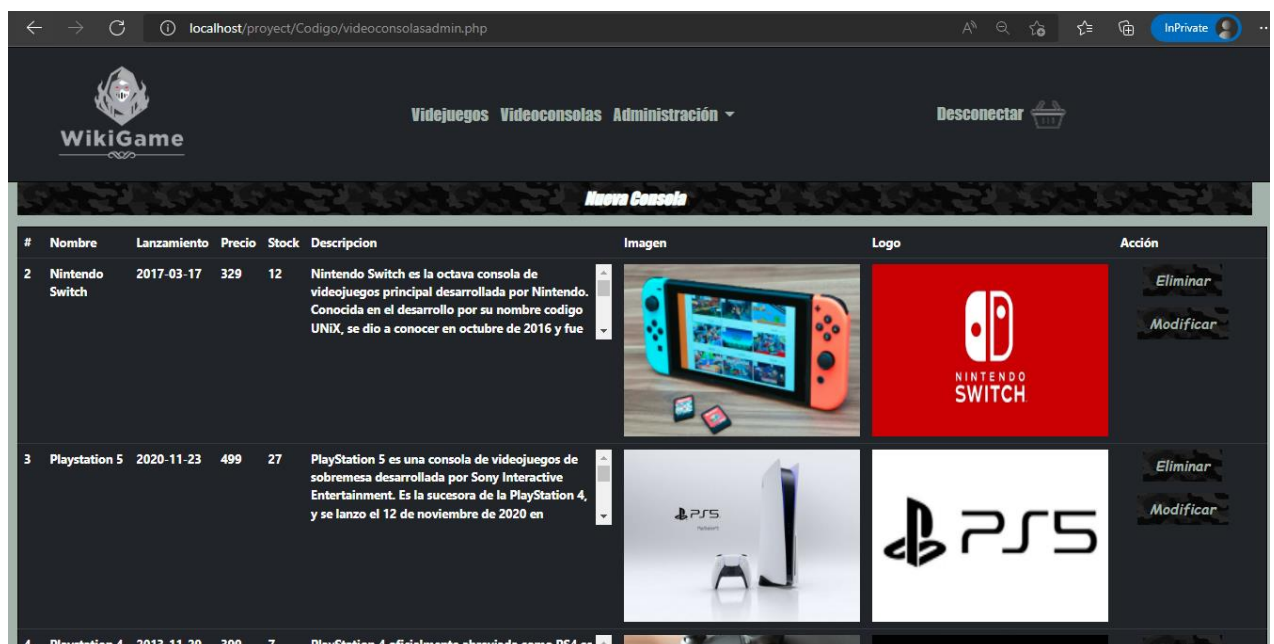
- **Producto**



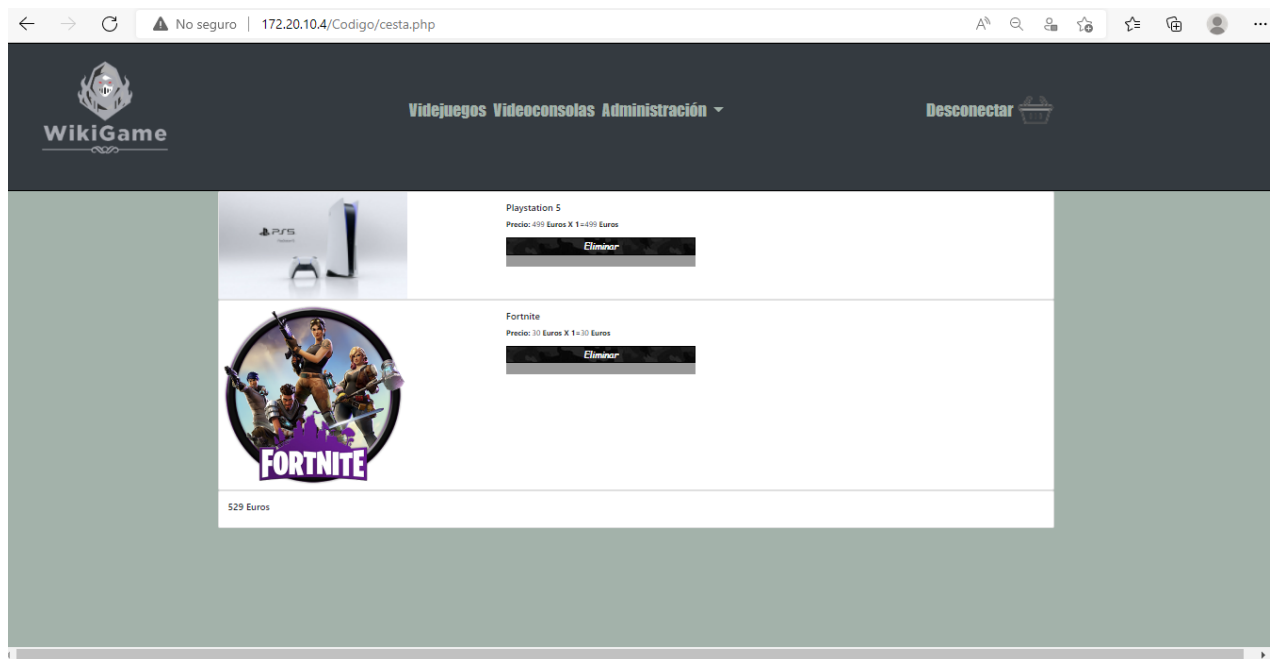
Este sería el resultado de mi HOME.



El login en nuestra Debian.



Esta seria mi panel de administración de videoconsola del suario admin.



Este seria el carrito con sus botones.

- Bibliografía.

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

<https://www.youtube.com/watch?v=HtWlrGqbEwI>

<https://www.lawebera.es/disenio-web/hacer-diseno-web-adaptable-responsive-design.php>

<https://www.youtube.com/watch?v=I75CUdSJfw&list=PLU8oAlHdN5BkinrODGXToK9oPAInIxmW>

<https://getbootstrap.com/>

<https://es.javascript.info/>

