**Registro de pruebas**

***Nombre del proyecto:***

OBDealer

***Autores del proyecto:***

Omar Eiyana Aboghrara

Badr Jebari Badireghioui

***Fecha: 05-05-2023***

**Contenido**

[1. Creación de BBDD y definir el modelo de E-R. 2](#_Toc134215345)

[1.1. 30/03/2023 2](#_Toc134215346)

[2. Creación del pool de conexiones. 2](#_Toc134215347)

[2.1. 20/03/2023 2](#_Toc134215348)

[3. Creación de los beans. 4](#_Toc134215349)

[3.1. 25/03/2023 4](#_Toc134215350)

[4. Login (Inicio de sesión). 5](#_Toc134215351)

[4.1. 01/04/2023 5](#_Toc134215352)

[5. Registro. 5](#_Toc134215353)

[5.1. 02/04/2023 5](#_Toc134215354)

[6. Proveedores. 6](#_Toc134215355)

[6.1. 21/04/2023 6](#_Toc134215356)

[7. Usuarios. 7](#_Toc134215357)

[7.1. 02/05/2023 7](#_Toc134215358)

[8. Productos. 7](#_Toc134215359)

[8.1. 03/05/2023 7](#_Toc134215360)

[9. Carrito. 8](#_Toc134215361)

[9.1. 03/05/2023 8](#_Toc134215362)

[10. Pago. 9](#_Toc134215363)

[10.1. 04/05/2023 9](#_Toc134215364)

# Creación de BBDD y definir el modelo de E-R.

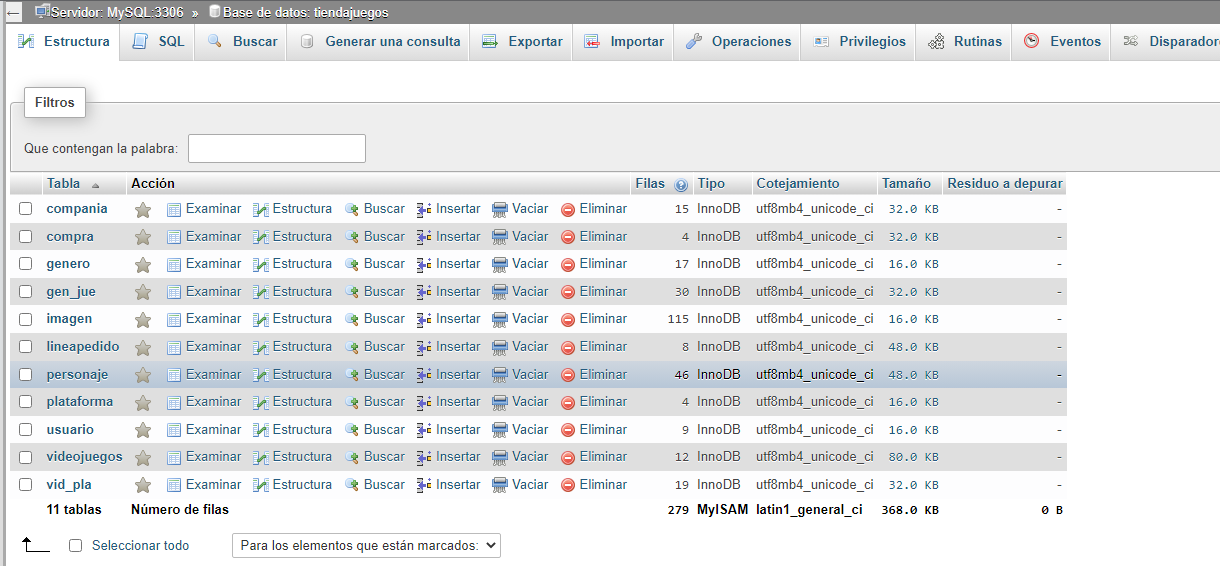
## 30/03/2023

**Escenario inicial:** Búsqueda de información relacionada con tiendas de videojuegos y alguna plantilla para utilizar como ejemplo ya desarrollado por nosotros anteriormente.

**Proceso seguido:** Desarrollamos la BBDD (tiendajuegos.sql) a través de phpMyAdmin (en SQL), definiendo las tablas, sus interrelaciones y agregando tuplas en estas.

**Resultado esperado:** Se esperaba contar con una BBDD completa y con una lógica de E-R bien definida, para que esta cuente con todas sus tablas comunicadas entre sí.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



BBDD (tiendajuegos) en phpMyAdmin.

# Creación del pool de conexiones.

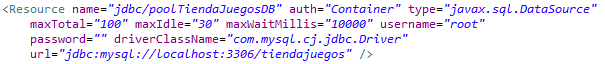
## 20/03/2023

**Escenario inicial:** Contábamos con varias conexiones a bases de datos, que habíamos realizado previamente.

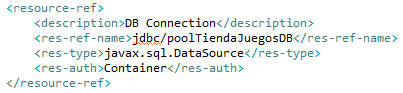
**Proceso seguido:** Elegir el tipo de conexión de base de datos que mejor consideramos y la implementamos en el código fuente. Hay que añadir un par delineas de código en el archivo context.xml del server, en el archivo web.xml del proyecto y creamos el archivo que hemos llamado BDConex.java también dentro del proyecto en una carpeta llamada conex.

**Resultado esperado:** Se esperaba tener una conexión la BBDD (tiendajuegos), y tener acceso a todas sus tablas.

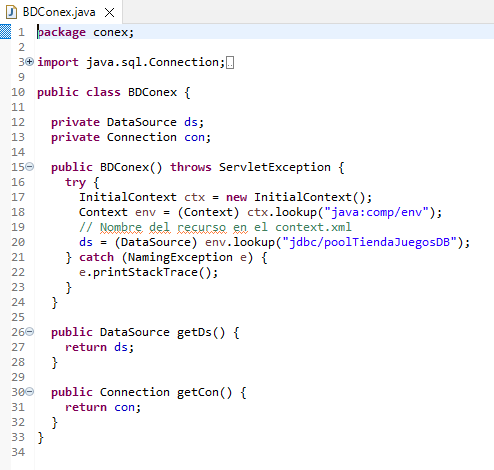
**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Context.xml



web.xml



BDConex.xml

# Creación de los beans.

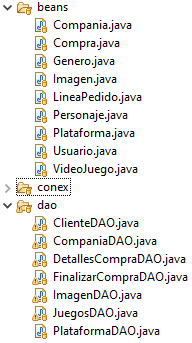
## 25/03/2023

**Escenario inicial:** Este es el comienzo de esta tarea, con lo cual el escenario inicial es nulo.

**Proceso seguido:** En el código fuente añadimos la carpeta *beans* donde creamos los objetos de la BBDD y también la carpeta de *daos* donde creamos los métodos con las consultas que nos devuelve la información que necesitamos de la BBDD.

**Resultado esperado:** Se esperaba era obtener objetos reutilizables y así evitar la tediosa tarea de programar la web.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Carpetas beans y daos en el código fuente del proyecto.

# Login (Inicio de sesión).

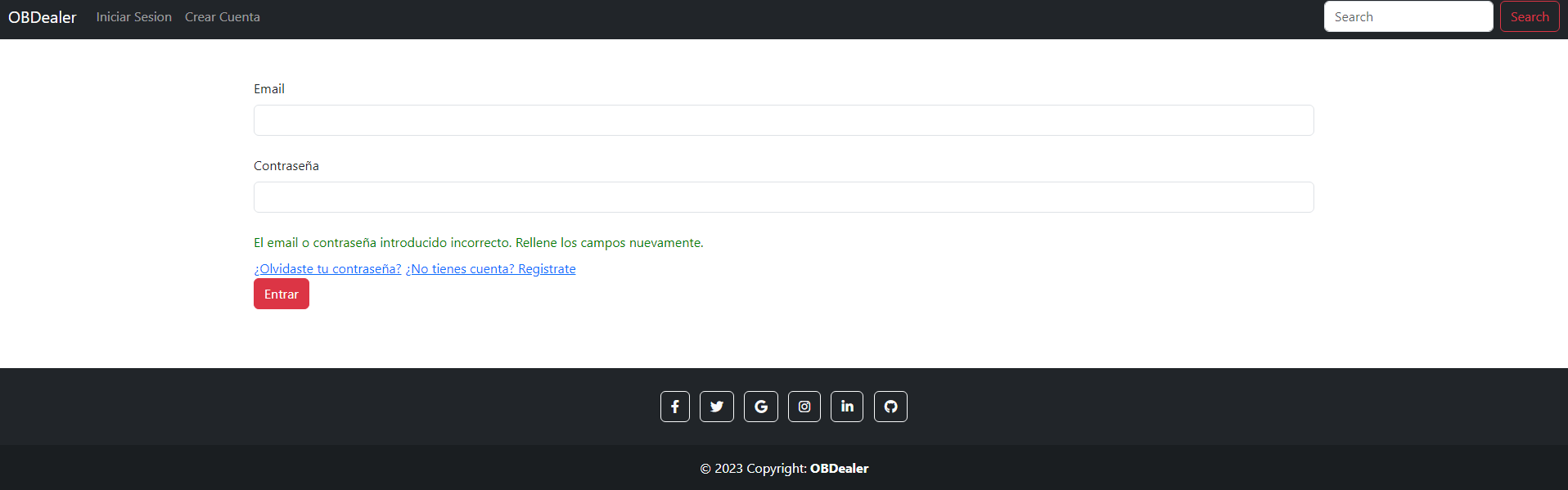
## 01/04/2023

**Escenario inicial:** A partir de este escenario es necesario tener terminados y bien desarrollados los putos anteriores (Creación de la base de datos, la conexión con la base de datos y los beans).

**Proceso seguido:** En el código fuente desarrollamos la vista y servlet del login.

**Resultado esperado:** Se esperaba verificase si el email y contraseña coincide con algún usuario que este registrado en la BBDD. En caso afirmativo comprobar si es un administrador o un cliente y darle los accesos pertinentes a cada uno.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Prueba intentando acceder con un email y contraseña incorrectos.

# Registro.

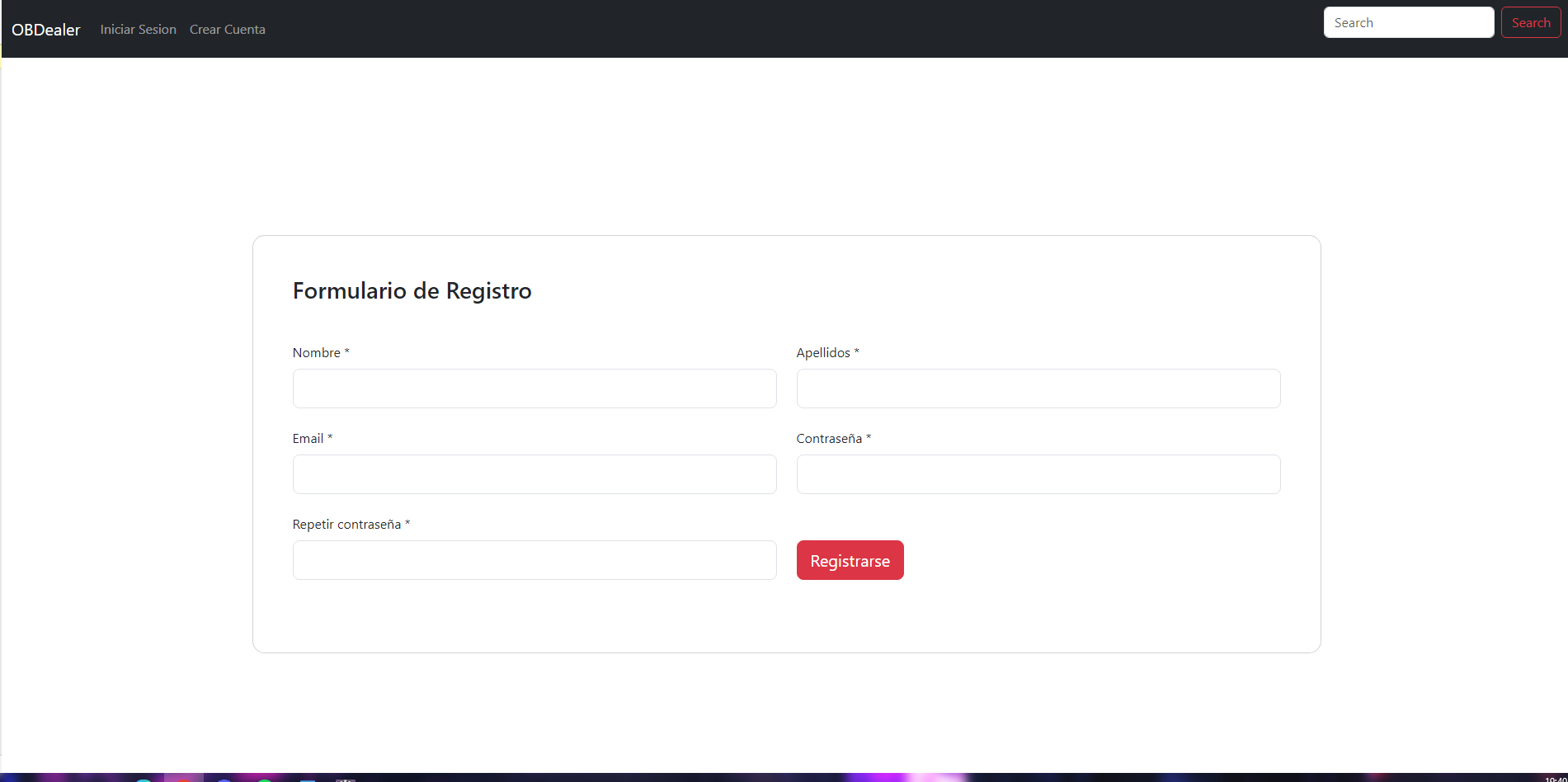
## 02/04/2023

**Escenario inicial:** A partir de este escenario es necesario tener terminados y bien desarrollados los putos anteriores (Creación de la base de datos, la conexión con la base de datos y los beans).

**Proceso seguido:** En el código fuente desarrollamos la vista y servlet del registro.

**Resultado esperado:** Introduciendo la información que se pide, crea un usuario y lo agrega a la BBDD en caso de que no exista un usuario con el mismo email ya registrado previamente. Muestra un mensaje de éxito o no dependiendo de si ha sido posible.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Interfaz para registrase.

# Proveedores.

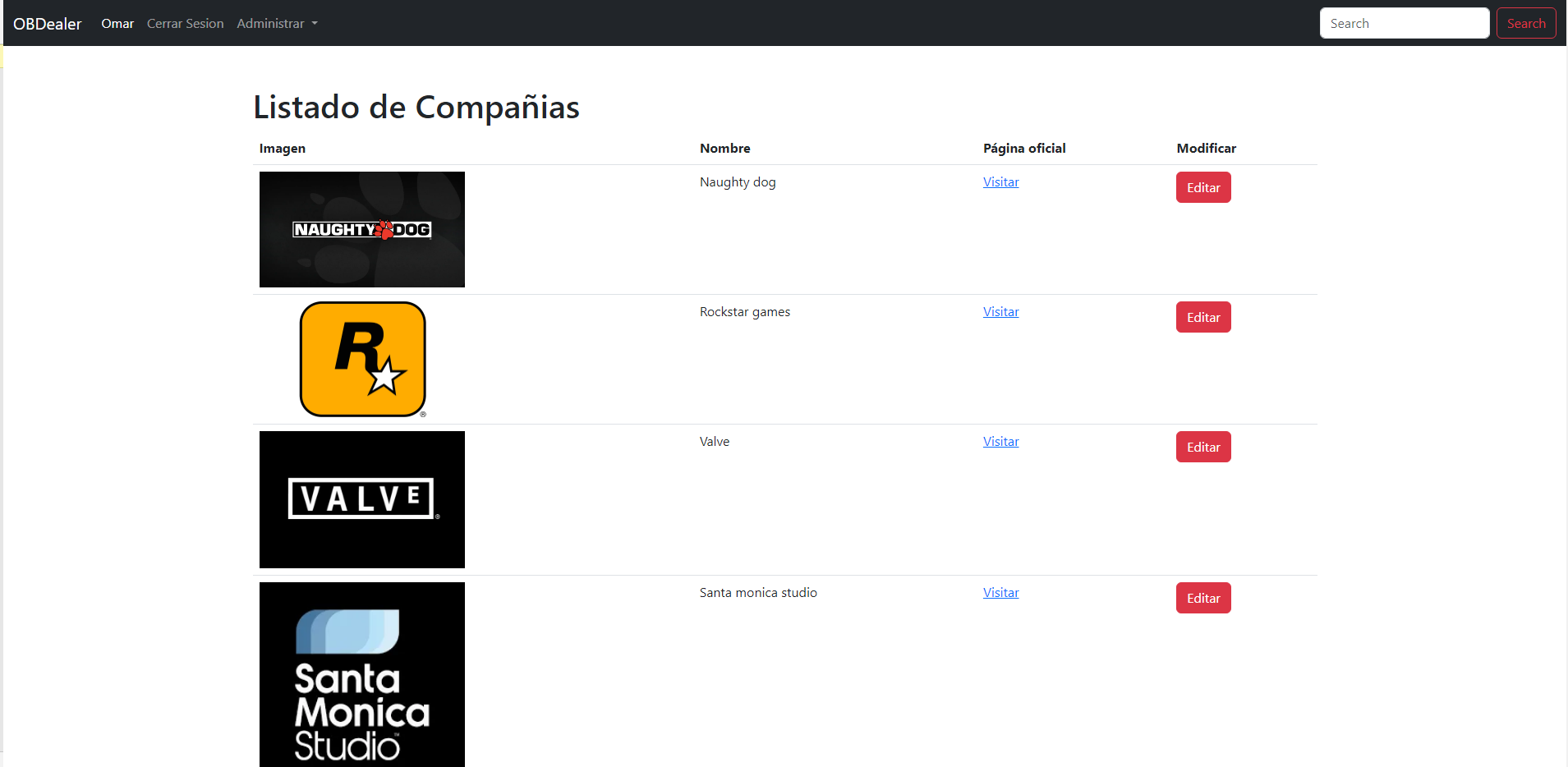
## 21/04/2023

**Escenario inicial:** Contamos con el login desarrollado, este verifica que es un administrador y nos da acceso a las compañías.

**Proceso seguido:** En el código fuente desarrollamos la vista y servlet del listado compañías y detalle de compañías.

**Resultado esperado:** Queremos que se listen todas las compañías, con la información que tenemos de estas almacenada. Al lado de cada compañía un botón que nos permite editar cada una de estas.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Listado de todas las compañías.

# Usuarios.

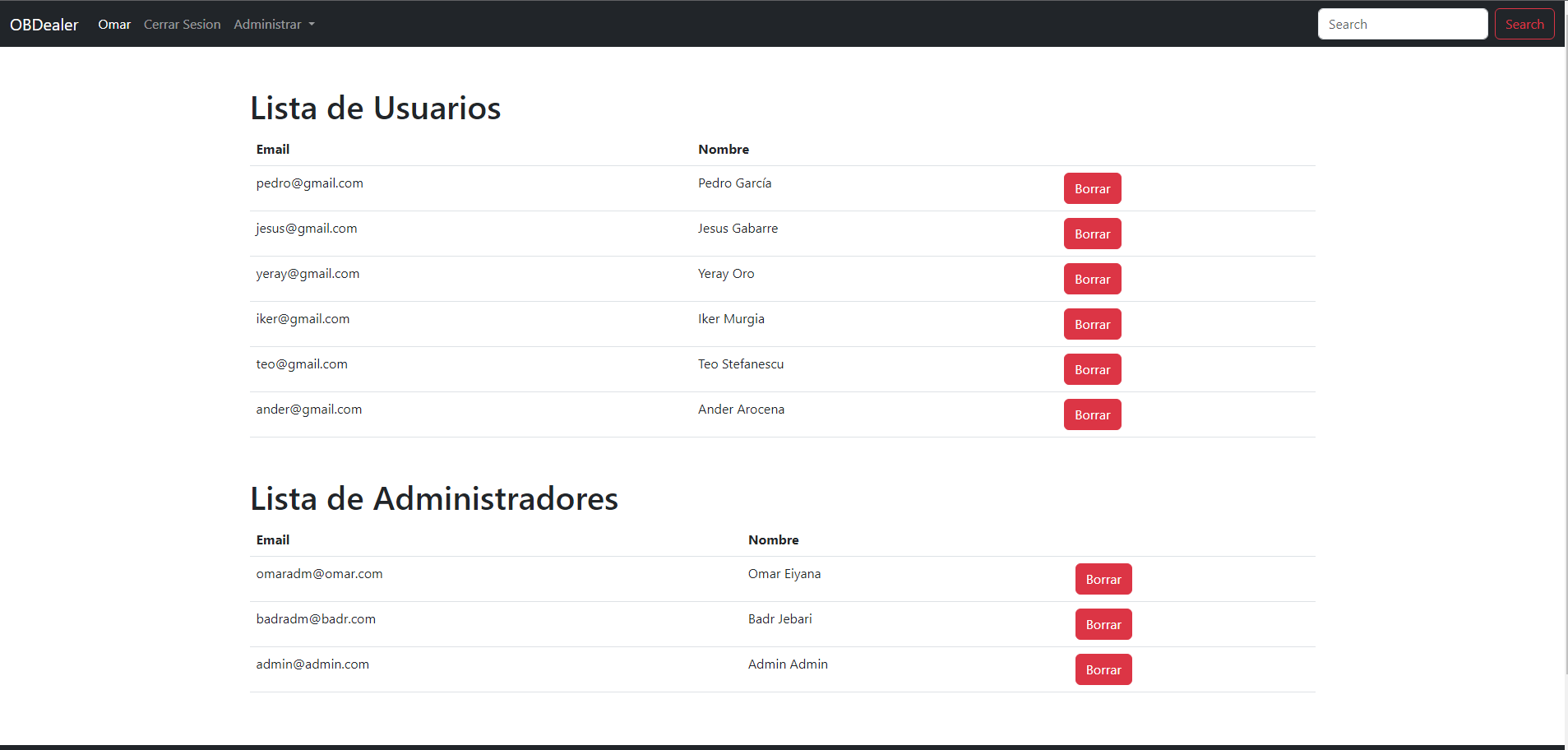
## 02/05/2023

**Escenario inicial:** Contamos con el login desarrollado, este verifica que es un administrador y nos da acceso a los usuarios.

**Proceso seguido:** En el código fuente desarrollamos la vista y servlet del listado usuarios y perfil usuario.

**Resultado esperado:** Queremos que se listen todos los usuarios registrados, en una tabla los clientes y en otra los administradores. Al lado de cada usuario un botón que nos permite borrar cada uno de estos. En el navegador una opción que nos redirige a otra sección donde nos permite ver la información del usuario que haya iniciado sesión actualmente. En esta muestra toda su información almacenada (datos privados y listado de compras que ha realizado), con una opción para cambiar la contraseña.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Listado de todos los usuarios registrados.

# Productos.

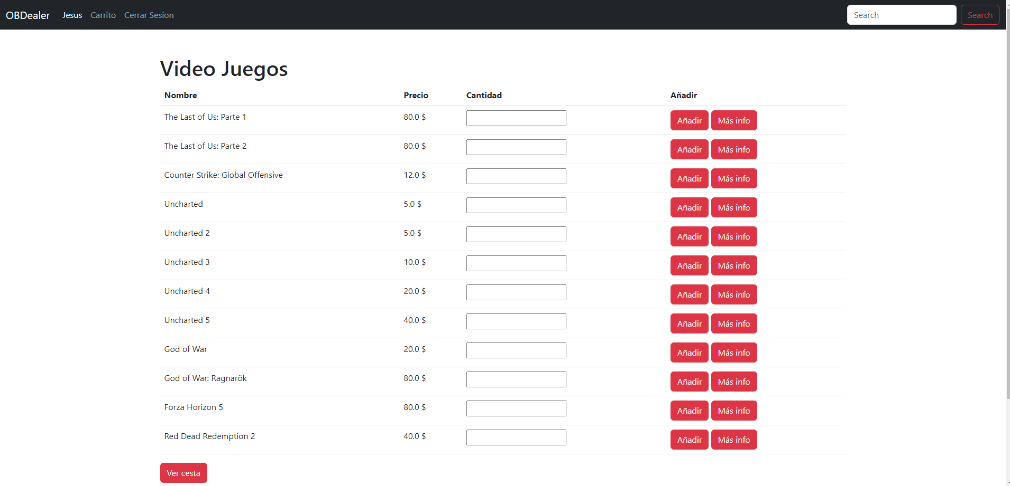
## 03/05/2023

**Escenario inicial:** Contamos con el login desarrollado, este verifica que es un usuario registrado y nos da acceso a los productos.

**Proceso seguido:** En el código fuente desarrollamos la vista y servlet del listado productos y detalles productos.

**Resultado esperado:** Queremos que se listen todos los productos (videojuegos) y una opción de que nos dirija a más información de cada videojuego. En caso de que sea un cliente este puede añadir una cantidad de cada producto al carrito o en caso de que sea un administrador de borra cada producto.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Listado de todos los videojuegos desde la vista de un cliente.

# Carrito.

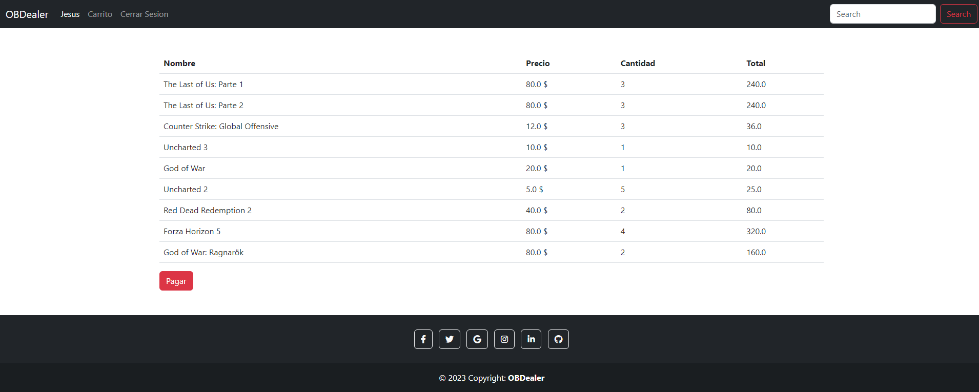
## 03/05/2023

**Escenario inicial:** Contamos con el login y listado productos desarrollados. Debe de iniciar sesión un usuario cliente y tener algún videojuego añadido al carrito.

**Proceso seguido:** En el código fuente desarrollamos la vista y servlet de carrito.

**Resultado esperado:** Se tiene que listar todos los videojuegos con la cantidad correspondiente que ha añadido el cliente al carrito. Se indica el precio total.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Carrito con los productos que se han añadido a la cesta.

# Pago.

## 04/05/2023

**Escenario inicial:** Contamos con el login, listado productos y carrito desarrollados. Debe de iniciar sesión un usuario cliente, tener algún videojuego añadido al carrito y darle a pagar.

**Proceso seguido:** En el código fuente desarrollamos la vista y servlet de pago.

**Resultado esperado:** Al confirmar el pago de los productos que se encuentran en la cesta del usuario cliente, debe salir un mensaje de confirmación del pago.

**Resultado final:** Se ha obtenido el resultado esperado, sin ningún problema.



Mensaje de confirmación pago.