



**ULPGC**  
Universidad de  
Las Palmas de  
Gran Canaria

Escuela de  
Ingeniería Informática



# Aplicación web con integración de sistemas especializada en la venta de segunda mano de videojuegos

**Alumno:** Cristian del Toro Méndez

**Tutor:** Francisco José Santana Pérez

**GRADO EN INGENIERÍA INFORMÁTICA**

Julio 2021

## Índice

1.INTRODUCCIÓN .....	6
2. ESTADO ACTUAL Y OBJETIVOS.....	7
EBAY.....	8
WALLAPOP .....	8
VINTED .....	9
FACTOR DIFERENCIADOR: .....	9
3. COMPETENCIAS ESPECÍFICAS CUBIERTAS .....	9
3.1 COMPETENCIAS COMUNES A LA INGENIERÍA INFORMÁTICA.....	9
3.2 COMPETENCIAS DE LA INTENSIFICACIÓN “SISTEMAS DE LA INFORMACIÓN” .....	10
3.3 COMPETENCIAS DEL TRABAJO DE FIN DE GRADO .....	11
4. RECURSOS NECESARIOS.....	11
4.1 RECURSOS HARDWARE.....	12
4.2 RECURSOS SOFTWARE.....	12
4.2.1 VISUAL STUDIO CODE .....	12
4.2.2 MYSQL WORKBENCH .....	13
4.2.3 XAMPP .....	13
4.3 LIBRERÍAS, LENGUAJES Y FRAMEWORKS EMPLEADOS .....	13
4.3.1 HTML.....	14
4.3.2 CSS ( <i>Cascading Style Sheets</i> ) .....	14
4.3.3 BOOTSTRAP .....	15
4.3.4 PHP 8.....	15
4.3.5 JAVASCRIPT .....	16
4.3.6 AJAX .....	16
4.3.7 MYSQL.....	16
4.3.8 LIBRERÍA TCPDF .....	17
4.3.9 GLIDE.JS .....	17
4.3.10 SWEETALARM2 .....	18
4.3.11 PASARELA DE PAGO.....	18
5. DESARROLLO.....	20
5.1 PLAN DE TRABAJO.....	20
5.2 ANÁLISIS DE VIABILIDAD Y FORMACIÓN EN WORDPRESS Y WOOCOMMERCE ...	21

5.2.1 WORDPRESS Y WOOCOMMERCE .....	21
5.3 INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA .....	22
5.3.1 INSTALACIÓN DE XAMPP .....	22
5.3.2 INSTALACIÓN DE VISUAL STUDIO CODE.....	23
5.3.3 INSTALACIÓN DE WORKBENCH .....	23
5.4 BASE DE DATOS.....	24
5.4.1 INTRODUCCIÓN .....	24
5.4.2 DISEÑO E/R Y CREACIÓN DE LA BASE DE DATOS.....	24
5.4.3 TABLAS Y CAMPOS.....	26
5.4.4 CONEXIÓN CON LA BASE DE DATOS.....	31
5.5 DESARROLLO DE LA TIENDA “GAME SECOND LIFE” .....	33
5.5.1 ESTRUCTURA DE “GAMES SECOND LIFE” .....	34
5.5.2 PÁGINAS CABECERA_HEAD.PHP Y CABEZERA.PHP .....	37
5.5.3 BUSCADOR.....	38
5.5.4 REGISTRAR Y EDITAR USUARIO .....	41
5.5.5 INICIAR SESIÓN Y CERRAR SESIÓN.....	44
5.5.6 CAMBIAR CONTRASEÑA .....	45
5.5.7 BORRAR PERFIL.....	46
5.5.8 AÑADIR, EDITAR Y ELIMINAR ANUNCIOS .....	47
5.5.9 COMPRAR Y GENERAR FACTURA.....	52
5.5.10 AÑADIR A FAVORITOS.....	60
5.5.11 VER LAS VENTAS REALIZADAS Y PEDIDOS REALIZADOS .....	63
5.5.12 TRADUCCIÓN .....	64
5.5.13 FUNCIONES AUXILIARES Y CONTROL DE ERRORES .....	66
6. ANÁLISIS DE COSTO Y MODELO DE NEGOCIO .....	70
7. CONCLUSIONES.....	73
7.1 CONCLUSIONES BASADAS EN EL RESULTADO .....	73
7.2 CONCLUSIONES BASADAS EN LOS OBJETIVOS .....	73
7.3 CONCLUSIONES BASADAS EN FUTURAS MEJORAS .....	73
7.4 CONCLUSIONES A NIVEL PERSONAL.....	74
8. BIBLIOGRAFÍA .....	75
Webgrafía .....	75
ANEXO I MANUAL DE USUARIO.....	78

ANEXO II CÓDIGO .....	85
estilo.css.....	85
cabecera_head.php .....	86
cabecera.php .....	87
index.php .....	95
anuncio.php .....	96
publicar.php.....	99
borraranuncio.php.....	101
borrarperfil.php .....	102
buscar.php .....	103
cambiarclave.php .....	104
comprar.php .....	105
pagado.php.....	107
conexion.php .....	109
config.php .....	109
contacto.php.....	110
favorito.php .....	110
favoritos.php.....	111
iniciar_sesion.php.....	112
pedidos.php .....	113
perfil.php .....	114
registrar.php .....	115
salir.php .....	118
traduccion.php.....	118
ventas.php .....	120

# 1.INTRODUCCIÓN

El auge del comercio online o también denominado **E-commerce** es algo que visualizamos en el día a día con cada año que pasa. Cada vez más tiendas físicas y autónomos se lanzan a digitalizar su negocio y por ello existen multitud de herramientas que facilitan el desarrollo web.

Por ello, este trabajo de fin de grado consiste en el **desarrollo de una aplicación web especializada en la venta online de segunda mano de videojuegos**, mediante la integración de diferentes herramientas, APIs y módulos de código abierto.

La motivación de **integrar estos distintos sistemas** es poder tener una página totalmente funcional en la que los distintos apartados al estar integrados permiten automatizar y simplificar funcionalidades de la web. Además de tener más facilidad a la hora de focalizar los errores.

A su vez otro objetivo fundamental es **afianzar los conocimientos adquiridos en la carrera**, a la vez que aprender el correcto funcionamiento y puesta en marcha de distintas tecnologías y herramientas.

Se ha optado por la venta de segunda mano de videojuegos debido a que existen multitud de aplicaciones que permiten la venta **C2C (Consumer to Consumer)** de artículos específicos como coches, ropa, entre otros. Pero existen muy pocas de videojuegos, por lo que elegimos tener ese factor diferenciador y captar ese sector.

La **aplicación** se ha diseñado para intentar ser lo más intuitiva posible:

Desde la **página principal** el usuario tendrá disponible una barra de navegación desde la que podrá dirigirse al formulario de contacto, registrarse o iniciar sesión y el buscador de productos. También visualizará un slider con acceso a los últimos productos añadidos a la web.

Para poder **añadir artículos** a la venta el usuario necesita registrarse en la web. Solo después de estar registrado se le habilitará las opciones para administrar sus productos.

En cambio, si permitimos comprar artículos a usuarios registrados como sin registrar. Gracias a integrar **PayPal** como nuestra forma de pago, no se requiere de registro para comprar un producto, ya que la dirección para enviar el artículo se recoge de la propia cuenta del usuario de PayPal

Al ser una web pensada para la venta de segunda mano entre personas de diferentes lugares, se optado por hacer la página **multilinguaje** (español, inglés y alemán).

## 2. ESTADO ACTUAL Y OBJETIVOS

Según se puede leer en el blog informativo del Instituto Nacional de Estadística («El salto del comercio electrónico», 2020):

“Uno de cada tres hogares realizó alguna compra por Internet en el último año (32,5%), según la Encuesta de presupuestos familiares. Estas compras concentraron el 2,1% de todo el gasto en consumo de los hogares en 2019, casi un punto más que hace tres años.

Los apartados con mayor proporción de gasto por Internet fueron Ocio y cultura (con el 10,9% del total del gasto de este grupo)”

Como se puede observar el aumento de compras online en la población va en aumento con cada año que pasa. Por ello, el objetivo de este trabajo consiste en el desarrollo de una aplicación web especializada en la venta online de segunda mano de videojuegos, mediante la integración de diferentes herramientas, APIs y módulos de código abierto.

Gracias a este auge del comercio online cada vez tenemos más herramientas disponibles a la hora de facilitar el diseño y desarrollo de una tienda online. Siendo posiblemente Wordpress la más usada.

Se planteó la posibilidad de elaborar un diseño en Wordpress ya que el proceso es bastante sencillo y podría establecerse una comparación con el desarrollo hecho a mano. Pero debido a que el uso de la herramienta conlleva una suscripción económica se descartó esa posibilidad.

Por lo que se integrarán sistemas de código libre y gratuito como explicaremos más detalladamente en el apartado de desarrollo.

En la actualidad existen multitud de tipos de comercio online:

- **B2C** (*Business to Consumer*): Es el estilo de negocio tradicional, en el que el comercio oferta los productos o servicios a los consumidores. Seguramente el ejemplo más sonado sea el de Amazon que comenzó vendiendo libros y actualmente oferta una variedad inmensa de artículos.
- **B2B** (*Business to Business*): Este tipo de comercio digital consiste en la “compraventa” de un negocio a negocio, ya sean servicios o productos. Gestorías online, agencias de publicidad, entre otros.
- **C2B** (*Consumer to Business*): Es el tipo de negocio menos frecuente en la que el consumidor genera valor para la empresa. Podemos ver este tipo de negocio en la relación de los influencers con los distintos comercios a la hora de hacer promociones.
- **C2C** (*Consumer to Consumer*): Se establece una comunicación comercial entre consumidores. Los particulares son quienes ofrecen sus servicios/productos y establecen el precio. Wallapop, Ebay, Vinted, etc.

Para el desarrollo de la app se ha optado por el modelo **C2C**, debido al creciente éxito de otras plataformas como Wallapop, Ebay, Vinted, entre otros. La gran mayoría simplemente ofrece un espacio en la que poder subir cualquier tipo de producto, no existe una especialización contando alguna excepción.

Lo que ofrece nuestra aplicación es la posibilidad de crear tu propio espacio online en la que administrar los videojuegos que pones a la venta a los demás usuarios. Para ello el formulario de añadir artículos contempla diferentes aspectos de los videojuegos, en los que el cliente con un solo vistazo a la ficha del videojuego ya sepa todo lo destacable de este tipo de productos.

Las principales aplicaciones que ofrecen un sistema de negocio online **C2C** son Ebay, Wallapop y Vinted:



EBAY

Ebay fundada el 3 de septiembre de 1995 es una de las apps pioneras en el negocio online. Es la app más usada cuando buscas objetos de coleccionismo o difíciles de encontrar, además de ser la mayor app de compraventa a nivel mundial.

El tipo de usuario medio es el que ya tiene experiencia comprando online, conoce cómo funcionan las aduanas y sabe identificar falsificaciones. En ella se pueden vender los artículos por venta directa o subasta.



WALLAPOP

Wallapop es una empresa española fundada en 2013, que ofrece una aplicación destinada a la venta de segunda mano de todo tipo de artículos. Para ello hace uso de la geolocalización de los dispositivos móviles para que los usuarios oferten y compren productos en relación a su distancia.

En ella se establecen distintas categorías tecnología, automóvil, electrodomésticos, etc. No se centran en ningún artículo específico, sino que quiere centrarse en la proximidad de los productos para conseguir la venta.



## VINTED

Vinted es una app que da la posibilidad a los usuarios de vender, comprar e intercambiar artículos de ropa y accesorios de segunda mano. Según su Web, Vinted tiene 20 millones de miembros con una media de 15 millones de usuarios registrados, siendo una app especializada en moda de segunda mano.

Al especializarse en moda ha logrado atraer a ese tipo de público y afianzarlo para usar su app a la hora de vender sus prendas antiguas o sin uso.

### FACTOR DIFERENCIADOR:

Por lo que nuestro factor diferenciador será la especialización en videojuegos, de la misma forma que Vinted ha captado su público orientando su negocio a la moda. Queremos conseguir el mismo efecto atrayendo al público del sector de los videojuegos, Proporcionarle un entorno más cómodo y rápido de encontrar ese tipo de productos.

## 3. COMPETENCIAS ESPECÍFICAS CUBIERTAS

En este apartado van a cubrir aquellas competencias que, a mi juicio, guarden una mayor relación con el desarrollo de este proyecto. Hay dos tipos de competencias, las comunes al grado de ingeniería informática y las específicas de la intensificación elegida, en mi caso sistemas de información.

### 3.1 COMPETENCIAS COMUNES A LA INGENIERÍA INFORMÁTICA

#### CII01

**(Extraído de: Plan de estudios del Grado de Ingeniería Informática, plan 40, de la ULPGC)** “Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.”

En la realización de este trabajo se ha llevado a cabo tanto las fases de diseño como de desarrollo en la aplicación de la tienda online. Se ha implementado también algunas restricciones de acceso a diferentes apartados de la página a usuarios no registrados. Como la posibilidad de añadir o modificar anuncios.

#### **CII02**

**(Extraído de: Plan de estudios del Grado de Ingeniería Informática, plan 40, de la ULPGC)** “Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.”

El desarrollo de esta aplicación, así como la puesta en marcha del servidor local se ha llevado a cabo en mi equipo. Se ha tenido en cuenta el impacto económico al segregar el tipo de contenido que se oferta en la aplicación.

#### **CII012**

**(Extraído de: Plan de estudios del Grado de Ingeniería Informática, plan 40, de la ULPGC)** “Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos.”

Dentro de este trabajo las bases de datos son una parte indispensable para su correcto funcionamiento. En ella almacenaremos toda la información sobre los usuarios y los anuncios publicados. Para ello haremos uso de “MySQL”.

#### **CII017**

**(Extraído de: Plan de estudios del Grado de Ingeniería Informática, plan 40, de la ULPGC)** “Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.”

En la fase de diseño se ha tenido en cuenta que la aplicación se pueda usar de manera intuitiva y funcional sin necesidad de conocimientos previos.

### **3.2 COMPETENCIAS DE LA INTENSIFICACIÓN “SISTEMAS DE LA INFORMACIÓN”**

#### **SI03**

“Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación”

Las fases de especificación, diseño e implementación al tratarse de un trabajo individual se han llevado a cabo por el autor del trabajo.

Al pasarse en un futuro a producción la fase de mantenimiento sería clave.

#### SI04

**(Extraído de: Plan de estudios del Grado de Ingeniería Informática, plan 40, de la ULPGC)** “Capacidad para comprender y aplicar los principios y prácticas de las organizaciones, de forma que puedan ejercer como enlace entre las comunidades técnica y de gestión de una organización y participar activamente en la formación de los usuarios.”

Durante el desarrollo de la aplicación, se ha tenido en cuenta la visión de usuarios sin conocimientos previos de Informática o Ingeniería, por lo que el *feedback* realizado ha sido una parte clave a la hora de diseñar la interfaz de usuario.

### 3.3 COMPETENCIAS DEL TRABAJO DE FIN DE GRADO

#### TFG01

**(Extraído de: Plan de estudios del Grado de Ingeniería Informática, plan 40, de la ULPGC)** “Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sintetizen e integren las competencias adquiridas en las enseñanzas.”

Este trabajo cumple con todos los requisitos recogidos en la competencia TFG01 que corresponde al trabajo de fin de grado. Se encuentra dentro del ámbito de la ingeniería en informática, se ha realizado de forma individual y defendido ante el tribunal universitario.

## 4. RECURSOS NECESARIOS

En el apartado de recursos necesarios se va a detallar el hardware, software y herramientas que han sido usadas en el desarrollo de este proyecto.

## 4.1 RECURSOS HARDWARE

El proyecto se ha realizado en un equipo con las siguientes características:

Sistema operativo:	Windows 10 Pro 64 bits (10.0, compilación 19041)
Idioma:	español (configuración regional: español)
Fabricante del sistema:	Micro-Star International Co., Ltd.
Modelo del sistema:	MS-7B48
BIOS:	BIOS Date: 08/03/18 08:55:50 Ver: V2.70
Procesador:	Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz (12 CPUs), ~3.2GHz
Memoria:	16384MB RAM
Archivo de paginación:	11790MB usados, 6971MB disponibles
Versión de DirectX:	DirectX 12

*Figura 4.1 Características del equipo personal usado en el proyecto*

Nombre:	NVIDIA GeForce GTX 1080
Fabricante:	NVIDIA
Tipo de chip:	GeForce GTX 1080
Tipo de DAC:	Integrated RAMDAC
Tipo de dispositivo:	Dispositivo de pantalla completa
Memoria total aprox.:	16244 MB
Memoria de pantalla (VRAM):	8079 MB
Memoria compartida:	8165 MB

*Figura 4.2 Características de la tarjeta gráfica del equipo*

## 4.2 RECURSOS SOFTWARE



### 4.2.1 VISUAL STUDIO CODE

Se ha optado por **Visual Studio Code** como entorno de desarrollo para la aplicación. Es un editor de código optimizado con tareas para desarrolladores como control de versiones y depuración de código. Con multitud de atajos para facilitar la inyección de código



#### 4.2.2 MYSQL WORKBENCH

MySQL Workbench es una herramienta que permite modelar diagramas de entidad-relación para bases de datos MySQL. Nos permite diseñar el esquema de la base de datos de forma visual.



#### 4.2.3 XAMPP

A su vez necesitaremos de un servidor Apache, en esta ocasión se ha elegido el servidor libre y gratuito **XAMPP**. Es muy cómodo de instalar y de usar, teniendo en el mismo paquete de instalación todos los recursos para empezar el desarrollo. MariaDB como gestor de base de datos y Php y Perl como lenguajes de programación. Se ha usado la versión 8.0.6.

### 4.3 LIBRERÍAS, LENGUAJES Y FRAMEWORKS EMPLEADOS



### 4.3.1 HTML

HTML (*HyperText Markup Language*) es el código que se usa para diseñar, estructurar y desarrollar una página web. Es un estándar del *World Wide Web Consortium*, la cual es una organización que se encarga de estandarizar las tecnologías ligadas a la web.

Para la realización de este trabajo se está empleando la última versión, en este caso la 5. El HTML se ha combinado con CSS y se ha hecho uso del *framework* de Bootstrap de los que se darán más detalles en sus apartados.

```
!DOCTYPE html
<html>
  <head>
    <?php include("cabecera_head.php"); ?>
  </head>
  <body>
    <?php
      include("cabecera.php");
    ?>
    <section class="admin">
      <div class="container" style="text-align: center">
        <button type="button" class="btn btn-primary" onclick="window.open('publicar.php');">Nuevo Anuncio</button>
        <div class="row">
          <h2>Favoritos del usuario</h2>
        </div>
      </div>
    </section>
  </body>
</html>
```

Figura 4.3 Ejemplo de código HTML



### 4.3.2 CSS (*Cascading Style Sheets*)

CSS también conocidas en español como Hojas de Estilo en Cascada nos permite aplicar estilo a las páginas de nuestro sitio web y por tanto controlar la apariencia de ella. La versión usada en este trabajo ha sido CSS3 y la combinación con Bootstrap.

Los estilos son aplicados a casi todos los elementos HTML que forman la página web (Tablas, textos, listas, enlaces, etc.)

```
*{
  margin: 0;
  padding: 0;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
header{
  box-shadow: 0 5px 30px #indigo;
  padding: 10px;
}
```

Figura 4.4 Ejemplo de código CSS



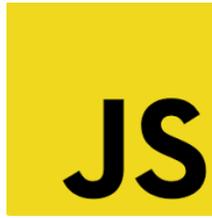
#### 4.3.3 BOOTSTRAP

Bootstrap es un *framework* multiplataforma de HTML, CSS y JavaScript de código abierto. Se ha empleado para facilitar las labores de diseño de en la web como sus formularios y para hacer la web *responsive* adaptándola a dispositivos móviles.



#### 4.3.4 PHP 8

PHP (*HyperText Preprocessor*) es un lenguaje de programación especialmente usado en el desarrollo de aplicaciones web. Todo el código generado en PHP debe ser procesado por un servidor web el cual devolverá una respuesta HTTP que puede ser de cualquier tipo. Se ha usado la versión 8 de PHP



#### 4.3.5 JAVASCRIPT

JavaScript es un lenguaje de programación orientado a objetos, se ejecuta del lado del cliente por lo que permite hacer las interacciones con la página web de manera más dinámica.

A su vez, nos brinda multitud de herramientas para validar formularios y mostrar alertas al usuario. Se ha hecho uso de las librerías de jQuery 3.4.1, de Sweetalert2 para mostrar las alertas y finalmente de Glide.js para generar el slider de imágenes.



#### 4.3.6 AJAX

AJAX significa JavaScript asíncrono y XML (*Asynchronous JavaScript and XML*). Permite que las aplicaciones web funcionen de forma asíncrona, por lo tanto, podemos enviar o recibir datos del servidor sin necesidad de estar cargando la página. En nuestro caso lo usaremos para habilitar la función de añadir o quitar de favoritos un producto.



#### 4.3.7 MYSQL

Se dispone del sistema de gestión de bases de datos MySQL. El diseño entidad-relación se ha llevado a cabo en MySQL Workbench y posteriormente desarrollado en phpMyAdmin que establecerá la conexión entre la base de datos y las paginas php.



#### 4.3.8 LIBRERÍA TCPDF

La librería TCPDF es de código abierto y se usa para el desarrollo de documentos pdf mediante código php.

La librería en sí tiene multitud de opciones para dar formato y estilo a los pdf, en cambio, la forma de usarse es algo engorrosa por la cantidad de líneas de código requeridas para dar formato al texto. Además, que la distribución de sus ficheros es algo liosa.

Para descargarnos la librería debemos acceder al siguiente enlace:

<https://tcpdf.org/>

En nuestro caso se usará solamente para generar el pdf con la factura que se enviará vía email tanto al comprador como al vendedor del producto.



#### 4.3.9 GLIDE.JS

Glide.js es una librería de JavaScript libre de dependencias. Gracias a la integración de Glide nos permite crear un objeto tipo glide que dará la posibilidad de tener un slider funcional y sencillo en nuestra página principal.

```
<script>
  new Glide('.glide', {
    type: 'carousel',
    startAt: 0,
    perView: 3,
    autoplay: 2000
  }).mount();
</script>
```

Figura 4.5 Constructor de la clase Glide



#### 4.3.10 SWEETALARM2

Sweetalarm2 es una librería libre de dependencias de JavaScript que nos permite crear y personalizar alertas *responsives* de forma fácil y sencilla. Gracias a la integración de esta librería nos permite crear mensajes para el usuario adaptados al dispositivo que esté usando.

```
function alertaPositiva($frase, $urlDespues = ""){
    ?>
    <script>
        Swal.fire({
            title: '<?= $frase ?>',
            icon: 'success',
            confirmButtonText: '<?= getT("Ok"); ?>'
        }).then((result) => {
            if (result.isConfirmed) {
                if ('<?= $urlDespues ?>' != '')
                    window.location.href = '<?= $urlDespues ?>';
            }
        });
    </script>
    <?php
}
}
```

Figura 6 Ejemplo de integración de Sweetalarm2 en el proyecto

#### 4.3.11 PASARELA DE PAGO

Para poder llevar a cabo el cobro y envío de los productos de manera segura, respetando la ley de protección de datos. Necesitaremos de una pasarela de pago segura, y se nos presentan varias opciones:



*Figura 7 Logo de Redsýs*

**Redsýs** es una de las plataformas más usadas a nivel mundial en el comercio online y físico. Gracias a ello, el usuario tiene depositada una mayor confianza en esta plataforma.

Uno de los inconvenientes a la hora de integrarlo, es que el sistema de cobro tiene que estar ligado a una entidad bancaria en la que debemos tener cuenta. Hay que negociar con esa entidad bancaria el coste de servicio para implementar esta pasarela de pago.

Dado que la implementación de la pasarela de pago en este proyecto es con fines académicos y no económicos. Se ha descartado esta opción ya que implican costes y el uso de nuestra cuenta bancaria.



*Figura 8 Logo Amazon pay*

Por otro lado, tenemos el reciente **Amazon Pay**, su implementación, aunque no conlleva ningún gasto y su implementación en apariencia, es muy sencilla. Sí que tiene el inconveniente de restringirnos los clientes a solo los que dispongan de una cuenta con Amazon. Así que para no limitar la cantidad de clientes que puedan comprarnos se ha preferido estudiar otra opción.



*Figura 9 Logo PayPal*

Finalmente se ha optado por integrar la plataforma de **PayPal** debido a su fácil integración. Además, la principal ventaja de PayPal es que al no hace falta tener vinculada ninguna entidad bancaria, los clientes no saldrán de dominio web.

Otra ventaja al no disponer de ninguna otra pasarela de pago es que permite hacer el pago aún sin disponer de cuenta de PayPal. El sistema de PayPal no permite el pago sin cuenta si se tiene implementado algún otro método de pago además de PayPal.

## 5. DESARROLLO

Una vez hemos descrito los distintos recursos que se han necesitado, pasamos a la fase de desarrollo. En esta fase abordaremos el proceso de instalación de las diferentes herramientas y módulos, como también se explicará el uso que se hace de ellas.

Este punto será el más importante del proyecto, en él detallamos los sistemas integrados y el proceso que se ha llevado para lograr realizar la aplicación web. Se explicará la base de datos, lenguajes de programación usados y casos de uso de la aplicación.

### 5.1 PLAN DE TRABAJO

El plan de trabajo que se ha llevado a cabo está descrito en la siguiente tabla:

<b>Fases</b>	<b>Duración Estimada (horas)</b>	<b>Tareas (nombre y descripción, obligatorio al menos una por fase)</b>
Estudio previo / Análisis	25	Tarea 1.1: Análisis de Viabilidad
		Tarea 1.2: Formación librería Bootstrap
		Tarea 1.3: Formación Wordpress
		Tarea 1.4: Formación Woocommerce
Diseño / Desarrollo / Implementación	235	Tarea 2.1: Instalación y configuración del sistema para su puesta en marcha
		Tarea 2.2: Creación de la base de datos y tablas
		Tarea 2.3: Diseño gráfico de la web
		Tarea 2.4: Desarrollo de creación de perfiles de usuario
		Tarea 2.5: Implementación Slide de productos
		Tarea 2.6: Desarrollo e implementación de formulario subida de productos
		Tarea 2.7: Implementación de buscador y filtros de búsqueda
		Tarea 2.8: Implementación de valoraciones de usuarios
		Tarea 2.9: Desarrollo de la factura mediante la librería TCPDF
		Tarea 2.1.0: Adaptar la página a multilinguaje
Evaluación / Validación / Prueba	20	Tarea 3.1: Pruebas de correcto Funcionamiento
Documentación / Presentación	20	Tarea 4.1: Memoria Final
		Tarea 4.2: Desarrollo de la presentación

Figura 10 Temporalización del proyecto

## 5.2 ANÁLISIS DE VIABILIDAD Y FORMACIÓN EN WORDPRESS Y WOOCOMMERCE

### 5.2.1 WORDPRESS Y WOOCOMMERCE

Durante la fase de análisis y formación con las posibles herramientas a usar, se llegó a la conclusión de que para realizar un trabajo elaborado y que conlleve un buen proceso de estudio y desarrollo. No se hará uso de Wordpress ni su *plugin* Woocommerce.

Se debe principalmente a los costes económicos que conlleva el uso de Wordpress, además de sumarle también el gasto de los *plugins* y herramientas que queramos integrar.

Por otro lado, el uso de estas herramientas simplificaría la fase de desarrollo de una manera extrema en número de horas. Gracias solamente al plugin de Woocommerce y una plantilla se podría tener una tienda online con su carro de compra, transferencias bancarias, gestor de usuarios, etc... En cuestión de pocas horas.

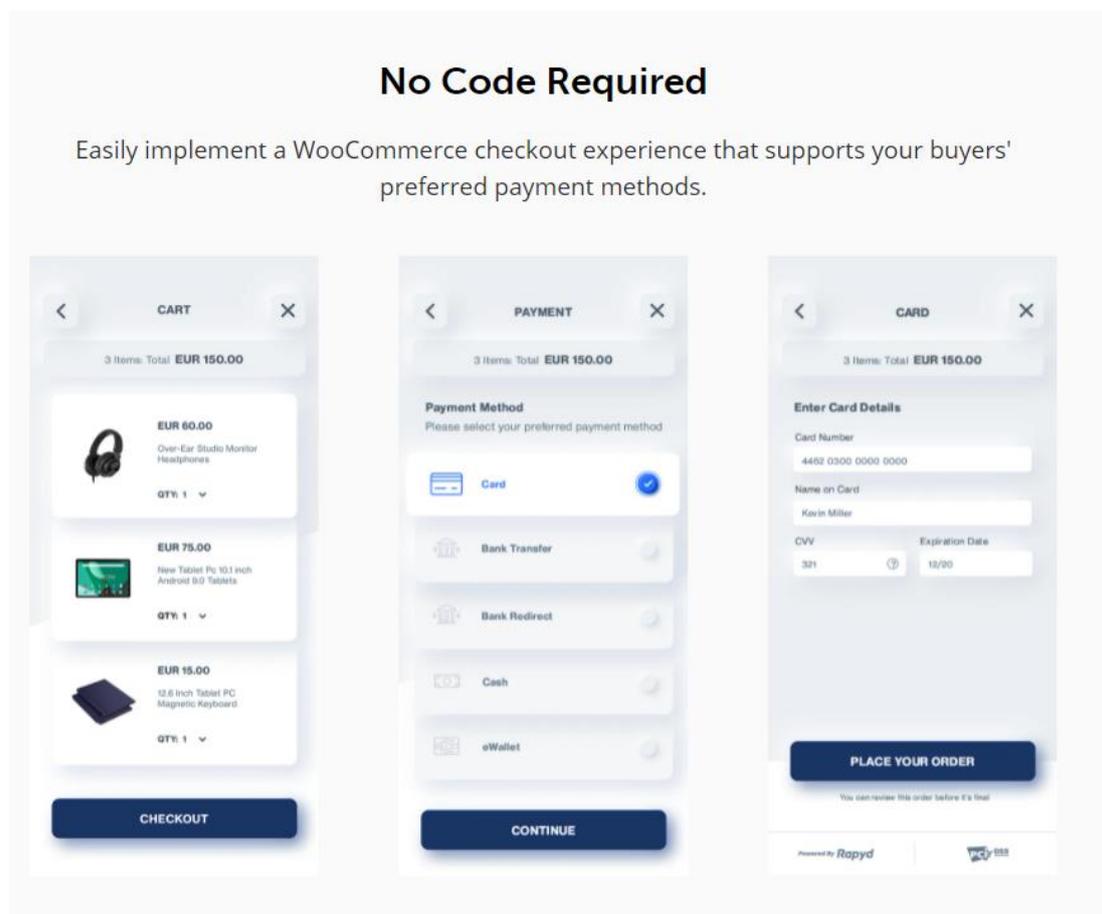


Figura 11 Ejemplo de resultados de la página de Wordpress

Por lo tanto, si nos basamos en el precio que hay que pagar y el poco proceso de desarrollo que se llevaría a cabo. Se ha elegido realizar el proceso de desarrollo de la aplicación escribiendo código e integrando las herramientas gratuitas que encajen en nuestro proyecto.

## 5.3 INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA

### 5.3.1 INSTALACIÓN DE XAMPP

Como hemos descrito en el apartado 4 de recursos necesarios, se va realizar la instalación del servidor de distribución libre de Apache XAMPP. En el que su paquete de instalación incluye PHP 8, MariaDB y Perl.

Gracias a la instalación del paquete XAMPP nos ahorramos la necesidad de tener que instalar cada componente necesario por separado.

Para realizar la descarga del paquete de XAMPP nos dirigiremos a la página oficial, en el siguiente enlace: <https://www.apachefriends.org/es/index.html>.

Nos encontraremos con la siguiente vista en el navegador, seleccionamos la descarga para el sistema operativo que se esté usando, en nuestro caso será la versión de “XAMPP para Windows 8.0.6”.



Apache Friends

Descargar Complementos Alojamiento Comunidad Acerca de   ES

# XAMPP Apache + MariaDB + PHP + Perl

### ¿Qué es XAMPP?

XAMPP es el entorno más popular de desarrollo con PHP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

**Descargar**  
Pulsa aquí para otras versiones

- XAMPP para Windows 8.0.6 (PHP 8.0.6)
- XAMPP para Linux 8.0.6 (PHP 8.0.6)
- XAMPP para OS X XAMPP-VM (PHP 8.0.6)

Figura 12 Página principal de XAMPP

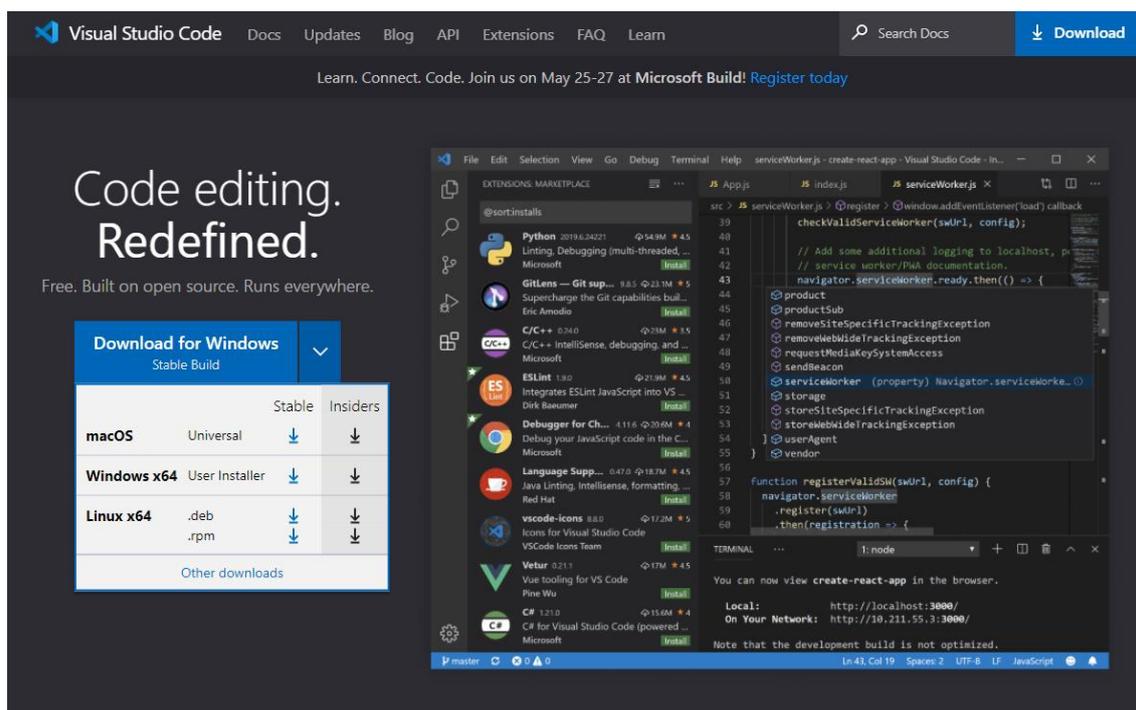
## 5.3.2 INSTALACIÓN DE VISUAL STUDIO CODE

Se ha elegido Visual Studio Code como entorno de desarrollo para escribir el código de la aplicación, debido a su interfaz clara y fácil de manejar. Además, está optimizado para tareas de desarrollador como la capacidad de depurar código o detectar y dar formato al lenguaje que se está usando.

Para su instalación nos dirigimos a su página principal desde este enlace:

<https://code.visualstudio.com/>

Una vez accedemos a la web, seleccionamos el paquete de instalación correspondiente a nuestro sistema operativo, como se aprecia en la siguiente imagen:



IntelliSense

Run and Debug

Built-in Git

Extensions

Figura 13 Página principal de Visual Studio Code

## 5.3.3 INSTALACIÓN DE WORKBENCH

Gracias a la instalación del Paquete de XAMPP ya contamos con un gestor de base de datos "phpmyadmin", pero para la fase de diseño de la base de datos se va a hacer uso de Workbench. Esto se debe a que Workbench nos permite crear la base de datos de una forma más visual y cómoda que el gestor de phpmyadmin. Posteriormente se

importará el SQL de la creación de la base de datos en phpmyadmin para administrarla desde ahí.

Para descargar la aplicación de Workbench nos dirigimos a la siguiente dirección:

<https://www.mysql.com/products/workbench/>

**MySQL Workbench**  
Enhanced Data Migration

Download Now »

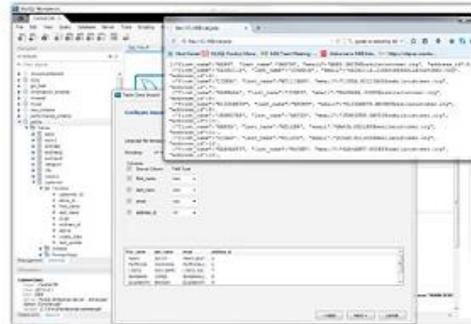


Figura 14 Descarga desde la página de MySQL Workbench

## 5.4 BASE DE DATOS

### 5.4.1 INTRODUCCIÓN

Una fase clave para llevar a cabo el desarrollo de la tienda online es diseñar y crear la base de datos donde almacenaremos la información de los usuarios, anuncios, plataformas y anuncios favoritos del usuario.

Usaremos MySQL para la creación de la base de datos. MySQL usa el sistema cliente-servidor para gestionar las bases de datos. El servidor será donde quedarán almacenados los datos y el cliente es el equipo donde se encuentra instalado y se ejecuta el software de gestión de la propia base de datos. Cada vez que un cliente quiera acceder a algún dato, se tendrán que conectar al servidor y solicitar la información que se necesita. El servidor enviará la respuesta siempre y cuando se tengan los derechos de acceso. Para realizar esta comunicación se hace uso del lenguaje SQL (*Structured query language*).

### 5.4.2 DISEÑO E/R Y CREACIÓN DE LA BASE DE DATOS

Primeramente, levantamos los módulos en XAMPP:

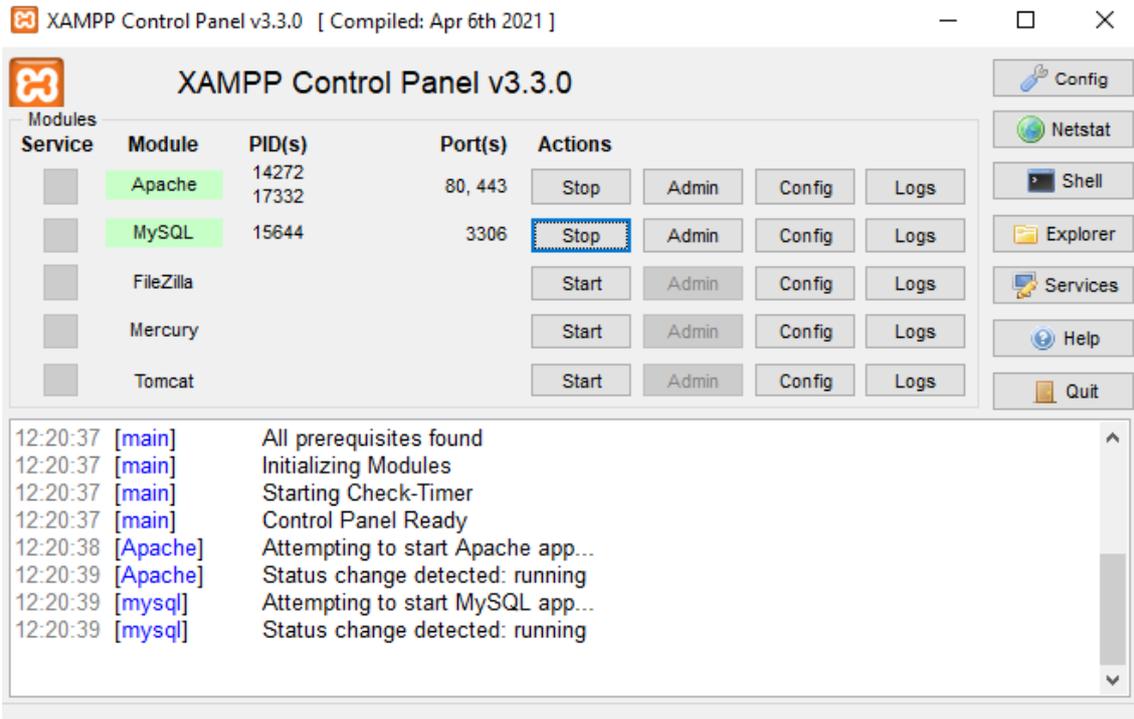


Figura 15 Puesta en marcha de los módulos de XAMPP

Accedemos al menú de gestión de phpmyadmin desde "localhost/phpmyadmin" para crear el nombre de la base de datos, en nuestro caso "bd-tienda" y el usuario "root" que será el administrador.

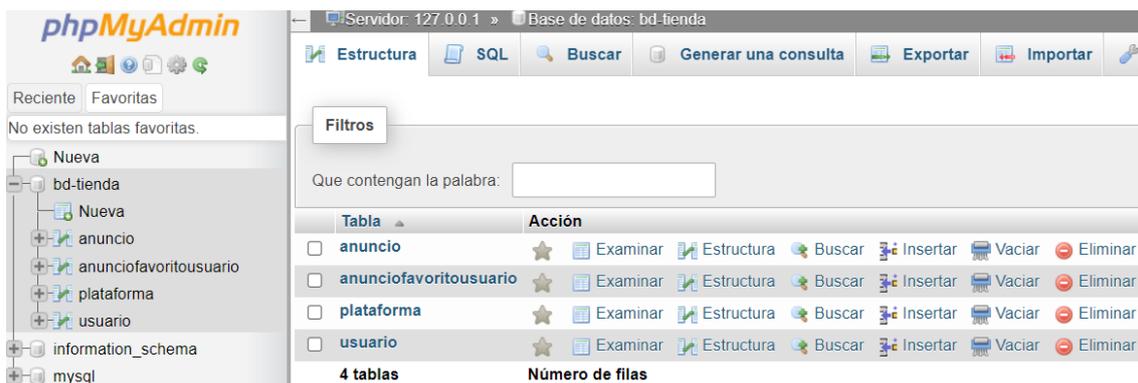


Figura 16 Vista de la base de datos "bd-tienda" desde phpmyadmin

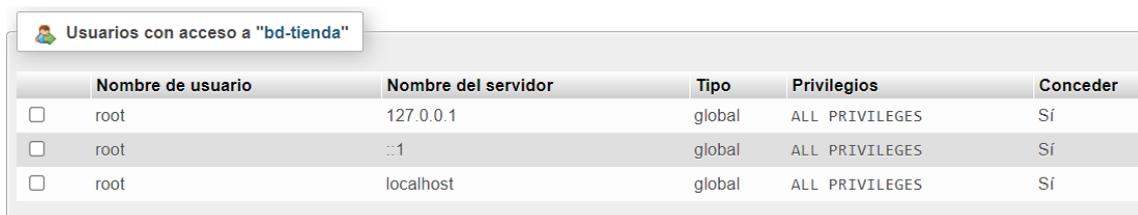


Figura 17 Usuarios creados para la base de datos "bd-tienda"

Como se ha descrito anteriormente haremos uso de la herramienta de Workbench para el diseño y creación de la base de datos. Para posteriormente importarlo en phpmyadmin.

La estructura de la base de datos “bd-tienda” es la siguiente:

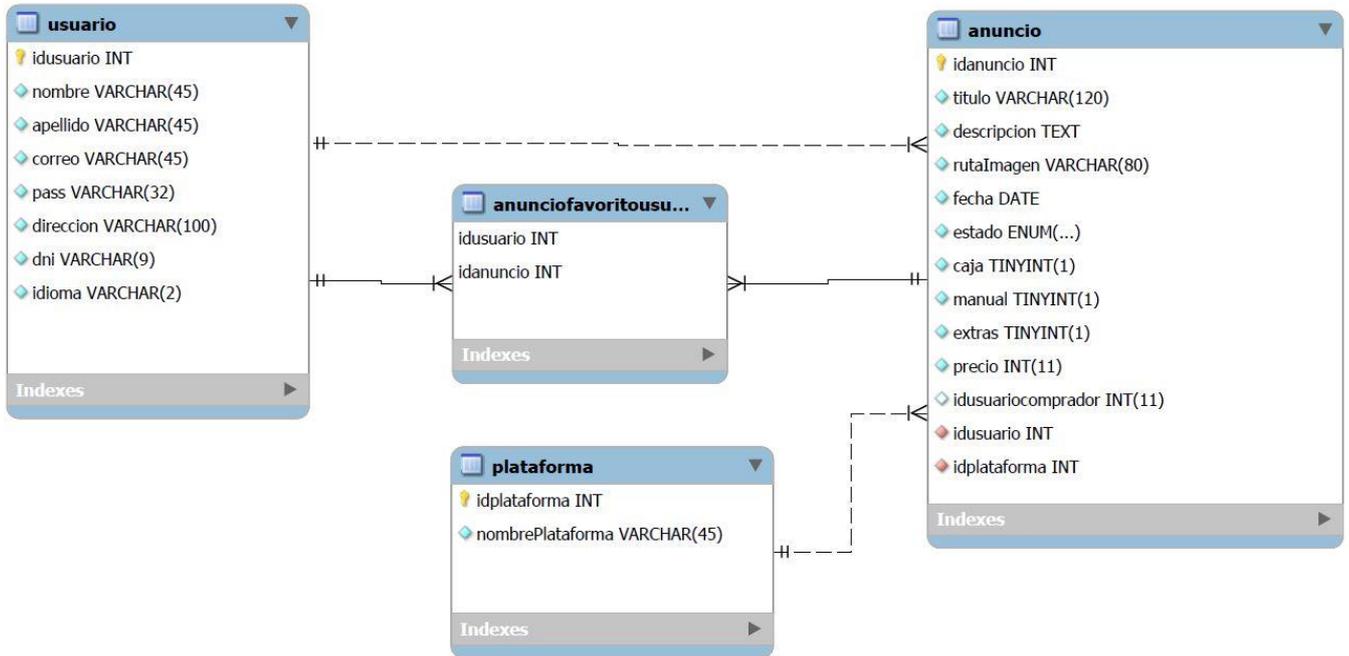


Figura 18 Diagrama E/R de la base de datos "bd-tienda"

Gracias al diagrama de Entidad-Relación podemos observar la estructura de la base de datos de forma clara, así como la relación existente entre ellas.

#### 5.4.3 TABLAS Y CAMPOS

A continuación, explicaremos detalladamente las diferentes tablas y sus campos que componen la base de datos “bd-tienda”.

**Tabla usuario:** En ella almacenaremos la información de los usuarios que se registren en nuestra tienda. Las entradas de los campos las recogeremos mediante el formulario de registro para posteriormente insertarlo en la tabla de usuario.

Esta tabla es de vital importancia, ya que accederemos a ella en multitud de ocasiones para diferentes casos de uso. Como pueden ser el inicio de sesión, generar la factura, anuncios favoritos, etc...

La estructura de la tabla es la siguiente;

CAMPO	TIPO	DESCRIPCIÓN
<b>Idusuario (PK)</b>	int(11)	Identificador del usuario que se autoincrementa. Clave primaria
<b>nombre</b>	varchar(45)	Nombre del usuario
<b>apellido</b>	varchar(45)	Apellido del usuario
<b>telefono</b>	varchar(45)	Nº de teléfono del usuario
<b>correo</b>	varchar(45)	Dirección de correo electrónico
<b>pass</b>	varchar(32)	Contraseña creada por el usuario
<b>direccion</b>	varchar(100)	Dirección física del usuario
<b>dni</b>	varchar(9)	DNI usuario
<b>idioma</b>	varchar(2)	Idioma del usuario

El script para la creación de la tabla es el siguiente:

```
CREATE TABLE `usuario` (
  `idusuario` int(11) NOT NULL,
  `nombre` varchar(45) NOT NULL,
  `apellido` varchar(45) NOT NULL,
  `telefono` varchar(45) NOT NULL,
  `correo` varchar(45) NOT NULL,
  `pass` varchar(32) NOT NULL,
  `direccion` varchar(100) NOT NULL DEFAULT '',
  `dni` varchar(9) NOT NULL,
  `idioma` varchar(2) NOT NULL DEFAULT ''
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figura 19 Script de creación de la tabla usuario

**Tabla anuncio:** En ella almacenaremos los videojuegos que pongan nuestros usuarios a la venta. Está tabla se relaciona con todas las demás, ya que necesitaremos saber de qué usuario es cada anuncio. Por otro lado, los anuncios deben pertenecer siempre a una plataforma. Y a su vez con la tabla “anunciofavoritousuario” de la que hablaremos más adelante.

CAMPO	TIPO	DESCRIPCIÓN
<b>Idanuncio (PK)</b>	int(11)	Identificador del anuncio que se autoincrementa. Clave primaria
<b>titulo</b>	varchar(120)	título del anuncio
<b>descripcion</b>	text	Descripción del anuncio escrita por el usuario
<b>rutaimagen</b>	varchar(80)	Almacenamos la ruta de la imagen subida por el usuario
<b>fecha</b>	date	Fecha que se genera cuando se sube el anuncio
<b>estado</b>	enum('Precintado','Excelente','Muy bueno','Regular','Malo','Muy malo')	Campo donde el usuario indica el estado en el que se encuentra el videojuego del anuncio
<b>caja</b>	tinyint(1)	Valor entre 0 y 1 que indica si el videojuego tiene la caja original o no
<b>manual</b>	tinyint(1)	Valor entre 0 y 1 que indica si el videojuego tiene el manual original o no
<b>extras</b>	tinyint(1)	Valor entre 0 y 1 que indica si el videojuego tiene los extras de la edición o no

<b>precio</b>	int(11)	Valor que indica el precio del anuncio
<b>idusuariocomprador</b>	int(11)	Valor en el que se almacena el id del usuario que ha comprado el artículo
<b>Idplataforma(FK)</b>	int(11)	Clave ajena que identifica la plataforma a la que pertenece el videojuego
<b>Idusuario (FK)</b>	int(11)	Clave ajena que identifica el usuario propietario del anuncio

El script para la creación de la tabla es el siguiente:

```
CREATE TABLE `anuncio` (
  `idanuncio` int(11) NOT NULL,
  `titulo` varchar(120) NOT NULL,
  `descripcion` text NOT NULL,
  `rutaImagen` varchar(80) DEFAULT NULL,
  `fecha` date DEFAULT NULL,
  `idusuario` int(11) NOT NULL,
  `idplataforma` int(11) NOT NULL,
  `estado` enum('Precintado','Excelente','Muy bueno','Regular','Malo','Muy malo') NOT NULL,
  `caja` tinyint(1) NOT NULL,
  `manual` tinyint(1) NOT NULL,
  `extras` tinyint(1) NOT NULL,
  `precio` int(11) NOT NULL,
  `idusuariocomprador` int(11) NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figura 20 Script de creación de la tabla anuncio

```
--
-- Indices de la tabla `anuncio`
--
ALTER TABLE `anuncio`
  ADD PRIMARY KEY (`idanuncio`),
  ADD KEY `fk_anuncio_usuario_idx` (`idusuario`),
  ADD KEY `fk_anuncio_plataforma1_idx` (`idplataforma`);
```

Figura 21 Script de creación de índices para la tabla anuncio

```
--
-- Filtros para la tabla `anuncio`
--
ALTER TABLE `anuncio`
  ADD CONSTRAINT `fk_anuncio_plataforma1` FOREIGN KEY (`idplataforma`) REFERENCES `plataforma` (`idplataforma`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_anuncio_usuario` FOREIGN KEY (`idusuario`) REFERENCES `usuario` (`idusuario`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

Es importante establecer en los filtros para la tabla anuncio la sentencia “DELETE CASCADE” para cuando un usuario decida eliminarse del sistema, también se borren todos los anuncios que le pertenecen.

**Tabla plataforma:** La tabla plataforma nos sirve únicamente para poder ir añadiendo plataformas conforme el mercado vaya evolucionando con salidas de nuevas consolas.

Su estructura es la siguiente:

CAMPO	TIPO	DESCRIPCIÓN
<b>Idplataforma (PK)</b>	int(11)	Identificador de la plataforma que se autoincrementa. Clave primaria
<b>nombrePlataforma</b>	varchar(45)	Nombre de la plataforma

Su script de creación es el siguiente:

```
CREATE TABLE `plataforma` (
  `idplataforma` int(11) NOT NULL,
  `nombrePlataforma` varchar(45) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figura 22 Script de creación de la tabla plataforma

**Tabla anunciofavoritousuario:** En esta tabla nos encargaremos de almacenar el identificador del usuario y el del anuncio que ha seleccionado como favorito. Las combinaciones de estos dos campos forman una tupla única que nos servirá de identificador.

Su estructura es la siguiente:

CAMPO	TIPO	DESCRIPCIÓN
<b>Idanuncio</b>	int(11)	Identificador del anuncio que se autoincrementa. Clave primaria
<b>idusuario</b>	int(11)	Identificador del usuario que se autoincrementa. Clave primaria

Su script de creación es el siguiente:

```
CREATE TABLE `anunciofavoritousuario` (  
  `idanuncio` int(11) NOT NULL,  
  `idusuario` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figura 23 Script de creación de la tabla anunciofavoritousuario

#### 5.4.4 CONEXIÓN CON LA BASE DE DATOS

Hasta ahora hemos definido la estructura de las tablas y la relación entre ellas para la base de datos “bd-tienda”.

Para establecer la conexión con la base de datos, lo primero que hemos hecho ha sido definir una página llamada “config.php”. En ella se definirán las constantes que nos hacen falta para realizar la conexión con la base de datos.

```

config.php
1  <?php
2      define("SERVER", "localhost");
3      define("BD", "bd-tienda");
4      define("USER", "root");
5      define("PASS", "");
6      define("NUM_ELEMENTOS_BUSQUEDA", 2);
7  ?>

```

Figura 24 Página config.php

**SERVER:** Se trata del nombre del servidor donde se encuentra almacenado el servidor de MySQL, en nuestro caso "localhost".

**BD:** Se trata del nombre de la base de datos a la que nos vamos a conectar. Como hemos definido previamente se llamará "bd-tienda".

**USER:** Se trata del nombre del usuario con el que vamos a conectar, en este caso será el usuario administrador "root".

**PASS:** Contraseña de conexión para el usuario elegido. En nuestro caso para mayor comodidad lo hemos dejado en blanco.

Una vez definida la página "config.php" con las constantes pasaremos a crear una nueva página llamada "conexión.php". De esta manera vamos separando las funciones por páginas siguiendo nuestro modelo de desarrollo por componentes.

Dentro de la página "conexión.php" la cual incluye "config.php", definiremos una función que nos devuelva el resultado de la conexión. Esto nos sirve para facilitar todas las veces que necesitaremos conectarnos a la base de datos durante el desarrollo.

Usaremos la función **mysqli\_connect** pasándole por parámetro las constantes anteriormente descritas para que nos devuelva el resultado de la conexión.

Si la variable \$conn lanzamos el error de conexión a la base de datos. Pero si es verdadera necesitamos indicarle el conjunto de caracteres a usar cuando se envíen los datos. Esto lo logramos gracias a la función **mysqli\_set\_charset**.

```

conexion.php
1  <?php
2      include("config.php");
3
4      function getConexionMySQL() {
5          // Crear conexión
6          $conn = mysqli_connect(SERVER, USER, PASS, BD);
7          // Verificación
8          if (!$conn) {
9              die("Error al conectar a la BD: " . mysqli_connect_error());
10         }
11         mysqli_set_charset($conn, "utf8");
12         return $conn;
13     }
14  ?>
15

```

Figura 25 Página conexion.php

## 5.5 DESARROLLO DE LA TIENDA “GAME SECOND LIFE”

El proyecto en sí consiste en crear una tienda online de venta de segunda mano especializada en videojuegos. Para ello vamos a integrar ciertos sistemas que nos ayudará a dotar a la tienda de más funcionalidades. Como puede ser PayPal como pasarela de pago, *iframe* de google docs para el formulario de contacto, Glide.js para crear un *slider* de productos en el *index*, TCPDF para la generación de facturas, etc...

El usuario al acceder a la página web podrá buscar productos, ver sus detalles, rellenar el formulario de contacto, registrarse como usuario, iniciar sesión e incluso comprar sin registrarse gracias a PayPal como explicaremos más adelante.

Todo usuario en nuestra tienda es tratado como un usuario comprador, aún sin registrarse. Solamente cuando el usuario se registra e inicia sesión, se le habilitarán los permisos de vendedor. Con esos permisos, el usuario podrá añadir, modificar y eliminar sus anuncios, editar su perfil y guardar anuncios como favoritos.

De acuerdo al uso actual de los dispositivos móviles, la tienda se ha desarrollado para que sea **responsive**. Esto quiere decir que se adaptará la interfaz para su correcta visualización en todo tipo de dispositivos. Esto se ha logrado gracias al modelo *grind* de Bootstrap donde se establecen clases que determinarán el tamaño en columnas a mostrarse, siendo **col-xl** dispositivos grandes, **col-md** dispositivos medios y **col-sm** dispositivos pequeños como un teléfono móvil.

### 5.5.1 ESTRUCTURA DE “GAMES SECOND LIFE”

Nuestra página principal llamada “index.php es lo primero que verá el usuario al entrar a nuestra aplicación.

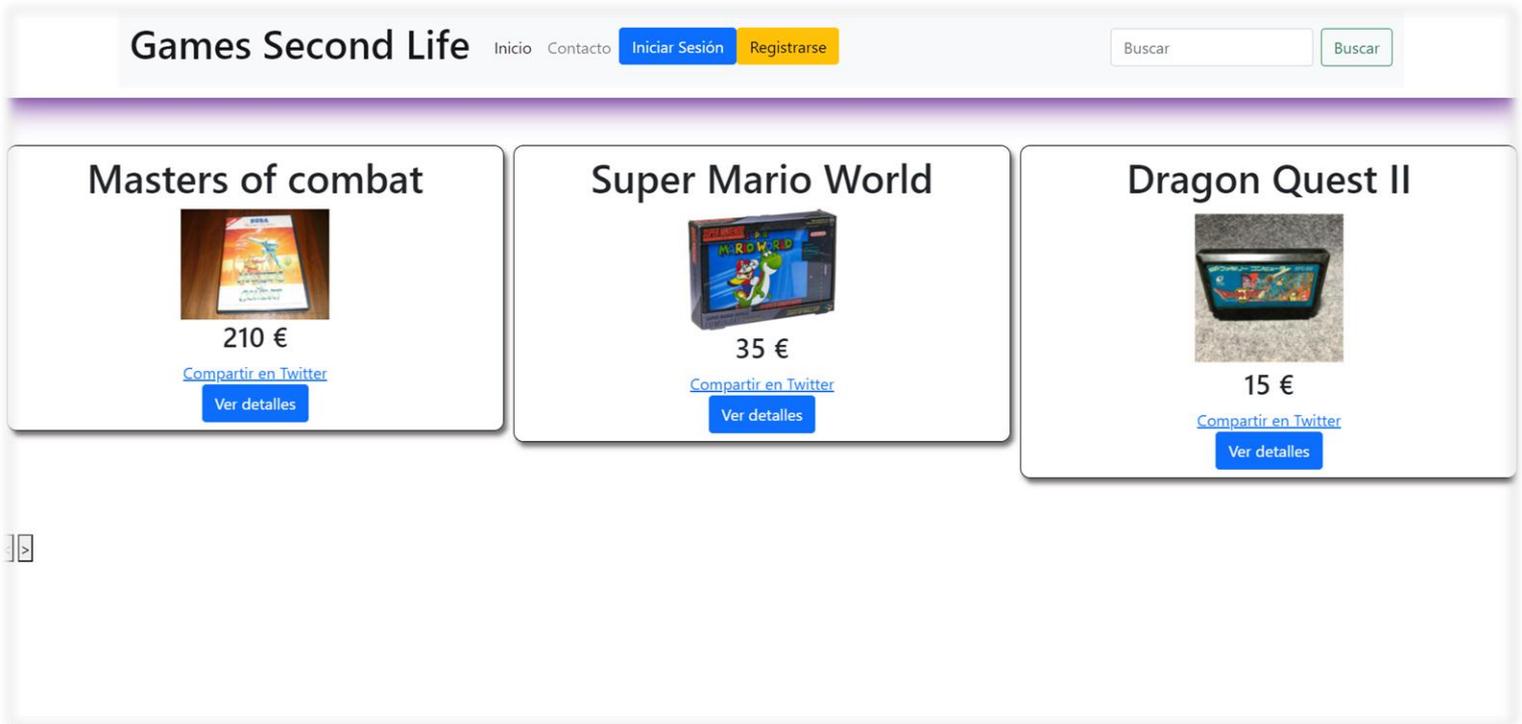


Figura 26 Página principal sin usuario logueado

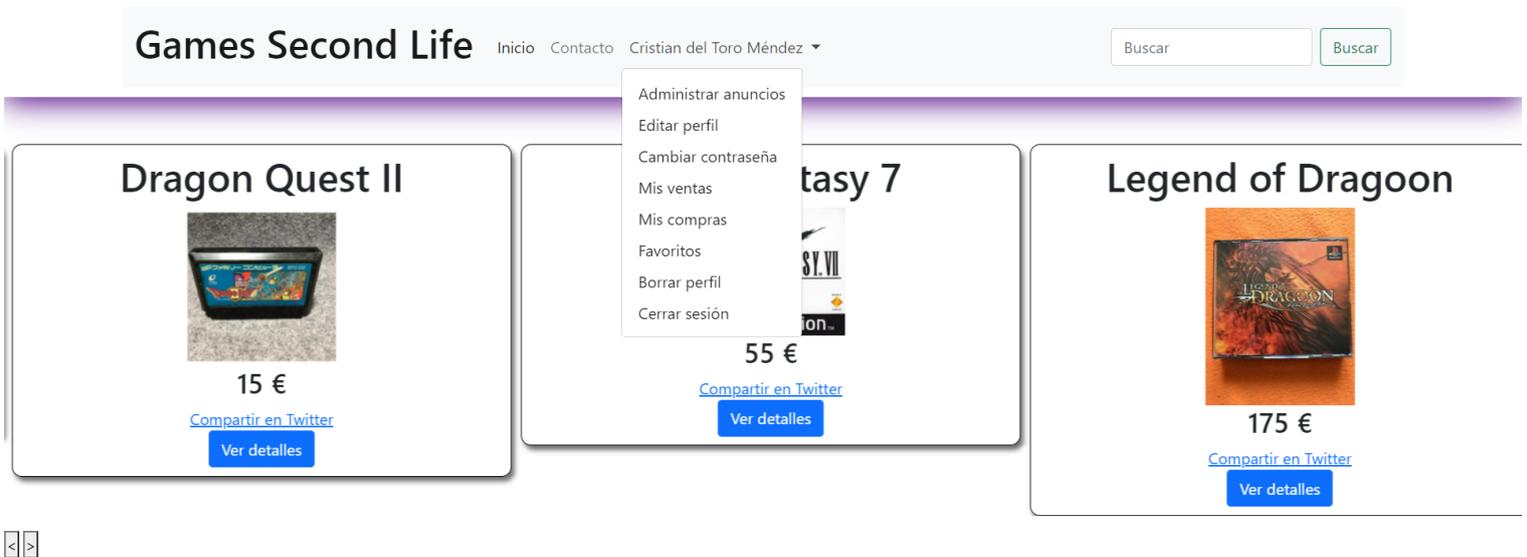


Figura 27 Página principal para usuarios registrados

La barra de navegación la hemos realizado gracias a **Bootstrap** con su clase **navbar**, si el usuario inicia sesión accederá a la barra de navegación para los usuarios registrados en la que al pinchar en su nombre se desplegará un menú con las funciones que puede realizar.

La barra de navegación se colapsará cuando alcance el tamaño establecido por **col-sm** en el **div** de cada elemento. Siendo este su resultado visto desde un **iPhone**:

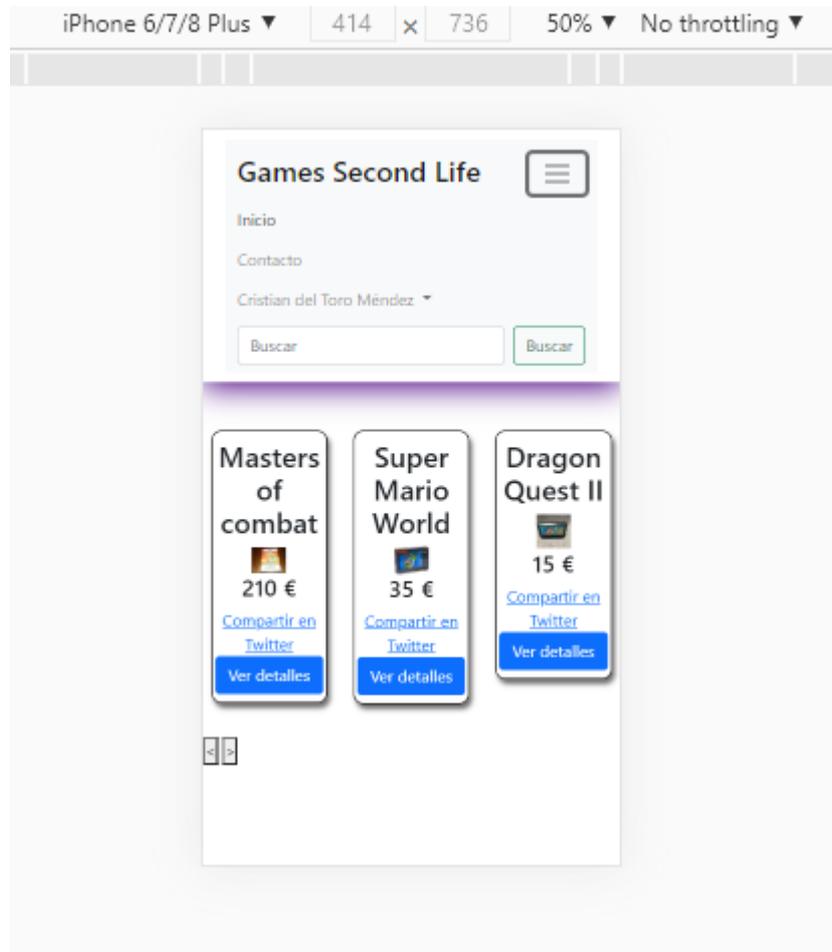


Figura 28 Vista del modelo responsive desde un Iphone 6

El **slider** de productos se ha hecho integrando **Glide.js** en la aplicación. Glide.js nos permite de forma sencilla con su constructor de clase, crearnos un objeto con los parámetros que nosotros queramos y montar un carrusel.

Para que Glide.js detecte los elementos a ingresar en el carrusel debemos encapsular cada elemento en una lista con la clase "**glide\_slide**". La clase "**glide\_bullet**" hace referencia al tipo de flecha que queremos para adelantar a la derecha o izquierda.

Para terminar, realizaremos el script donde le pasaremos los parámetros que queramos para montar el objeto tipo **Glide**. Con el parámetro **type** definimos el estilo de slider que queremos dentro de la oferta de la librería. Con **startAt** indicamos desde donde queremos que empiece. Con **perView** indicamos cuantos elementos queremos que se

vean en el carrusel. Por último, **autoplay** indica la cantidad en milisegundos para adelantarse automáticamente un puesto.

```
<body>
  <?php
    include("cabecera.php");

    $mysql = getConexionMySql();
    $sql = "SELECT * FROM anuncio WHERE (idusuariocomprador = 0)
ORDER BY fecha DESC LIMIT 6";
    $res = mysqli_query($mysql, $sql);
  ?>
  <div class="glide mt-5">
    <div class="glide__track" data-glide-el="track">
      <ul class="glide__slides">
        <?php
          while ($fila = mysqli_fetch_array($res)) {
            echo "<li class='\"glide__slide\"'>";
            mostrarAnuncio($fila, PonerCompartir::SI);
            echo "</li>";
          }
        ?>
      </ul>
    </div>
    <div class="glide__bullets" data-glide-el="controls[nav]">
      <button class="glide__bullet" data-glide-
dir="<">&lt;</button>
      <button class="glide__bullet" data-glide-
dir=">">&gt;</button>
    </div>
  </div>
  <script>
    new Glide('.glide', {
      type: 'carousel',
      startAt: 0,
      perView: 3,
      autoplay: 2000
    }).mount();
  </script>
</body>
```

Si el usuario accede al enlace de Contacto se le dirigirá a la página de “**contacto.php**”, la cual contiene un **Iframe de Google Docs** con un formulario simple de contacto. Dicho formulario se puede actualizar desde la propia página de google y el resultado se verá reflejado en la web sin tener que estar cambiando código, las respuestas serán enviadas a un email creado para este proyecto [tiendatfg2@gmail.com](mailto:tiendatfg2@gmail.com).

Figura 29 Visualización del Iframe de Google en la página contacto.php

La otra función del navegador común a todos los usuarios será el buscador, pero ya que esta función tiene su complejidad la trataremos en el tema 5.5.3.

### 5.5.2 PÁGINAS CABECERA\_HEAD.PHP Y CABEZERA.PHP

La página “**cabezera\_head.php**” la usaremos para ahorrar líneas de código al hacer un *include* de ella en el **header** de las páginas que vayamos creando. También escribimos las referencias a las librerías y estilos que usaremos durante el desarrollo.

```

<meta charset="UTF-8">
<title>Games Second Life</title>
<meta name="descripcion" content="Sitio web para la búsqueda de juegos de segunda mano">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="css/estilo.css"/>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-e0JMYsd53ii+sc0/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-

```

```
JEW9xMcG8R+pH31jmwH6WwP0WintQrMb4s7Z0dauHnUtXwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>
<script src="//cdn.jsdelivr.net/npm/sweetalert2@10"></script>
<script src="js/jquery-3.5.1.min.js"></script>
```

La página de “**cabecera.php**” es el eje principal de nuestra tienda, en ella es donde se define la barra de navegación y por eso la incluimos en todas las páginas de la aplicación. A su vez, ya que la incluiremos en todas las páginas también se definen las funciones que hemos creado para poder hacer uso de ellas en cualquier momento.

Por último, en esta página es donde recogemos y tratamos toda la información enviada por los distintos formularios de la aplicación.

### 5.5.3 BUSCADOR

Empezaremos por crear la página “**buscar.php**” que es donde redirigiremos a los usuarios cuando hagan *click* en el botón **buscar**.

```
<form action="buscar.php" method="post" class="d-flex">
  <input class="form-control me-2" type="search" name="busqueda" id="busqueda" placeholder="Buscar" aria-label="Search">
  <button class="btn btn-outline-success" name="buscar" type="submit">Buscar</button>
</form>
```

Figura 30 Formulario para el caso de uso buscar

Una vez en la página de “**buscar.php**” recogemos lo que ha introducido el usuario y definimos la variable `offset`, que en combinación con la constante `NUM_ELEMENTOS_BUSQUEDA` de la página “**config.php**”, nos servirá para crear una paginación.

```
$busqueda = $_REQUEST["busqueda"];
$offset = 0;
if (array_key_exists("offset", $_REQUEST))
  $offset = $_REQUEST["offset"];
$idUsuario = 0;
if (array_key_exists("idusuario", $_SESSION))
  $idUsuario = $_SESSION["idusuario"];
```

Posteriormente nos conectamos a la base de datos con la función que definimos previamente `getConexionMySQL()` y creamos la sentencia SQL **SELECT** a realizar para la búsqueda que necesita la aplicación.

La búsqueda que necesitamos que nos muestre debe de estar limitada al número de elementos que queremos mostrar por pantalla. También debe estar limitada por la cadena escrita por el usuario que debe de coincidir en parte con algún título de algún anuncio en la tabla “**anuncio**”. Por último necesitamos comprobar si un anuncio es favorito o no del usuario para mostrárselo.

```

$sql = "
SELECT a.*, CASE WHEN afu.idusuario IS NOT NULL THEN 1 ELSE 0 END AS esfavorito
FROM anuncio a LEFT JOIN anunciofavoritousuario afu ON (a.idanuncio = afu.idanuncio) AND (afu.idusuario = ".$idUsuario.")
WHERE (a.titulo LIKE '%{$busqueda}%') AND (a.idusuariocomprador = 0) ORDER BY fecha LIMIT ".(NUM_ELEMENTOS_BUSQUEDA + 1)." OFFSET ".$offset;
$res = mysqli_query($mysql, $sql);
$num = mysqli_num_rows($res);

```

Figura 31 Sentencia SQL que realiza la búsqueda y paginación de la tienda

Primero se solicita que nos devuelva todos los campos de la tabla “**anuncio**” a la que le hemos dado un alias **a**. Le añadiremos un campo llamado “**esfavorito**” en el que su valor será **1** en el caso de que el **LEFT JOIN** cumpla la condición. Esto nos indica que el anuncio es favorito del usuario conectado. En caso de devolvernos un **null** se indica que no es favorito del usuario.

Por último, añadimos la sentencia **WHERE** con la cadena de búsqueda escrita por el usuario, la ordenamos por fecha y limitamos las tuplas a devolver a la constante **NUM\_ELEMENTOS\_BUSQUEDA + 1** y le pasamos el **offset** que sería la posición a seguir cuando haga *click* en “**siguiente**” el usuario.

Con esto lo que queremos lograr es buscar siempre un elemento más de los que queremos mostrar realmente. Recogemos el número de tuplas devuelta por la sentencia y lo almacenamos en la variable “**num**”. Para cuando nos devuelva un número menor que la constante **NUM\_ELEMENTOS\_BUSQUEDA**, sepamos que no hay más anuncios a mostrar por lo que podemos quitar el botón de **siguiente**.

En el **offset** se guarda la posición donde tiene que seguir mostrando los elementos, por lo que gracias a esta variable sabemos que cuando su valor es distinto de 0, no existe botón de **anterior** así que lo quitamos. Con esto logramos un sistema de paginación en la misma sentencia de búsqueda.

```

if ($num > 0) {
    $i = 0;
    while (($fila = mysqli_fetch_array($res)) && ($i < NUM_ELEMENTOS_BUSQUEDA)) {
        mostrarAnuncio($fila, PonerCompartir::SI);
        $i++;
    }
    $offsetAnterior = $offset - NUM_ELEMENTOS_BUSQUEDA;
    $offsetSiguiente = $offset + NUM_ELEMENTOS_BUSQUEDA;
    echo "<br/>";
    if ($offset != 0)
        echo "<a class='\"btn btn-primary mx-2\"' href='\"?offset=\".$offsetAnterior.\"&busqueda=\".$busqueda.\"'>Anterior</a>";
    if ($num > NUM_ELEMENTOS_BUSQUEDA)
        echo "<a class='\"btn btn-primary mx-2\"' href='\"?offset=\".$offsetSiguiente.\"&busqueda=\".$busqueda.\"'>Siguiente</a>";
}
else
    echo '<h2>Sin resultados</h2>';
?>

```

Figura 32 Implementación del sistema de paginación

Dentro de cada iteración a la hora de mostrar el anuncio haremos uso de la función creada en la página “**cabecera.php**”. Con ella simplemente mostramos los resultados en la página.

```
function mostrarAnuncio($fila, $ponerCompartir = PonerCompartir::SI) {
    ?>
    <div class="col-sm-12 col-md-12 videojuego mx-2 py-2 text-center">
        <h1><?= $fila['titulo'] . " " ?></h1>
        ">
        <h3><?= $fila['precio'] . '&nbsp;€' ?></h3>
        <?php
            if (array_key_exists("esfavorito", $fila)) {
                if ($fila["esfavorito"] == 1)
                    echo "<h3>favorito</h3>";
                else
                    echo "<h3>no favorito</h3>";
            }
            if ($ponerCompartir == PonerCompartir::SI) {
                echo "<a class=\"twitter-share-button\"
                    href=\"https://twitter.com/intent/tweet?original_referer=http%3A%2F%2Flocalhost%2F&ref_src=twsrc%5Etfw&tw_p=tweetbutton&url=http%3A%2F%2Flocalhost/tiendaTfg/anuncio.php?id="
                    . $fila['idanuncio'] . "\">.getT("Compartir en Twitter")."</a>";
                echo "<br>";
            }
        ?>
        <button type="button" class="btn btn-
primary" onclick="window.open('anuncio.php?id=<?= $fila['idanuncio'] ?>', '_blank');"><?= get
T("Ver detalles") ?></button>
    </div>

    <?php
}
```

En la función lo único que hacemos es definir de qué manera deseamos que se muestre cada campo. Comprobamos que la variable que añadimos anteriormente en la sentencia **SQL “esfavorito”** para indicar si es favorito del usuario o no.

Integramos el botón de compartir en twitter por si el usuario desea compartir el anuncio. Indicamos la **url** a compartir al final de la sentencia de la etiqueta **href** de Twitter.

A su vez cada anuncio mostrado dispondrá de un botón ver detalles que llevará al usuario a la página “**anuncio.php**”. Pasamos el identificador del anuncio en la url de la página de cada anuncio.

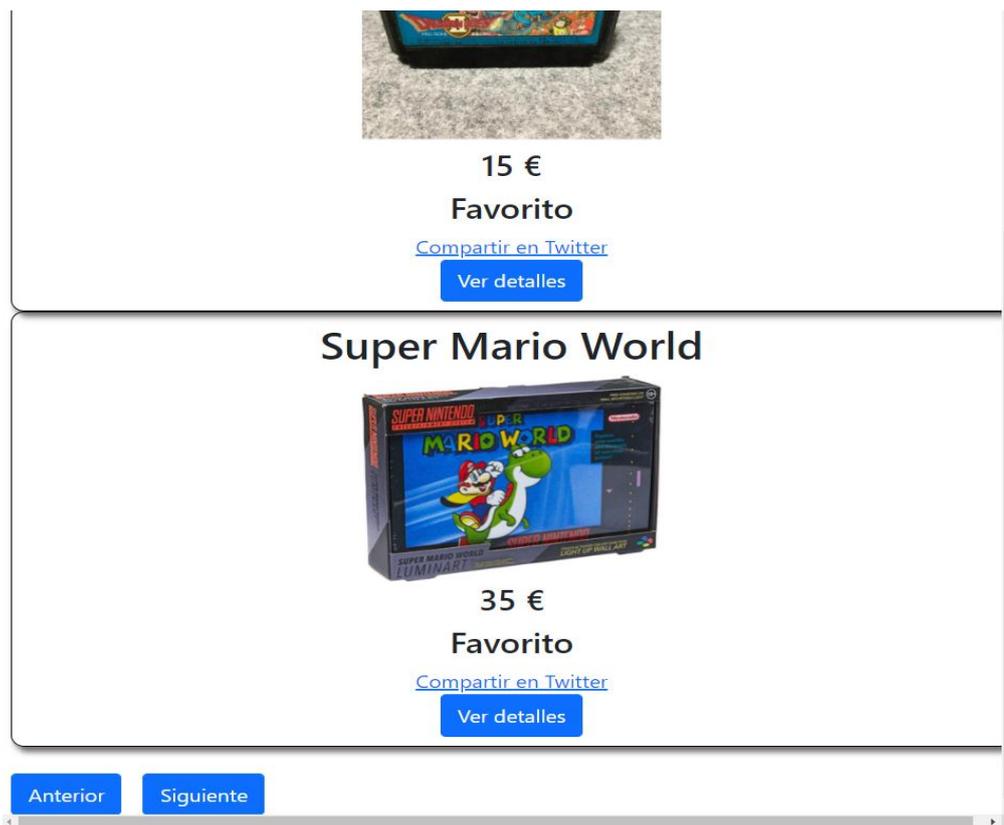


Figura 33 Ejemplo de visualización de anuncios gracias al método mostrarAnuncio()

#### 5.5.4 REGISTRAR Y EDITAR USUARIO

##### **Registro:**

Se necesitará un sistema de *login*, ya que los usuarios necesitarán privilegios para usar las funciones de vendedor. Para ello se va a crear un formulario en html/php donde se recogerán los datos a almacenar en cada campo de la tabla usuario.

El usuario al hacer *click* en el botón “**Registrarse**” se le redirigirá a la página “**registrar.php**” donde se encuentra el formulario de registro.

The image shows a registration form with the following fields and values:

- Correo:** vagnardpl4yers@gmail.com
- Nombre:** John
- Apellido:** Doe
- Teléfono:** 674889966
- Dirección:** Calle Leon Y Castillo 88, 3a, 35009, Las palmas de G.C
- Contraseña:** ....
- Dni:** 8888752h
- Idioma:** Español

At the bottom left of the form is a yellow button labeled "Aceptar".

Figura 34 Formulario de registro en la tienda

Todos los campos del formulario son requeridos y validados para su correcto almacenamiento en la tabla “usuarios”. Por ejemplo, para la validación del campo “**Correo**” se ha hecho uso de la siguiente expresión regular para establecer un patrón de caracteres que nos valide si el correo es correcto o no. Por último, se comprueba que no exista otro correo igual en la tabla “**usuario**”.

```
<script type="text/javascript">
  function validarEmail(elemento) {
    var texto = document.getElementById(elemento.id).value;
    var regex = /^[-\w.%.+]{1,64}@(?:[A-Z0-9-]{1,63}\.){1,125}[A-Z]{2,63}$/i;

    if (!regex.test(texto)) {
      document.getElementById("resultado").innerHTML = "Correo inválido";
    } else {
      document.getElementById("resultado").innerHTML = "";
    }
  }
</script>
```

Figura 35 Regex usado en la comprobación del correo

Una vez el usuario introduce todos los campos y hace *click* en el botón “Aceptar”. Se le redirigirá a la página principal y visualizará una alerta positiva en caso de que el INSERT en la tabla de “usuario” fuese exitosa o negativa en caso contrario. Todas las alertas del proyecto se han generado integrando la librería “Sweetalert2” que detallaremos en el tema 5.5.12 Funciones auxiliares.



Figura 36 Mensaje de "Sweetalert2" indicando el éxito del registro

### Editar perfil:

Se reutilizará “registrar.php” con el propósito de que sirva también como página de edición del perfil de usuario. Cuando se redirige al usuario a la página de “registrar.php” desde el botón del menú de usuario “Editar perfil” recogeremos el id del usuario *logueado*. Por lo que nada más entrar en la página comprobaremos si el campo “idusuario” existe y de ser el caso lo almacenamos.

Necesitaremos hacer una consulta para recoger sus datos e introducirlos en el formulario, de modo que el usuario solamente tenga que cambiar el campo que desee.

Al dotar de dos funciones diferentes al mismo formulario, necesitamos una variable que guarde el valor de la operación a realizar. Una vez se haga *click* en aceptar, para el caso de editar perfil la variable operación tendrá el valor “editarPerfil” y se pasará en un campo hidden al “index.php”. Por último, se ejecutará la sentencia SQL UPDATE que actualizará la tupla correspondiente al usuario logueado en la tabla “usuario”

Hacer esto conlleva que tengamos que crear una variable por cada campo del formulario inicializadas a cadena vacía. A su vez asignamos el valor “registrar” en la variable operación para que en caso de no haber usuario logueado se considere un registro y realice el INSERT.

### 5.5.5 INICIAR SESIÓN Y CERRAR SESIÓN

#### Inicio de Sesión:

Una vez el usuario se ha registrado correctamente deberá hacer *click* en el botón de “Iniciar sesión” para redirigirle a la página “iniciar\_sesion.php”. El usuario encontrará un sencillo formulario donde se le solicita la dirección de correo electrónico con la que se registró y la contraseña creada para la cuenta.

Dicho formulario lo redirigirá a la página principal donde el campo *hidden* “operacion” tendrá el valor de “iniciarSesion”. Realizaremos la comprobación del correo y la contraseña correspondan con alguna tupla de la tabla “usuario”. En caso de ser correcto almacenaremos la información que necesitemos para las funciones de la web en variables \$\_SESSION y mostramos la alerta indicando que se ha conectado correctamente.

```
if (array_key_exists("operacion", $_REQUEST)) {
    if (strcmp($_REQUEST["operacion"], "iniciarSesion") == 0) {
        $correo = $_REQUEST["correo"];
        $clave = $_REQUEST["clave"];
        $mysql = getConexionMySQL();
        $sql = "SELECT * FROM usuario WHERE correo = '" . $correo . "' AND pass = '" . $clave . "'";
        $resExiste = mysqli_query($mysql, $sql);
        $fila = mysqli_fetch_array($resExiste);
        if (mysqli_num_rows($resExiste) > 0) {
            $_SESSION["idusuario"] = $fila[0];
            $_SESSION["nombrecompleto"] = $fila[1] . " " . $fila[2];
            $_SESSION["correo"] = $correo;
            $_SESSION["idioma"] = $fila["idioma"];
            alertaPositiva("Has iniciado sesión correctamente");
        } else {
            alertaError("Algún dato introducido es incorrecto");
        }
    }
}
```

Figura 37 Inicio de sesión

### Cerrar Sesión:

La función de cerrar sesión es muy sencilla, para ello cuando el usuario haga *click* en cerrar sesión se le redirigirá a la página “salir.php”.

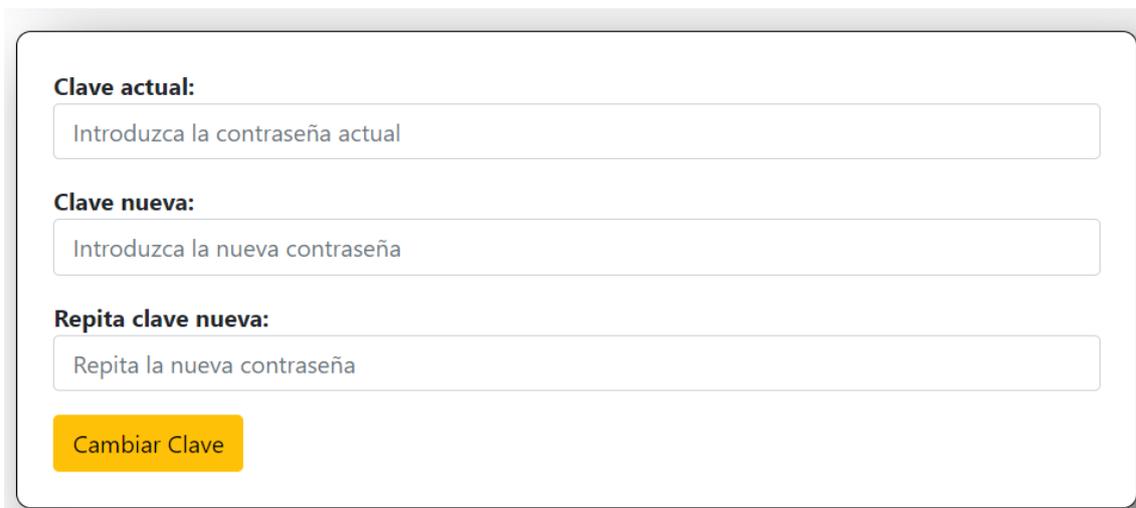
```
salir.php
1 <?php
2     session_start();
3     session_destroy();
4     header("Location: index.php");
5 ?>
```

Figura 38 Página salir.php

Se destruye la sesión iniciada y redirigimos a la página principal.

### 5.5.6 CAMBIAR CONTRASEÑA

Respetando nuestro modelo por componentes crearemos una página “**cambiarclave.php**” intentando desgranar lo más posible el proyecto por funciones. Cuando el usuario haga *click* en “**Cambiar contraseña**” se le presentará el siguiente formulario.



El formulario de cambio de contraseña está contenido en un recuadro con un borde gris. Incluye tres campos de entrada de texto con el texto de ayuda "Introduzca la contraseña actual", "Introduzca la nueva contraseña" y "Repita la nueva contraseña". Debajo de los campos hay un botón amarillo con el texto "Cambiar Clave".

Figura 39 Formulario de cambio de contraseña

En el caso para cambiar la contraseña el valor de nuestra variable operación será “**cambiarclave**” y la pasaremos por un campo hidden. Como siempre, redirigimos al usuario a la página principal donde “**cabezera.php**” se encargará de realizar la comprobación de la contraseña antigua. En caso de que la antigua contraseña sea

correcta y la nueva contraseña escrita en los dos campos coincidan, se realizará el **UPDATE** del campo “**pass**” de la tabla “**usuario**”

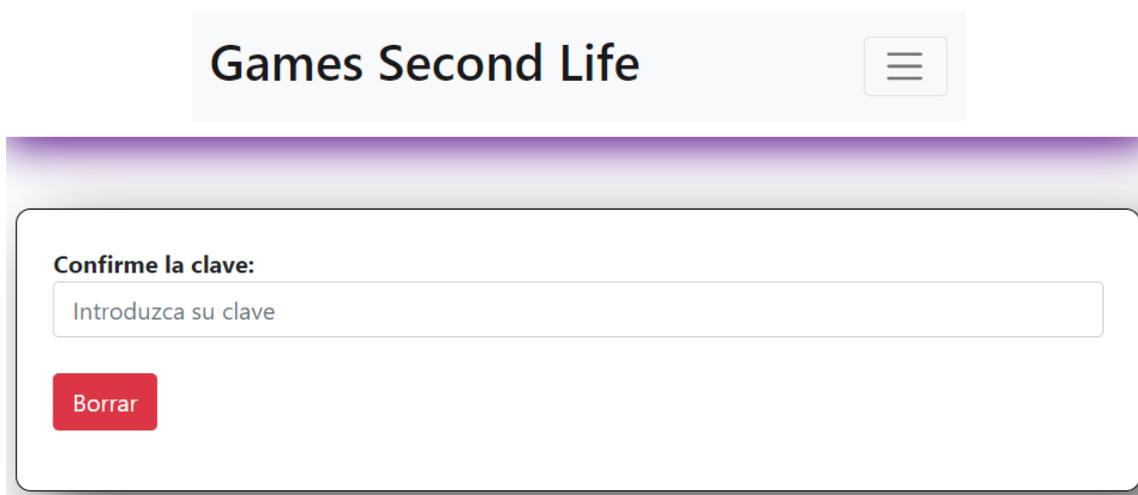
```
else if (strcmp($_REQUEST["operacion"], "cambiarclave") == 0){
    $claveActual = $_REQUEST["claveactual"];
    $claveNueva = $_REQUEST["clavenueva"];
    $claveNueva2 = $_REQUEST["clavenueva2"];
    $mysql = getConnectionMySQL();
    $sql = "UPDATE usuario SET pass = '". $claveNueva.'"
    WHERE (idusuario = '".$_SESSION["idusuario"]."') AND (''. $claveNueva.'" = '". $claveNueva2.'"')
    AND (''. $claveActual.'" = pass)";
    mysqli_query($mysql, $sql);
    if (mysqli_affected_rows($mysql) == 1) {
        alertaPositiva("La contraseña se ha cambiado con éxito");
    }
    else{
        alertaError("Error al cambiar la contraseña");
    }
}
```

Figura 40 Lógica para realizar el cambio de contraseña

### 5.5.7 BORRAR PERFIL

Tenemos que dar la posibilidad a los usuarios de la tienda a darse de baja. Gracias al borrado en cascada que hemos establecido en la base de datos, sabemos que cuando un usuario se borre se eliminarán también todos sus anuncios.

Para evitar que el usuario borre su perfil por error, cuando el usuario haga *click* en el botón de borrar perfil le redirejiremos a la página “**borrarperfil.php**”.



The screenshot shows a web interface for 'Games Second Life'. At the top, there is a navigation bar with the site name and a hamburger menu icon. Below this, a modal form is displayed with the title 'Confirme la clave:'. The form contains a single text input field with the placeholder text 'Introduzca su clave'. Below the input field is a red button labeled 'Borrar'.

Figura 41 Formulario borrar usuario

En ella le solicitaremos la contraseña y al presionar en el botón borrar realizaremos la comprobación. Si la comprobación de la clave es positiva realizaremos la sentencia SQL

**DELETE** correspondiente al usuario logueado. Mostramos la alerta indicando el éxito o no de la eliminación del perfil y en caso de borrarse también lo redirigimos a “salir.php” para cerrar la sesión.

```
} else if (strcmp($_REQUEST["operacion"], "borrarPerfil") == 0) {  
    //BORRAR PERFIL  
    $clave = $_REQUEST["clave"];  
    $mysql = getConexionMySQL();  
    $sql = "DELETE FROM usuario WHERE pass = '".$_REQUEST["clave']."' AND idusuario = '".$_SESSION["idusuario"]";  
    $resultado = mysqli_query($mysql, $sql);  
    if ($resultado) {  
        alertaPositiva("Usuario borrado correctamente", "salir.php");  
    }  
    else{  
        alertaError("Error al borrar el usuario");  
    }  
}
```

Figura 42 Lógica para borrar el usuario

### 5.5.8 AÑADIR, EDITAR Y ELIMINAR ANUNCIOS

#### Añadir anuncios:

Debemos de brindar al usuario de un espacio donde poder ver los anuncios que ha publicado y añadir otros nuevos. De modo que cuando el usuario haga *click* en “Administrar anuncios” le redirigimos a la página que llamaremos “perfil.php”.

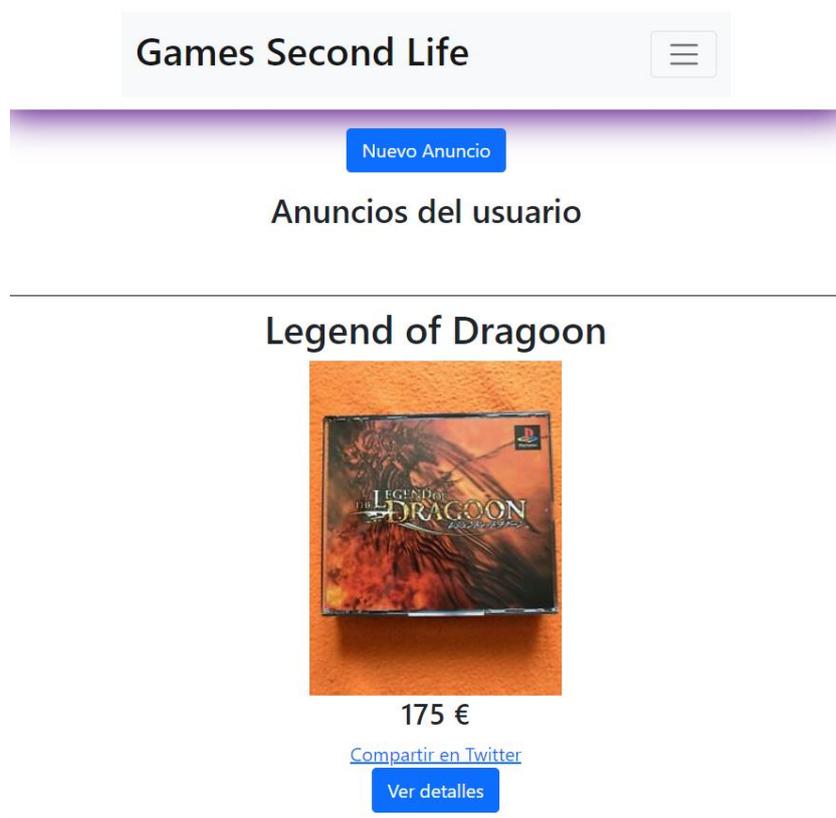
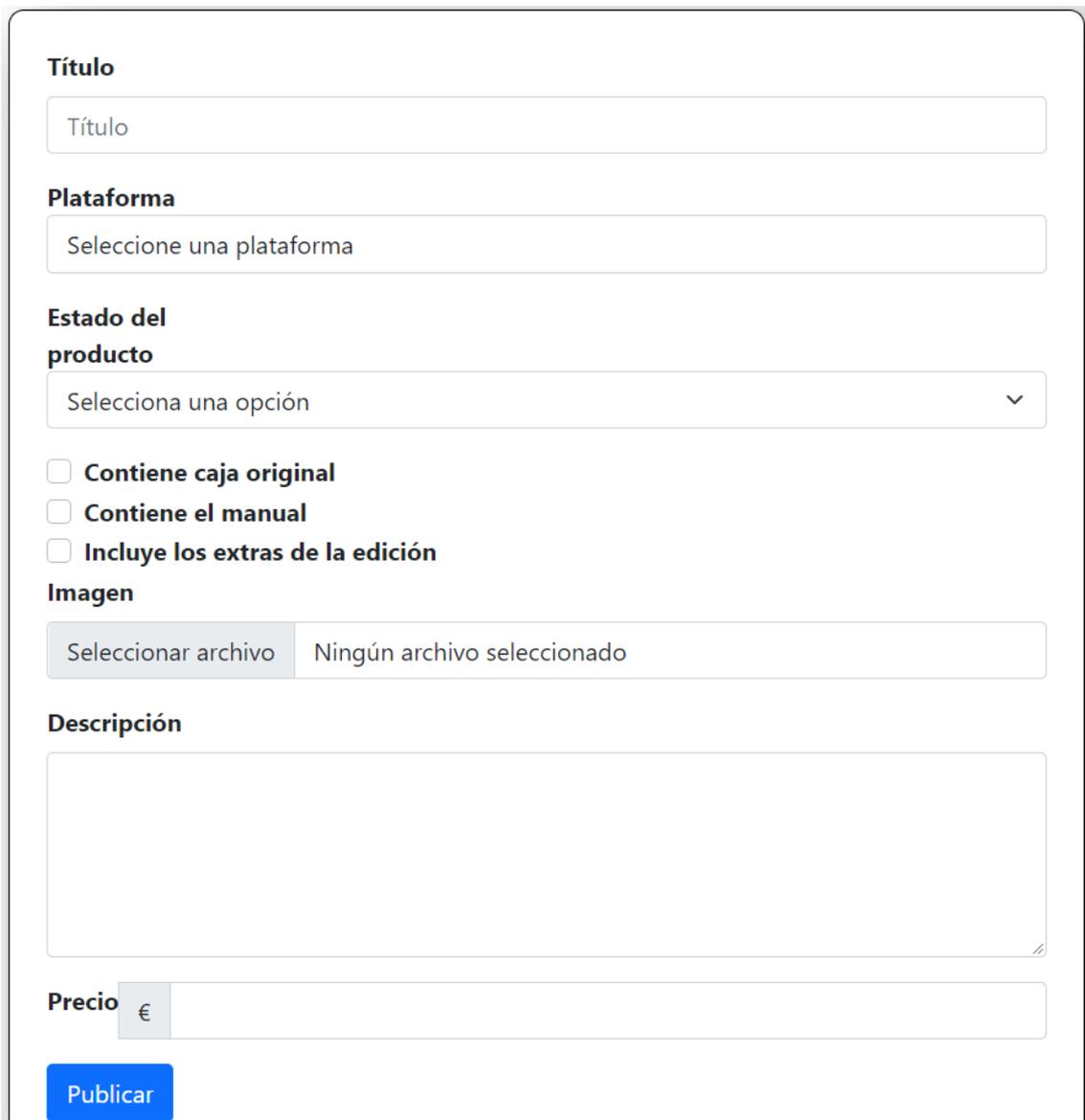


Figura 43 Página perfil.php

Una vez dentro de la página “**perfil.php**”, el usuario puede hacer click en el botón añadir anuncio si quiere vender algún videojuego nuevo. Una vez haga *click*, le redirigiremos a la página “**publicar.php**” que contendrá el formulario de entrada para subir un nuevo videojuego a la tienda.



Formulario de publicación de anuncios con los siguientes campos:

- Título:** Campo de texto con el placeholder "Título".
- Plataforma:** Campo de selección con el placeholder "Seleccione una plataforma".
- Estado del producto:** Campo de selección con el placeholder "Selecciona una opción" y un ícono de flecha hacia abajo.
- Características:** Tres casillas de verificación:
  - Contiene caja original
  - Contiene el manual
  - Incluye los extras de la edición
- Imagen:** Campo de selección de archivos con el placeholder "Seleccionar archivo" y "Ningún archivo seleccionado".
- Descripción:** Área de texto grande para la descripción del producto.
- Precio:** Campo de texto con el símbolo "€" a la izquierda.
- Publicar:** Botón azul con el texto "Publicar".

Figura 44 Formulario de publicación de anuncios

El formulario debe de ser completamente cumplimentado y la única comprobación que se realizará será a la hora de importar la imagen del videojuego a publicar.

Se solicitará que el archivo de imagen sea de tipo **JPEG**, así evitamos formatos de mayor resolución que son más pesados en bytes para almacenar. Como los nombres de las imágenes suelen dejarse por defecto y no podemos determinar el tamaño de la cadena, convertiremos el nombre usando el cifrado md5. Por último, almacenamos la imagen dentro de la carpeta “**img**” de nuestro servidor y también la ruta de destino para realizar el INSERT.

Se ha creado una función auxiliar para realizar la publicación del anuncio creando la sentencia SQL INSERT. Con el propósito de brindar una lectura más clara del código y facilitar su uso. El último valor corresponde al **idusuariocomprador** que siempre será 0 cuando se publique un anuncio, ya que indica que el videojuego está a la venta.

```
function publicarAnuncio($titulo, $plataforma, $imagen, $descripcion, $id, $estado, $caja, $manual, $extras, $precio) {
    $mysql = getConexionMySQL();

    if ($imagen["type"] != "image/jpeg") {
        alertaError("Tiene que subir un archivo con extensión jpg");
    } else {
        $rutaOrigen = $imagen["tmp_name"];
        $rutaDestino = "img/" . md5($imagen["tmp_name"]) . ".jpeg";
        $resultado = move_uploaded_file($rutaOrigen, $rutaDestino);
        if ($resultado) {
            $sql = "INSERT INTO anuncio VALUES (null, '$titulo', '$descripcion',
                . " '$rutaDestino', NOW(), '$id', '$plataforma', '$estado', '$caja', '$manual', '$extras', '$precio',0)";
            $res = mysqli_query($mysql, $sql);

            if ($res) {
                alertaPositiva("Anuncio publicado correctamente");
            } else {
                alertaError("Hubo un error al subir el anuncio");
            }
        } else {
            alertaError("Hubo un error al subir el anuncio");
        }
    }
}
```

Figura 45 Función publicarAnuncio()

### Editar anuncio:

Además de la misma manera que usamos anteriormente la página de “registrar.php” para editar perfil, usaremos esta para editar los productos publicados de la misma forma que hicimos con el perfil.

Para ello la variable “operacion” tendrá un valor de “actualizarAnuncio” en el campo hidden realizando el **UPDATE**. O en caso de tratarse de un nuevo anuncio su valor será de “publicarAnuncio”, realizando el **INSERT** en la base de datos.

Para que un usuario pueda editar un anuncio simplemente tendrá que acceder a la página de detalles del anuncio, la cual llamaremos “anuncio.php”. En ella simplemente se muestra una vista más detallada del videojuego en cuestión, a la vez que detectamos si un anuncio pertenece o no al usuario conectado.

Si el anuncio no pertenece al usuario logueado le aparecerá la opción de **comprar**. Mientras que, si el anuncio pertenece al usuario, le saldrá la opción de **editar** o **borrar** el anuncio.



Figura 46 Opciones como propietario de un anuncio

#### Borrar anuncio:

Cuando el usuario quiera borrar un anuncio, simplemente tendrá que hacer *click* en el botón borrar del anuncio en cuestión. Se le abrirá una ventana emergente que hemos realizado gracias a la integración de **Sweetalert2**, donde se le preguntará si realmente quiere borrar el anuncio.

```
function clickBorrar() {  
  Swal.fire({  
    title: '¿Borrar?',  
    showDenyButton: true,  
    showCancelButton: false,  
    confirmButtonText: 'Sí',  
    denyButtonText: 'No'  
  }).then((result) => {  
    if (result.isConfirmed) {  
      window.location.href = 'borraranuncio.php?idanuncio=<?= $id ?>';  
    }  
  });  
}
```

Figura 47 Ventana emergente para borrar un anuncio

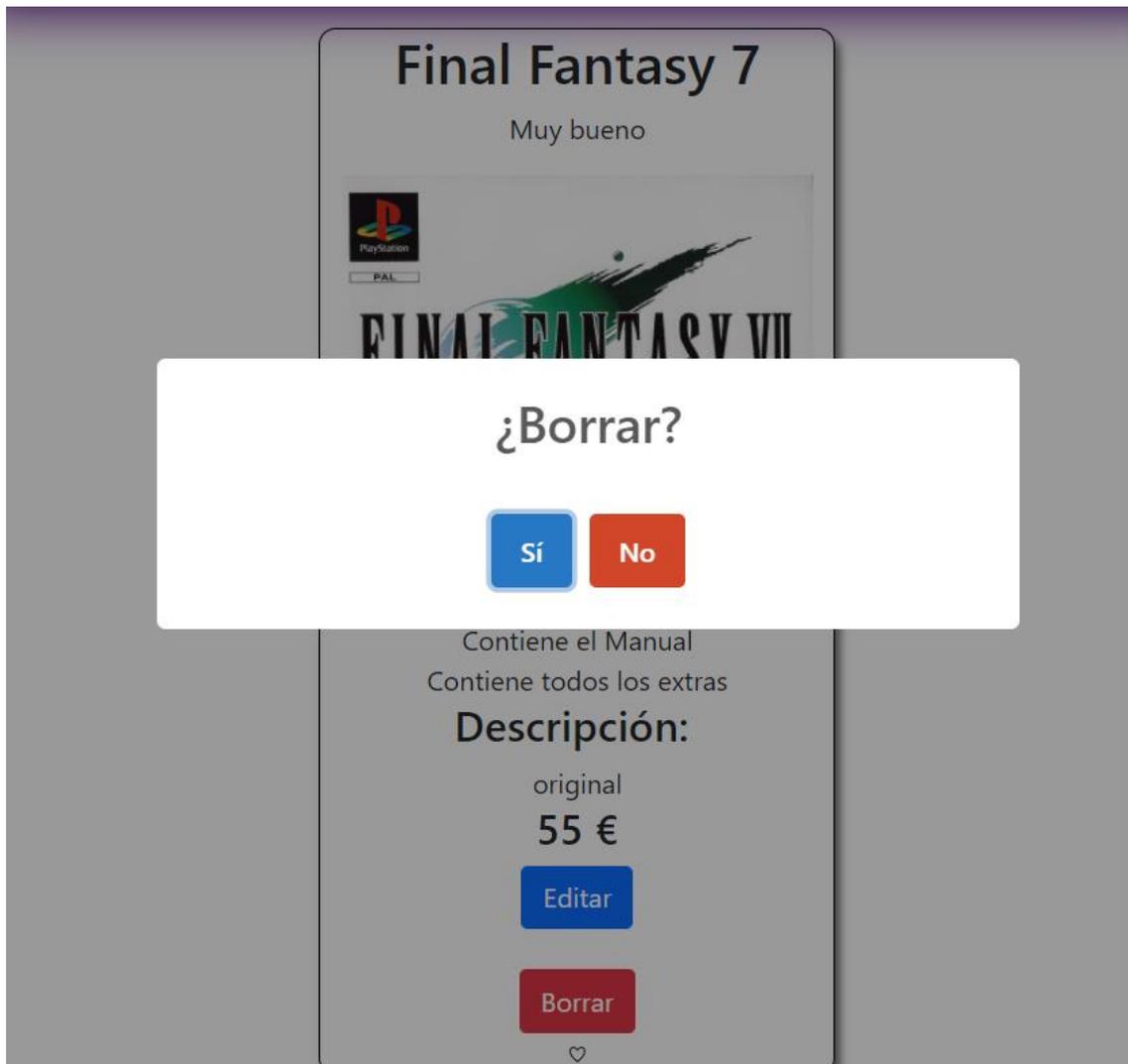


Figura 48 Vista de la ventana emergente para borrar un anuncio

Una vez el usuario confirma que quiere borrar el anuncio se le redirigirá a la página “**borraranuncio.php**” junto al identificador del anuncio en la url. Donde realizaremos la sentencia SQL **DELETE** para el anuncio indicado.

```

<body>
  <?php
    include("cabecera.php");

    $idAnuncio = 0;
    if (array_key_exists("idanuncio", $_REQUEST))
      $idAnuncio = $_REQUEST["idanuncio"];
    if ($idAnuncio != 0) {
      $mysql = getConexionMySQL();
      $sql = "DELETE FROM `anuncio` WHERE idanuncio = ".$idAnuncio;
      $resultado = mysqli_query($mysql, $sql);
      if ($resultado) {
        alertaPositiva("Anuncio borrado correctamente");
      }
      else{
        alertaError("Error al borrar el anuncio");
      }
    }
  ?>
</body>

```

Figura 49 Lógica para borrar un anuncio en la base de datos

### 5.5.9 COMPRAR Y GENERAR FACTURA

El caso de uso de comprar un videojuego y generar la factura fue el más delicado de todos. La Api de pasarela de pago, en nuestro caso “**PayPal**”, da muy poco margen de modificación, ya que integra sus propios protocolos de seguridad y los datos se deben ingresar como se define en la Api.

Por otro lado, la cantidad de opciones y parámetros de la librería TCPDF para darle estilo al pdf resultante puede resultar abrumadora. Se ha tratado de hacer una factura sencilla en la que se impriman los datos necesarios para que el vendedor pueda enviar el artículo al comprador.

#### **Comprar:**

Como se observa en la figura 50, el usuario al acceder a los detalles de un anuncio del que no es propietario podrá optar a comprarlo. Cuando haga *click* en comprar se le redirigirá a la página “**comprar.php**”.

**Final Fantasy 7**  
Muy bueno



PAL



**FINAL FANTASY VII**

SQUARESOFT



Contiene la caja original  
Contiene el Manual  
Contiene todos los extras

**Descripción:**

original  
**55 €**

**Comprar**

♡

Figura 50 Ejemplo detalles del anuncio del que no se es propietario

Lo primero que necesitamos en dicha página será integrar la pasarela de pago de **PayPal**. Para ello se necesitará una cuenta “**Sandbox**” de Paypal, lo que sería una cuenta de desarrollador. Para crearnos la cuenta debemos dirigirnos a la siguiente url:

<https://developer.paypal.com/home>

# Build Modern Commerce with PayPal

Developer tools and resources to integrate PayPal Commerce Platform.  
 Learn more about our mission to improve the PayPal developer experience.

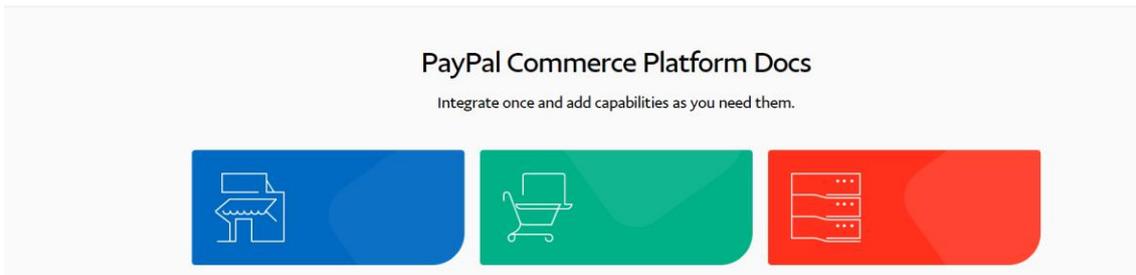


Figura 51 Página para desarrolladores de PayPal

Posteriormente iniciamos sesión con la cuenta que hayamos creado y visualizaremos el panel de control del entorno de pruebas de Paypal.

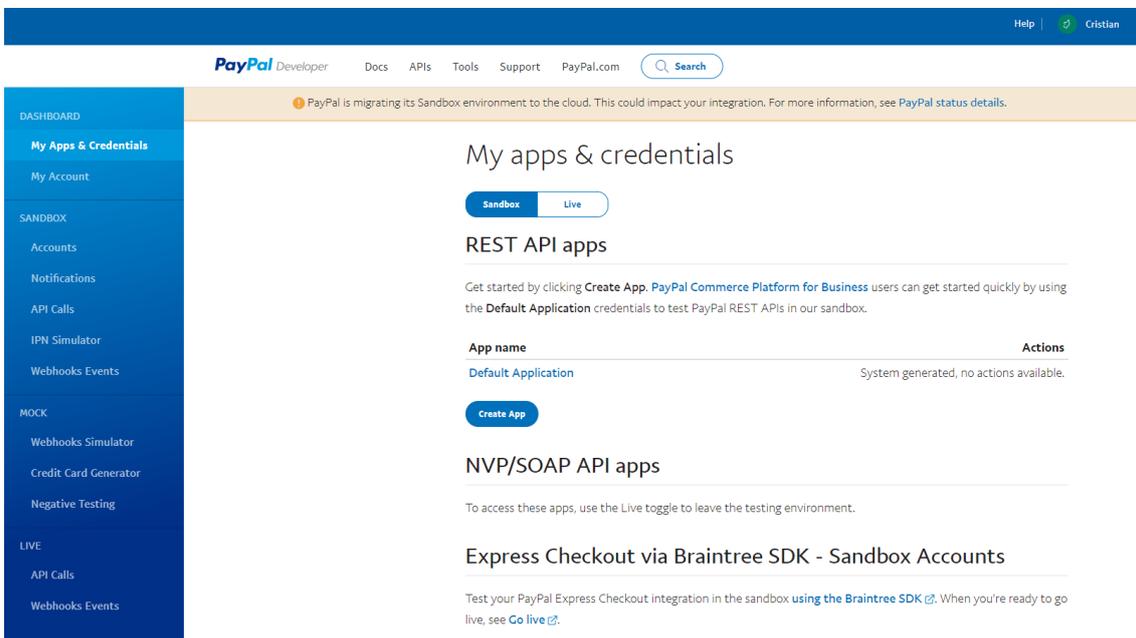


Figura 52 Panel de control de Paypal Developer

Acto seguido nos dirigimos al apartado “**Sandbox**” y haremos *click* en “Accounts” para crear dos cuentas de prueba. Una imitará una cuenta personal de cliente y la otra será la cuenta comercial de la tienda.

## Sandbox Accounts:

Create bulk accounts

Create account

Total Accounts: 4

<input type="checkbox"/> Account name	Type	Country	Date created	Manage accounts
<input type="checkbox"/> sb-a9zfy5956656@personal.example.c...	Personal	ES	17 Apr 2021	...
<input type="checkbox"/> sb-43apqr5959207@business.example....	Business	ES	17 Apr 2021	...

Figura 53 Cuentas usadas para las pruebas de funcionamiento

Estas cuentas sirven para realizar pruebas de conexión y verificar el correcto funcionamiento de la pasarela de pago.

A la hora de escribir el código necesario para integrar la pasarela de pago. Lo primero será escribir la siguiente referencia:

```
<script src="https://www.paypal.com/sdk/js?client-id=Ac5JBnAC-  
Kn1XAoIM9FuetSh6RfBsYWAY3ZopfIb15Ap-  
2sXNXm9V45RvNfKyKnoB_417UR556IcBUA&currency=EUR"></script>
```

Después debemos ejecutar una sentencia SQL SELECT que nos devuelva la tupla del anuncio que se va a comprar. De modo que podamos recoger los datos que necesitamos para generar la factura y que funcione correctamente la pasarela de pago.

```
<?php  
include("cabecera.php");  
$idAnuncio = $_REQUEST["id"];  
$idUserario = 0;  
if (array_key_exists("idusuario", $_SESSION))  
    $idUserario = $_SESSION["idusuario"];  
$mysql = getConnectionMySQL();  
$sql = "SELECT * FROM anuncio WHERE idanuncio = ".$idAnuncio;  
$res = mysqli_query($mysql, $sql);  
$num = mysqli_num_rows($res);  
if ($num > 0) {  
    if ($fila = mysqli_fetch_array($res)) {  
        echo '<div class="justify-content-center">';  
        echo "<div id='paypal-button-container'></div>";  
        echo "<script>paypal.Buttons({  
            createOrder: function(data, actions) {  
                return actions.order.create({  
                    purchase_units: [{  
                        amount: {  
                            value: '>".$fila["precio"]."</div>";  
            }  
        }  
    }  
    },  
    onApprove: function(data, actions) {  
        // This function captures the funds from the transaction.  
        return actions.order.capture().then(function(details) {  
            // This function shows a transaction success message to your buyer.  
            window.location = 'pagado.php?idanuncio=".$idAnuncio."</div>";  
        }  
    }  
    }  
    }).render('#paypal-button-container');</script>";  
    echo '</div>';  
}  
}
```

Figura 54 Integración de PayPal como pasarela de pago

Dentro de la estructura que propone la Api de PayPal podemos indicar en la variable **“purchase\_units”** el número de unidades que se compra de ese producto. En nuestro caso siempre será 1 por lo que podemos dejarlo vacío. En la variable **“value”** indicamos el precio que se ha de pagar por el producto. En **“onApprove”** se capturan los fondos para realizar la transacción y posteriormente enseñará el mensaje de éxito o no de la transacción. Al finalizar el pago redirigimos al usuario a la página **“pagado.php”** junto al identificador del anuncio comprado.

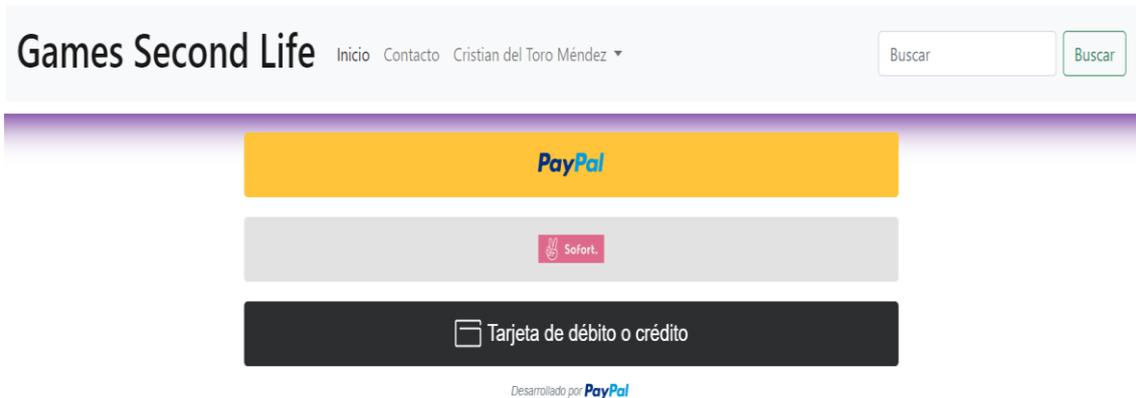


Figura 55 Vista del botón de PayPal en la web

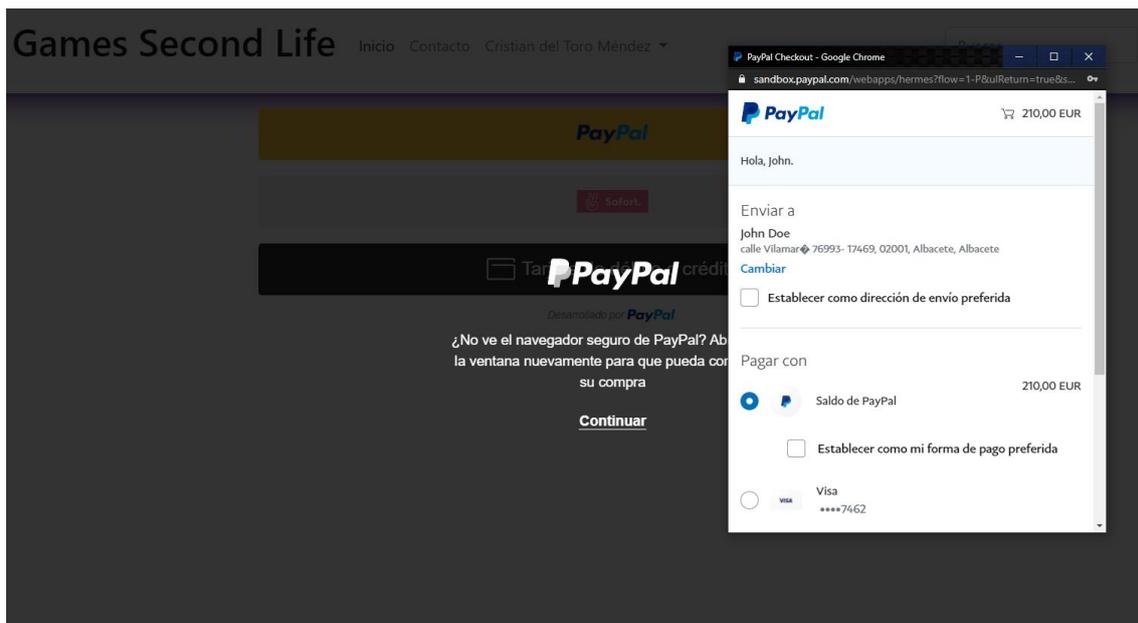


Figura 56 Ejemplo de uso de la pasarela de pago de PayPal

En la página de **“pagado.php”** lo que haremos será marcar el anuncio como comprado, para que no aparezca como disponible. Para ello modificaremos el valor del campo

“**idusariocomprador**” de la tabla “**anuncio**” con el id del usuario que ha comprado el videojuego.

En esta página también generamos la factura apoyándonos en la librería TCPDF. Necesitaremos ejecutar una consulta SQL para obtener los datos del cliente y del vendedor.

Nada más ingresar en la página se enviará la factura generada por correo electrónico tanto al cliente como al vendedor. Para poder enviar correos electrónicos necesitaremos configurar el archivo “**php.ini**” y el “**sendmail.ini**” del servidor de XAMPP. Usaremos la cuenta de Gmail que hemos creado para el proyecto llamada [tiendatfg2@gmail.com](mailto:tiendatfg2@gmail.com) para enviar los correos con la dirección de la factura a descargar.

```
[mail function]
; For Win32 only.
; http://php.net/smtp
SMTP=smtp.gmail.com
; http://php.net/smtp-port
smtp_port=587

; For Win32 only.
; http://php.net/sendmail-from
sendmail_from =tiendatfg2@gmail.com

; For Unix only. You may supply arguments as well (default: "sendmail -t -i").
; http://php.net/sendmail-path
sendmail_path ="\"C:\xampp\sendmail\sendmail.exe\" -t"
```

Figura 57 Configuración php.ini

```
auth_username=tiendatfg2@gmail.com
auth_password=
```

Figura 58 Configuración sendmail.ini

Antes de describir el funcionamiento de la página “pagado.php”, recalcar que dentro de la configuración de la cuenta de Gmail hay que permitir el acceso a aplicaciones poco seguras. Ya que, sino el comando **mail** se ejecutará, pero google bloqueará el acceso, por lo que no se enviará ningún correo.

```

require_once('TCPDF/tcpdf.php');

function generarFactura($fila) {
    // create new PDF document
    $pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT, true, 'UTF-8', false);
    // remove default header/footer
    $pdf->setPrintHeader(false);
    $pdf->setPrintFooter(false);
    // set default monospaced font
    $pdf->SetDefaultMonospacedFont(PDF_FONT_MONOSPACED);
    // set margins
    $pdf->SetMargins(PDF_MARGIN_LEFT, PDF_MARGIN_TOP, PDF_MARGIN_RIGHT);
    // set auto page breaks
    $pdf->SetAutoPageBreak(TRUE, PDF_MARGIN_BOTTOM);
    // set image scale factor
    $pdf->setImageScale(PDF_IMAGE_SCALE_RATIO);
    // set font
    $pdf->SetFont('times', '', 11);
    // add a page
    $pdf->AddPage();
    // set some text to print
    $html = "<h1 style=\"text-align: center;\">Factura N&ordm;".$fila["idanuncio"]."</h1>";
    $html .= "<table>";
    $html .= "<tr><td>Concepto</td><td>Descripción</td><td>Cantidad</td><td>Importe</td></tr>";
    $html .= "<tr><td>".$fila["titulo"]."</td><td>".$fila["descripcion"]."</td><td>1</td><td>".$fila["precio"]." &euro;</td></tr>";
    $html .= "</table>";
    $html .= "<br/><br/><br/>";
    $html .= "Cliente:<br/>";
    $html .= $fila["nombrec"]." ".$fila["apellidoc"]."<br/>";
    $html .= $fila["direccionc"]."<br/>";
    $html .= "DNI: ".$fila["dnic"]."<br/>";
    $html .= "Teléfono: ".$fila["telefonoc"]."<br/>";
    $html .= "Correo: ".$fila["correoc"]."<br/>";
    $pdf->writeHTML($html, true, false, true, false, '');
    // TODO: Escribir el cuerpo de la factura
    //Close and output PDF document
    $pdf->Output(__DIR__.'/facturas/factura_'.$fila["idanuncio"].'.pdf', 'F');
}

```

Figura 59 Uso de la librería TCPDF

En la figura anterior se puede apreciar la función que hemos definido como **generarFactura(\$fila)**. En esta función crearemos el objeto TCPDF al que iremos pasándole los valores correspondientes al formato que queremos. Entre ellos **“SetMargin”** establece donde quieres poner los márgenes, **“SetFont”** indica el tipo de fuente y tamaño, etc...

Posteriormente para imprimir el pdf haremos uso del método writeHTML() de la librería. La ventaja de este método es que podemos darle formato escribiendo en lenguaje HTML y guardándolo en una variable y posteriormente el método lo traduce a formato pdf.

Con Output indicamos donde deseamos guardar la copia de la factura, en nuestro caso lo guardaremos en la ruta **“facturas/factura\_idanuncioXX.pdf”**

```

$idAnuncio = $_REQUEST["idanuncio"];
$idUsuario = 0;
if (array_key_exists("idusuario", $_SESSION))
    $idUsuario = $_SESSION["idusuario"];
$mysql = getConnectionMySQL();
$sql = "SELECT a.idanuncio, a.titulo, a.descripcion, a.precio, uc.nombre
AS nombrec, uc.apellido AS apellidoc, uc.telefono AS telefonoc, uc.correo AS
correoc, uc.direccion AS direccionc, uc.dni AS dnic, uv.nombre
AS nombrev, uv.apellido AS apellidov, uv.telefono AS telefonov, uv.correo AS correov, uv.direccion
AS direccionv, uv.dni AS dniv FROM anuncio a JOIN usuario uc JOIN usuario uv WHERE a.idanuncio = ".$idAnuncio."
AND uc.idusuario = ".$idUsuario." AND (a.idusuario = uv.idusuario)";
$res = mysqli_query($mysql, $sql);
$num = mysqli_num_rows($res);
if ($num > 0) {
    if ($fila = mysqli_fetch_array($res)) {
        generarFactura($fila);
        marcarComprado($idAnuncio, $idUsuario);
        // TODO: Enviar email a comprador y vendedor
        echo "<div class=\"container col-sm-12 col-md-12 mx-2 py-5 text-center\">";
        echo "<h2>Se ha enviado la factura a su correo electrónico</h2>";
        echo "<a class=\"btn btn-primary\" target=\"_blank\" href=\"facturas/factura_".$idAnuncio.".pdf\">Descargar factura</a>";
        echo "</div>";

        mail($_SESSION["correo"], "Factura para comprador", "http://localhost/tiendaTfg/facturas/factura_".$idAnuncio.".pdf");
        mail($fila["correov"], "Factura para vendedor", "http://localhost/tiendaTfg/facturas/factura_".$idAnuncio.".pdf");
    }
}

```

Figura 60 Lógica del caso de uso comprar para la web

La consulta SQL es tan larga ya que necesitamos los datos de dos usuarios distintos pero que comparten los mismos campos. Debemos renombrar cada campo para diferenciar entre el comprador y el vendedor.

Posteriormente, si la consulta se realiza con éxito generamos la factura pasándole como parámetro el array asociativo resultante de la consulta SQL. Marcamos el videojuego como “comprado”, para ello nos hemos creado una función que lo único que hace es un UPDATE en la base de datos, añadiendo el identificador del usuario que compró el producto en el campo “**idusuariocomprador**”

Por último, la función de PHP **mail** enviará el correo electrónico basándose en los archivos de configuración descritos anteriormente.

Se ha enviado la factura a su correo electrónico

[Descargar factura](#)

Figura 61 Vista del resultado al comprar un videojuego



Figura 62 Ejemplo de recepción de factura

**Factura N°12**

Concepto	Descripción	Cantidad	Importe
Masters of combat	Juego muy cotizado de Master System, muy difícil de conseguir, no rebajo.	1	210 €

Cliente:  
 Cristian del Toro Méndez  
 C/ León y Castillo 284 2A, 35005, Las Palmas  
 DNI: 78512322M  
 Teléfono: 679718098  
 Correo: cristian.del.toro.mendez@gmail.com

Figura 63 Factura resultante

### 5.5.10 AÑADIR A FAVORITOS

Esta función está integrada en la página de “**anuncio.php**” la cual hemos descrito que se accede al ver los detalles de un anuncio. En ella se mostrará un corazón en la parte

inferior del producto, el cual, si está negro significa que el anuncio está marcado como favorito y si es blanco, al contrario.

La página “anuncio.php” mediante la función en **AJAX** que hemos creado llamada **corazon(operacion)**, consulta continuamente el estado del corazón.

```
<script>
function corazon(operacion) {
    $.ajax({
        url: 'favorito.php',
        data: {operacion: operacion, idanuncio: <?= $id ?>},
        type: 'GET',
        dataType: 'json'
    }).done(function(json) {
        //alert('corazón: ok');
        var obj = jQuery.parseJSON(json);
        if (obj)
            $('#corazon').html('&#9829;');
        else
            $('#corazon').html('&#9825;');
    }).fail(function(xhr, status, errorThrown) {
        //alert('corazón: error');
        console.log(xhr);
        console.log(status);
        console.log(errorThrown);
    });
}

$(document).ready(function() {
    corazon('consultar');
});
</script>
```

Figura 64 Función AJAX para guardar favoritos

Dicha consulta se lleva a cabo en la página “**favorito.php**”.

```

function getEsFavorito($idAnuncio, $idUserario) {
    $ret = false;
    $sql = "SELECT * FROM anunciofavoritousuario WHERE idanuncio = ".$idAnuncio." AND idusuario = ".$idUserario;
    $mysql = getConexionMySql();
    mysqli_query($mysql, $sql);
    if (mysqli_affected_rows($mysql) > 0)
        $ret = true;
    return $ret;
}

$operacion = $_REQUEST["operacion"];
$idAnuncio = $_REQUEST["idanuncio"];
$idUsuario = $_SESSION["idusuario"];
$esFavorito = getEsFavorito($idAnuncio, $idUserario);
if ($operacion == "modificar") {
    if ($esFavorito)
        $sql = "DELETE FROM anunciofavoritousuario WHERE idanuncio = ".$idAnuncio." AND idusuario = ".$idUserario;
    else
        $sql = "INSERT INTO anunciofavoritousuario VALUES (".$idAnuncio.", ".$idUserario.")";
    $mysql = getConexionMySql();
    mysqli_query($mysql, $sql);
    $esFavorito = getEsFavorito($idAnuncio, $idUserario);
}
echo json_encode($esFavorito);

```

Figura 65 Página favorito.php con la lógica para la función en AJAX corazon()

Dentro de la página usamos la función **getEsFavorito(\$idAnuncio, \$idUserario)** para identificar si un anuncio ya es favorito o no. Si se ha realizado la consulta en la función **corazon(operacion)** con el valor predeterminado por parámetro “consulta” devolveremos el objeto JSON sin entrar en el IF. Gracias a esto conseguimos que se mantenga el corazón negro si es favorito o blanco si no lo es.

En cambio, cuando el usuario hace *click* en el corazón siempre indicará un cambio de estado, por lo que, si antes era favorito, ahora hay que eliminarlo y viceversa. Para ello cuando el usuario haga *click* en el corazón le pasaremos a la función **corazon(operacion)** el valor “modificar” como parámetro. Al hacerlo provocaremos que entre en la sentencia IF y realice la consulta SQL DELETE o INSERT de la tabla “**anunciofavoritousuario**”.

Como es lógico, si dotamos de la posibilidad de marcar anuncios como favoritos. También debemos darle la opción de visualizarlos. En el menú de usuario al hacer *click* en favoritos se le redirigirá a la página “**favoritos.php**” donde podrá visualizar todos los anuncios marcados como favoritos.

Nuevo Anuncio

## Favoritos del usuario

# Legend of Dragoon



Figura 66 Página de favoritos del usuario

### 5.5.11 VER LAS VENTAS REALIZADAS Y PEDIDOS REALIZADOS

Ambas páginas “**ventas.php**” y “**pedidos.php**” funcionan de la misma forma cambiando la condición para mostrar el anuncio. En la página de ventas la condición para mostrar una venta es que el valor del campo “idusuariocomprador” del anuncio sea diferente de 0 y el de idusuario sea el suyo.

Games Second Life Home Contact Pedro Romero ▾

---

[Nuevo Anuncio](#)

**Ventas del usuario**

---

**Masters of combat**



210 €

[Share with twitter](#)

[See details](#)

---

Figura 67 Página de ventas del usuario

La diferencia para mostrar los artículos comprados en la página “**pedidos.php**” es que hay que seleccionar los anuncios cuyo campo “**idusuariocomprador**” sea el mismo que el identificador del usuario logueado.

Games Second Life Inicio Contacto Cristian del Toro Méndez ▾

---

[Nuevo Anuncio](#)

**Pedidos del usuario**

---

**Masters of combat**



210 €

[Compartir en Twitter](#)

[Ver detalles](#)

---

Figura 68 Página de pedidos del usuario

### 5.5.12 TRADUCCIÓN

Para la traducción haremos uso de arrays asociativos uno por cada idioma que queramos incluir en la web. En este momento se dispone de inglés “en”, español (es) y alemán (de).

Deberemos escribir por cada frase o palabra que queramos traducir su valor en el otro idioma y almacenarlo en los arrays. Posteriormente con la función `getT("texto")` indicamos que el texto pasado por parámetro aparezca traducido en el idioma que el usuario seleccionó en el registro. Por defecto si no se detecta ninguna coincidencia en el array se dejará el texto como está.

```
<?php
    $T = array();
    $T["es"] = array();
    $T["en"] = array();
    $T["de"] = array();
    $T["de"]["Descripción"] = "Beschreibung";
    $T["en"]["Descripción"] = "Description";
    $T["en"]["Comprar"] = "Buy";
    $T["de"]["Comprar"] = "Kaufen";
    $T["en"]["Editar"] = "Edit";
    $T["de"]["Editar"] = "Bearbeiten";
    $T["en"]["Borrar"] = "Delete";
    $T["de"]["Borrar"] = "Löschen";
    $T["en"]["Inicio"] = "Home";
    $T["de"]["Inicio"] = "Home";
    $T["en"]["Contacto"] = "Contact";
    $T["de"]["Contacto"] = "Kontakt";
    $T["en"]["Administrar anuncios"] = "Manage ads";
    $T["de"]["Administrar anuncios"] = "Anzeige bearbeiten";
```

Figura 69 Array asociativo usado en la traducción de la página

```
function getT($frase) {
    global $T;
    $idioma = "es";
    if (array_key_exists("idioma", $_SESSION))
        $idioma = $_SESSION["idioma"];
    if (array_key_exists($idioma, $T)) {
        if (array_key_exists($frase, $T[$idioma]))
            return $T[$idioma][$frase];
        else
            return $frase;
    }
    else
        return $frase;
}
```

Figura 70 Función que realiza la traducción de la frase

Al estar guardando el valor del idioma seleccionado por el usuario en un \$\_SESSION, si el usuario cambia el idioma editando el perfil, tendrá que cerrar sesión e iniciarla para visualizar los cambios

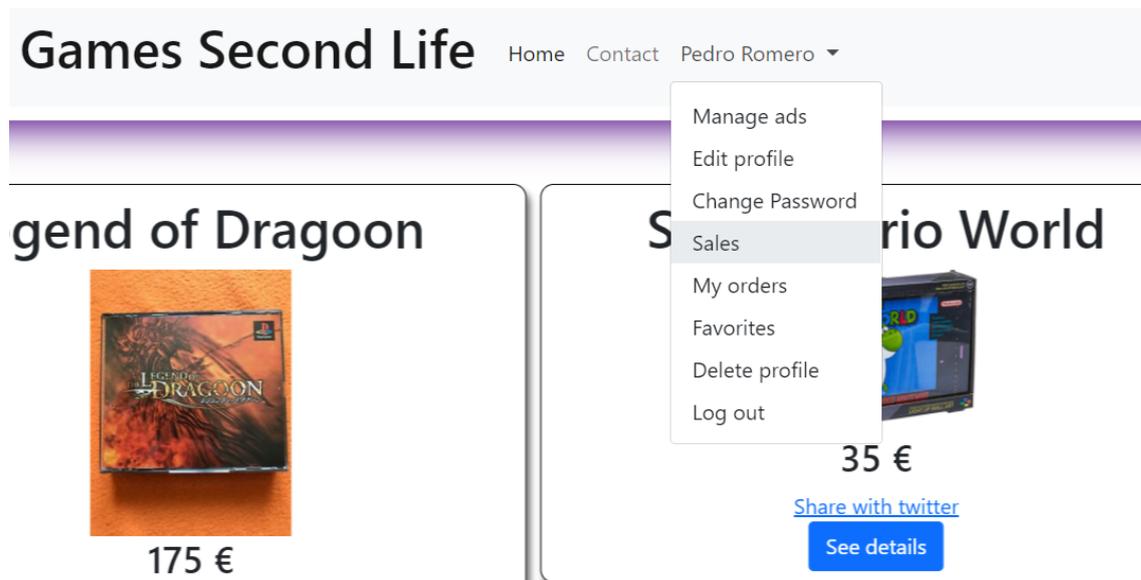


Figura 71 Vista de la página en ingles

### 5.5.13 FUNCIONES AUXILIARES Y CONTROL DE ERRORES

#### **Alertas y ventanas emergentes:**

En este apartado se detallará las funciones que aun sin ser de vital importancia para el funcionamiento de la aplicación, si requieren de cierta explicación.

Como es el caso de las alertas, mensajes de éxito o error y ventanas emergentes usadas durante el proceso de desarrollo. Todas ellas han sido realizadas integrando **Sweetalarm2**.

Su uso es bastante sencillo, la función fire lanzará la ventana emergente con los parámetros que indiquemos. En este caso el título será la frase para mostrar, el icono será el icono de éxito o error y por último el texto del botón de confirmación.

```

function alertaPositiva($frase, $urlDespues = ""){
    ?>
    <script>
        Swal.fire({
            title: '<?= $frase ?>',
            icon: 'success',
            confirmButtonText: '<?= getT("Ok"); ?>'
        }).then((result) => {
            if (result.isConfirmed) {
                if ('<?= $urlDespues ?>' != '')
                    window.location.href = '<?= $urlDespues ?>';
            }
        });
    </script>
    <?php
}

function alertaError($frase){
    ?>
    <script>
        Swal.fire({
            title: '"<?= $frase?>"',
            icon: 'error',
            confirmButtonText: '<?= getT("Cerrar"); ?>'
        })
    </script>
    <?php
}

```

*Figura 72 Funciones usadas para crear las alertas para el control de errores en la web*

Para comprobar que las alertas y control de errores funcionan correctamente, forzaremos los errores para la mayoría de los casos. Estos pueden ser:

- Introducir un correo ya almacenado en el formulario de registro.
- Equivocarse a la hora de introducir la contraseña en cualquier formulario que se solicite.
- Subir una imagen que no sea de tipo JPEG.
- Error inesperado al almacenar la información en la base de datos
- Error de conexión a la base de datos
- Error a la hora de comprar un videojuego

A continuación, se mostrarán algunos de los mensajes de error lanzados por estas pruebas.

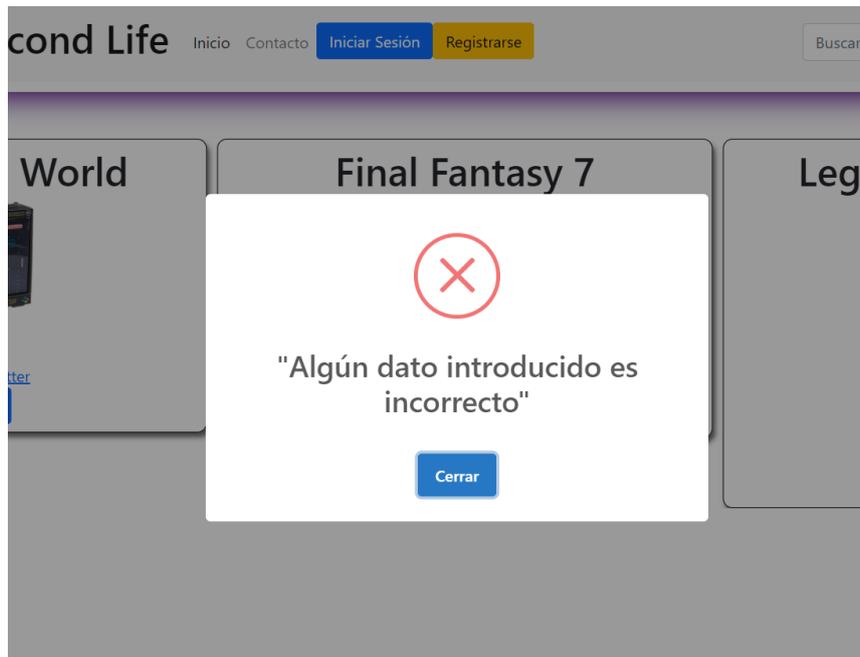


Figura 73 Ejemplo de error al introducir datos erróneos al registrarse en la web

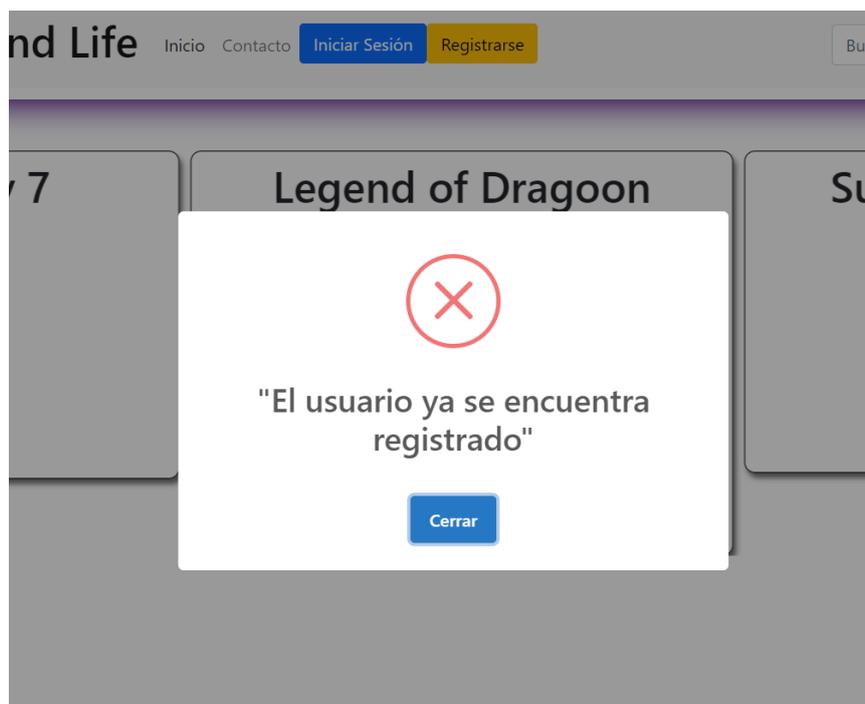


Figura 74 Error al intentar registrarse con un usuario ya almacenado

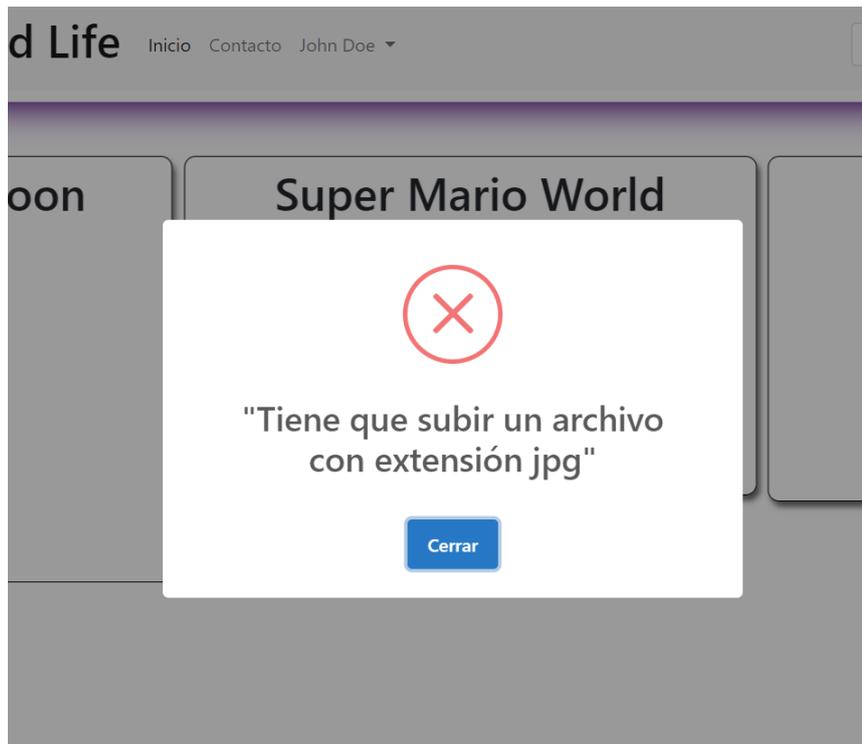


Figura 75 Error al intentar subir una imagen que no sea de tipo JPEG

Como prueba final, se ha testado la aplicación con 8 usuarios con estudios no relacionados con la informática. El *feedback* recabado ha sido muy útil para diseñar la interfaz lo más intuitiva posible. Siendo la versión final la más aceptada como fácil e intuitiva de usar por los usuarios de prueba.

#### **Funciones auxiliares:**

Se ha creado dos funciones con el objetivo de facilitar la inserción de un selector de plataforma y otro de estado en el formulario de publicar anuncio. Su única función es crear dinámicamente el selector de manera que podamos agregar plataformas y estados en la base de datos y no tener la necesidad de añadirlos a cada formulario.

```

function crearCategorias($idPlataformaSeleccionar) {
    $mysql = getConexionMySQL();
    $sql = "SELECT * FROM plataforma";
    $res = mysqli_query($mysql, $sql);
    $respuesta = '<select name="plataforma" class="form-control" require>';
    $respuesta .= '<option value="">Seleccione una plataforma</option>';
    while ($fila = mysqli_fetch_array($res)) {
        $seleccionar = ($fila[0] == $idPlataformaSeleccionar) ? " selected" : "";
        $respuesta .= '<option value="' . $fila[0] . '" . $seleccionar . '>' . $fila[1] . '</option>';
    }
    $respuesta .= '</select>';
    return printf($respuesta);
}

function crearEstados($estadoSeleccionar) {
    $opciones = array("Selecciona una opción", "Precintado", "Excelente", "Muy bueno", "Regular", "Malo", "Muy malo");
    foreach ($opciones as $indice => $valor) {
        $seleccionar = (strcmp($valor, $estadoSeleccionar) == 0) ? " selected" : "";
        echo "<option value=\"\" . $indice . \"\" . $seleccionar . \">\" . $valor . "</option>";
    }
}

```

Figura 76 Funciones para crear selects dinámicos en los formularios web

## 6. ANÁLISIS DE COSTO Y MODELO DE NEGOCIO

### Análisis de costo:

Si tuviéramos que establecer un coste del desarrollo de esta aplicación, debemos basarnos en que no es lo mismo una web desarrollada por un experto que un programador jr. Por lo que nuestro precio por hora debe ser mucho menor.

Al precio por hora trabajada que ronda actualmente para un programador junior suele rondar los 25 euros/hora. Por lo que partimos de las 235 horas empleadas en el diseño y desarrollo de la tienda online. De esas 235 horas se han invertido 50 en comprender y probar las herramientas y sistemas que se desconocían a la hora de hacer el proyecto. Así que a esas 50 horas de estudio se le dará un valor de 10 euros/hora. Las 185 horas restantes invertidas puramente en desarrollo se cobrarán al precio medio de 25 euros/hora.

A su vez toda página web lleva un costo por el Dominio y el Hosting que se paga mensual o anualmente. Los precios variaran en función de los requisitos que necesitemos para el funcionamiento de nuestra web. Siendo una tienda profesional tendríamos que decantarnos por un hosting con un buen espacio de almacenamiento y un dominio .com y .es. Lo que nos sumaría un precio aproximado de unos 84,08 euros al año.

**Alojamiento Profesional**  
 El compañero idóneo para tu tienda online, blog o magazine  
 Características destacadas: Elige la versión PHP, HTTP/2 y DNS Anycast

Olvídate de preocupaciones, el servicio de Alojamiento Profesional es el que buscas para darle el mejor hogar a tu Web y no tener que preocuparte de ella. Mucho más espacio web, tráfico y correo.

<b>PÁGINA WEB</b> 5 GB Espacio web		<b>50 GB</b> Tráfico mensual*		<b>CORREO ELECTRÓNICO</b> 20 Cuentas email		<b>10 GB</b> Espacio por cuenta		<b>60'00</b> €/año	CONTRATAR HOSTING
--	--	----------------------------------	--	--	--	------------------------------------	--	-----------------------	-------------------

\* 0,50 € por cada GB adicional Ver características detalladas →

Figura 77 Página principal de DonDominio y precio de 1 año de alojamiento web

Buscar dominios

tudominio.es Buscar dominios

.es 6,95 €  
 .com 12,95 €  
 .net 13,95 €  
 .store 4,45 €  
 .online 4,45 €  
 .org 9,99 €  
 .eu 4,50 €  
 .site 2,95 €  
 .tech 4,45 €  
 Ver másss →

Figura 78 Ejemplo costes de un dominio en la página de DonDominio

<b>gamesecondlife.com</b>		Eliminar	<b>CONFIGURAR</b>
> Registro de dominio	1 año		12,95 €
> <a href="#">Añadir alojamiento web/correo</a>			
<b>gamesecondlife.es</b>		Eliminar	<b>CONFIGURAR</b>
> Registro de dominio	1 año		6,95 €
> <a href="#">Añadir alojamiento web/correo</a>			
Subtotal			19,90 €
Cuota IVA (21,00 %)			4,18 €
<b>Total</b>			<b>24,08 €</b>

Figura 79 Coste del dominio de Game Second Life durante 1 año

En la siguiente tabla se desglosa el costo final del desarrollo del proyecto:

Fases	Horas	Precio	IVA	Total
<b>Estudio</b>	50	10€	-	500€
<b>Desarrollo</b>	185	25€	-	4.625€
<b>Dominio .es</b>	-	6.95€	1.46€	8.41€
<b>Dominio .com</b>	-	12.95€	2.72€	15.67€
<b>Alojamiento Web</b>	-	60€	12.60€	72.60€
<b>Costo total:</b>				5221.68€

### Modelo de negocio:

Existen varias vías para obtener beneficio de los comercios electrónicos tipo **C2C**:

- Cobrar un suplemento por destacar un anuncio en la página principal.
- Cobrar por destacar tu perfil de vendedor en la página principal.
- Comisión de venta por cada videojuego vendido en la aplicación.
- Establecer algún convenio con empresa de transporte que permita un sistema de envío integrado en la tienda, del que nos llevaríamos una comisión.
- Cobrar por incluir anuncios en la página
- Publicidad en la tienda online

Al ser nuevos en el mercado se optará por modelos de negocio que no limiten demasiado las funcionalidades del usuario. Por ejemplo, sería contraproducente cobrar por anuncio publicado, ya que eso repercutiría en una cantidad mucho menor de videojuegos a la venta para un comercio recién lanzado. Provocando un menor atractivo por el poco catalogo disponible.

Por lo tanto, se ha preferido optar por el modelo de anuncios y perfiles Premium. El usuario puede destacar un anuncio que será visible tanto en la parte superior de la página principal, como en la parte superior de las búsquedas. Se agregaría dentro de un Slider situado encima de las búsquedas del usuario. De esta forma no se interrumpe la vista de la búsqueda del usuario. La cantidad que pagar dependerá del tiempo que quiera el usuario que permanezca su anuncio en la categoría de anuncios destacados.

De la misma forma se destacaría un perfil en una categoría especial, facilitando el acceso a su perfil a otros usuarios y poder visualizar sus productos a la venta.

Ya una vez se haya establecido la tienda y tenga una cantidad respetable de usuarios activos, se podría expandir el modelo de negocio. Podríamos establecer algún convenio con una empresa de transporte para realizar envíos desde la app. Esto nos permitiría poder cobrar una comisión de la venta del videojuego.

Gracias a los cobros se suplen los costes de **mantenimiento** que conlleva tener una tienda online en funcionamiento. Al ser un modelo de negocio donde los propios usuarios venden y compran, es de suma importancia el **feedback** de ellos para mejorar las funciones de la app.

## 7. CONCLUSIONES

En este tema detallaremos las conclusiones alcanzadas durante el desarrollo de este proyecto. Para ello vamos a dividirlo en conclusiones basadas en el resultado, basadas en los objetivos, basadas en futuras mejoras y basadas a nivel personal.

### 7.1 CONCLUSIONES BASADAS EN EL RESULTADO

Aunque se puede decir que el resultado del aspecto de la web peca de simplista, se ha querido profundizar y centrarse en las fases de aprendizaje de herramientas, así como su correcta integración y funcionamiento en el proyecto.

Se ha conseguido integrar multitud de sistemas entre ellos PayPal, Twitter, Google docs, TCPDF, Sweetalarm2, Glide.js, etc...

Por lo que si nos basamos en el funcionamiento de la aplicación creo que se ha conseguido un resultado satisfactorio. Algunas de las sentencias SQL han sido pensadas y replanteadas muchas veces con el fin de optimizar los accesos a la base de datos, intentando que sean los mínimos posibles.

### 7.2 CONCLUSIONES BASADAS EN LOS OBJETIVOS

Según mi postura se ha conseguido alcanzar los objetivos fijados en el desarrollo de la aplicación. Pretendíamos desarrollar una tienda online de segunda mano especializada en videojuegos con integración de sistemas.

Por lo que, en mi opinión, al dotar a los usuarios la posibilidad de subir videojuegos a la venta, editarlos, borrarlos y comprarlos. Se consigue alcanzar los objetivos propuestos.

### 7.3 CONCLUSIONES BASADAS EN FUTURAS MEJORAS

En primera instancia este proyecto ha sido planteado con fines académicos. Esto quiere decir que se ha tenido más en cuenta el aprendizaje de las nuevas herramientas y

lenguajes usados que otros aspectos a tener en cuenta a la hora de realizar una tienda online.

Debido a esto se podría continuar el proyecto para dotarle de más profesionalidad con algunos cambios y añadidos. Como pueden ser:

- Actualmente al subir un videojuego en nuestra aplicación solamente da la posibilidad de añadir una imagen. Podría agregarse la posibilidad de tener una galería de imágenes en los anuncios.
- Se podría implementar un sistema de valoración y comentarios para los usuarios.
- Se podría implementar un sistema de chat en vivo entre usuarios, con el que poder solicitar más información de los productos de ser necesario.
- El aspecto general de la tienda online, aunque se comporte según el modelo *responsive*, tiene bastante margen de mejora.
- Se podría incluir un sistema de localización por el que detectar los anuncios más próximos a tu posición.
- Se podría establecer un XML de traducción que facilite la introducción de nuevos idiomas a la web.

## 7.4 CONCLUSIONES A NIVEL PERSONAL

A nivel personal este desarrollo ha conllevado un fuerte crecimiento personal. Al tener que estudiar tantos distintos lenguajes de programación, APIs, sistemas y herramientas me ha hecho más independiente y resolutivo.

Al combinar tantos lenguajes en un mismo proyecto me ha resultado complicado detectar los errores a medida que iban surgiendo. Por lo que también he tenido que aprender otras formas de depuración haciendo uso del navegador de Chrome, el cual me ha sido muy útil.

La investigación sobre los posibles sistemas a integrar que encajarán dentro del proyecto me ha dado a conocer funciones y herramientas que seguramente siga usando en mi carrera profesional.

A nivel general puedo decir que esta experiencia me ha servido para conocer la tremenda labor que hay detrás de cada pequeña página. Ha conseguido que aumenten mis conocimientos de los lenguajes vistos en el grado. He aprendido muchas nuevas tecnologías y herramientas que me serán de gran utilidad en mi carrera profesional.

Pero sobre todo, ha conseguido motivarme a seguir estudiando y adaptándome a las nuevas tecnologías que vayan surgiendo para contribuir a mi crecimiento profesional. En conclusión, estoy bastante satisfecho con el trabajo realizado.

## 8. BIBLIOGRAFÍA

En este capítulo se nombrará las fuentes de información consultadas para el desarrollo del proyecto.

- **McLaughlin, B. (2013) PHP & MySQL.: The missing manual. 2nd ed. O'Reilly (The missing manual).** Manual de PHP y MySQL que se ha usado con el fin de comprender mejor el desarrollo de las consultas SQL.
- **Weijun Wang et al. (2008) 'Development of China C2C E-commerce from the Perspective of Goods Delivery', in *Integration & Innovation Orient to E-Society Volume 1*, pp. 179–186.** Se realiza un estudio comparativo entre el modelo de comercio electrónico C2C y el B2B en china.
- **Somalo, I. (2017) *El comercio electrónico: una guía completa para gestionar la venta online*. 1ª ed. Esic (Libros Profesionales de Empresa).** Libro bastante informativo sobre las distintas estrategias de negocio, captación de clientes, etc.
- **Holdener, A. T. (2008) *Ajax: the definitive guide*. O'Reilly.** Manual de Ajax consultado durante la realización del proyecto.

### Webgrafía

- **Stack Overflow:** Seguramente es la fuente de información más consultada para la elaboración del proyecto. Consiste en un foro de preguntas y respuestas para programadores informáticos. Se ha recurrido a él muchas veces buscando soluciones cuando parte del código no funcionaba. De la misma forma también ha servido de mucha ayuda a la hora de integrar Glide.js, Ajax, Javascript, etc...

<https://es.stackoverflow.com>

- **W3schools:** Es una página web con infinidad de tutoriales referentes a la creación de páginas web. Se ha consultado como manual de referencia para el HTML fundamentalmente.

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

- **PayPal Developer:** Es la página oficial para desarrolladores de PayPal. En la APIs oficial, se encuentra la información necesaria para la configuración y el correcto funcionamiento de la pasarela de pago en nuestro proyecto.

<https://developer.paypal.com/docs/api/overview/>

- **Sweetalert2:** Es la página oficial donde se encuentra la APIs y documentación de Sweetalert2. Se ha consultado con el fin de poder integrar su uso en el proyecto creando las alertas para el usuario.

<https://sweetalert2.github.io/>

- **CSSmatic:** Es una página web que permite generar una sombra con la forma y color indicados de manera visual y sencilla. Se ha usado para generar la sombra que rodea la tarjeta de cada videojuego.

<https://www.cssmatic.com/es/box-shadow>

- **Boletín informativo del instituto nacional de estadística («El salto del comercio electrónico», 2020):** Se ha consultado la siguiente entrada del blog informativo del INE con el fin de entender el estado actual del comercio electrónico en España.

[https://www.ine.es/ss/Satellite?L=es\\_ES&c=INECifrasINE\\_C&cid=1259952923622&p=1254735116567&pagename=ProductosYServicios%2FINECifrasINE\\_C%2FPYSDetalleCifrasINE](https://www.ine.es/ss/Satellite?L=es_ES&c=INECifrasINE_C&cid=1259952923622&p=1254735116567&pagename=ProductosYServicios%2FINECifrasINE_C%2FPYSDetalleCifrasINE)

- **Glide:** En la página oficial de Glide obtenemos la información necesaria para su integración dentro del proyecto. En este caso ha sido necesario para implementar el carrusel de la página principal del proyecto.

<https://glidejs.com/>

- **Wordpress:** En el proceso de diseño se estuvo evaluando la posibilidad de elaborar la página con wordpress. Dándonos así la posibilidad de integrar y probar la gran variedad de plugins disponibles para el E-commerce en esta plataforma. Debido a su fuerte crecimiento y popularidad, el hacer uso de su plataforma conlleva una suscripción económica.

<https://wordpress.com/es/ecommerce/>

- **Youtube:** Se ha consultado con el mismo fin que hacíamos con Stack Overflow. Ya que al buscar soluciones para las dudas que van surgiendo con el desarrollo del proyecto. Esas búsquedas llevan muchas veces a algún video-tutorial de mucha ayuda.

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

- **Manual de PHP:** Se ha consultado el manual online disponible en la página oficial de PHP con el fin de comprender la sintaxis de ciertas operaciones.

<https://www.php.net/manual/es/index.php>

- **Manual de MySQL:** Se ha consultado muchas veces en relación a la creación de las consultas SQL que usamos dentro del proyecto. Dispone de toda la información relevante a la sintaxis de las consultas.

<https://www.mysqltutorial.org/>

- **Don dominio:** Página de venta de dominios y alojamiento web que hemos usado para el análisis de costos del proyecto.

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

- **Apuntes y Prácticas del Grado de Ingeniería Informática:** Se han consultado los apuntes y prácticas que se ha realizado durante la carrera. En especial de las asignaturas de PR4, SSR, REDES, ICEBT e IRS.

## ANEXO I MANUAL DE USUARIO

The screenshot shows the top navigation bar of the 'Games Second Life' website. It includes the site name, links for 'Inicio', 'Contacto', 'Iniciar Sesión', and 'Registrarse'. There are also search bars with the text 'Buscar'. Below the navigation bar, three game listings are displayed in a row. Each listing features a game cover image, the game title, the price in Euros, a 'Compartir en Twitter' link, and a 'Ver detalles' button. The games listed are 'Masters of combat' (210 €), 'Super Mario World' (35 €), and 'Dragon Quest II' (15 €).

Al acceder a Games Second Life debes crear una cuenta de usuario para empezar a ejercer las funciones de vendedor. Para ello nos dirigiremos al botón de **Registrarse**, haremos *click* en el para dirigirnos al formulario de registro web.

This is a close-up screenshot of the navigation bar from the previous image. The 'Registrarse' button is highlighted with a red circle, indicating the next step in the user manual.

Rellenamos los datos requeridos por el formulario asegurándonos de que sean correctos.

**Correo:**

**Nombre:**

**Apellido:**

**Teléfono:**

**Dirección:**

**Contraseña:**

**Dni:**

**Idioma:**

Una vez realizado el registro y visualizado el mensaje de éxito de la operación, podemos efectuar el inicio de sesión.

Para ello nos dirigimos al botón de **Iniciar Sesión** que nos dirigirá al formulario de inicio de sesión. Procederemos a rellenar el campo correo y contraseña con los datos introducidos al registrarse. Si hemos metido correctamente los datos, nos saldrá una ventana indicando que hemos iniciado la sesión.

es Second Life [Inicio](#) [Contacto](#) [Iniciar Sesión](#) [Registrarse](#)

**Correo:**

**Clave:**

Ahora deberíamos de ver en la barra de navegación el nombre con el que nos hemos registrado. Si hacemos *click* en él, se nos desplegarán las opciones disponibles al ser usuario registrado.

The screenshot shows the top navigation bar of the 'Games Second Life' website. The site name 'Games Second Life' is on the left, followed by links for 'Inicio' and 'Contacto'. On the right, the user's name 'Cristian del Toro Méndez' is displayed with a dropdown arrow. A white menu is open, listing the following options: 'Administrar anuncios', 'Editar perfil', 'Cambiar contraseña', 'Mis ventas', 'Mis compras', 'Favoritos', 'Borrar perfil', and 'Cerrar sesión'. Below the navigation bar, two product listings are visible. The first is 'Legend of Dragoon' with a game box image. The second is 'Super Mario World' with a game box image and a price tag of '35 €'.

En el apartado de administrar anuncios se podrán ver los anuncios en venta actualmente por el usuario. Si hacemos *click* en ver detalles de alguno de ellos, veremos que nos dará la opción de editarlo o borrarlo.

This block shows a detailed view of a 'Super Mario World' game listing. At the top, the title 'Super Mario World' is displayed in a large, bold font. Below the title is an image of the game's box set. Underneath the image, the price '35 €' is shown. Below the price, there is a blue link that says 'Compartir en Twitter'. At the bottom of the listing, there is a blue button with the text 'Ver detalles', which is circled in red.

# Descripción:

En buen estado le falta manual

35 €

Editar

Borrar



Encima de la lista de productos a la venta podemos observar el botón **añadir anuncio**, si hacemos *click* en él nos llevará al formulario para subir un videojuego a la venta. Si los datos son correctos al hacer *click* en **publicar**, el anuncio quedará publicado y disponible para la venta.

Second Life

Inicio Contacto Cristian del Toro Méndez ▾

Buscar

Nuevo Anuncio

Anuncios del usuario

**Título**

**Plataforma**

**Estado del producto**

▼

**Contiene caja original**

**Contiene el manual**

**Incluye los extras de la edición**

**Imagen**

Seleccionar archivoNingún archivo seleccionado

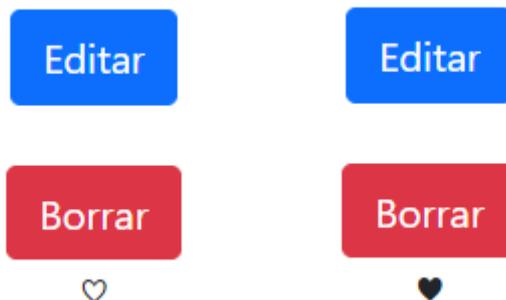
**Descripción**

**Precio** €

Publicar

El apartado mis ventas, mis pedidos y mis favoritos es donde se puede visualizar tanto los videojuegos que se ha vendido, los que se ha comprado a otro usuario y los marcados como favoritos.

Para añadir un anuncio a favoritos simplemente tenemos que ir a los detalles del anuncio y hacer *click* en el corazón debajo de los detalles. De la misma forma haríamos click de nuevo en el corazón para eliminarlo de favoritos.



Si desea comprar un anuncio no tiene nada más que ir a los detalles del anuncio elegido. Dentro de los detalles verá el botón de comprar que le redirigirá a la plataforma de pago de PayPal.



**SQUARESOFT®**

COMPUER

# PlayStation™

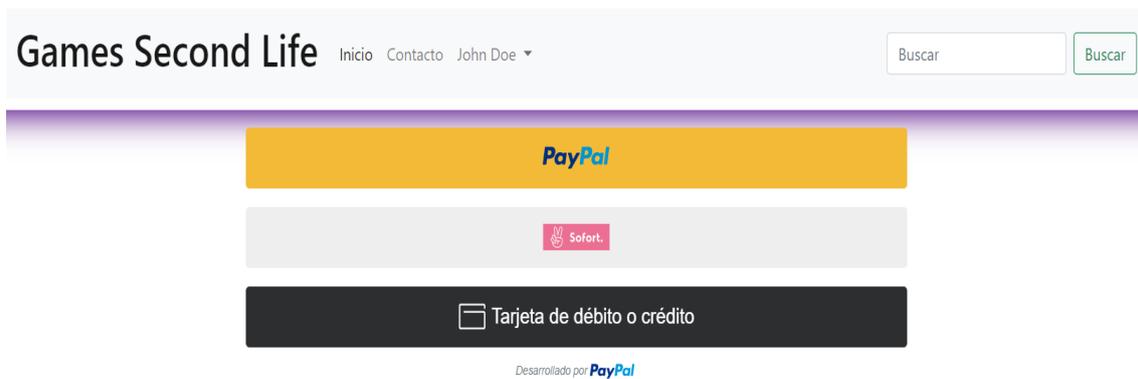
711719024708

Contiene la caja original  
Contiene el Manual  
Contiene todos los extras

**Descripción:**

original  
**55 €**

**Comprar**



**Games Second Life** Inicio Contacto John Doe ▾

Buscar Buscar

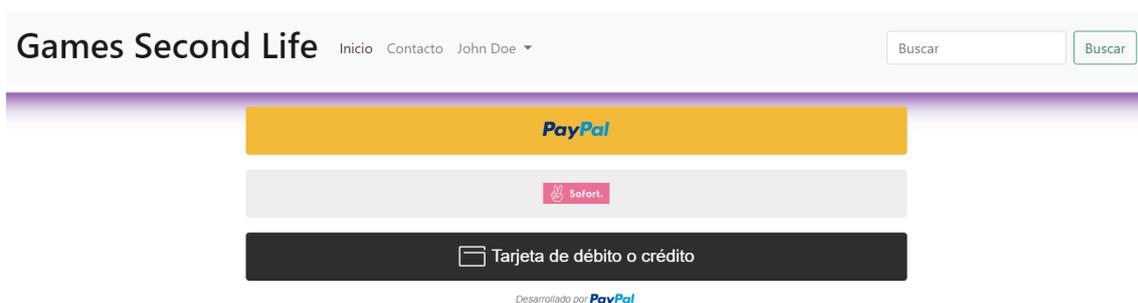
**PayPal**

Sofort.

Tarjeta de débito o crédito

Desarrollado por **PayPal**

Una vez pagado el anuncio se le presentará un enlace para descargar la factura en pdf. Por otro lado, la factura también se le enviará al correo electrónico con el que se ha registrado.



**Games Second Life** Inicio Contacto John Doe ▾

Buscar Buscar

**PayPal**

Sofort.

Tarjeta de débito o crédito

Desarrollado por **PayPal**

Para cualquier duda relacionada con el funcionamiento de la aplicación web o las cuentas de usuario. Puede ponerse en contacto con nosotros desde el apartado de "Contacto". Dentro encontrará el formulario de contacto para el centro de soporte de Gaming Second Life

## Centro de soporte Tienda TFG

¿En qué podemos ayudarte?

**\*Obligatorio**

Nombre \*

Tu respuesta

Correo electrónico \*

Tu respuesta

Comentarios \*

Tu respuesta

## ANEXO II CÓDIGO

### estilo.css

```
*{
  margin: 0;
  padding: 0;
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
header{
  box-shadow: 0 5px 30px indigo;
  padding: 10px;
  background-color: #f8f9fa!important;
}

.btns-index {
  text-align: right;
  padding-top: 20px;
  padding-right: 10px;
}

.admin{
  margin-top: 4px;
}

.navegacion {
  background-color: rgb(234, 234, 241);
}

.formu {
  border: 1px solid black;
  border-radius: 10px;
  -webkit-box-shadow: 23px 21px 45px 6px rgba(113,110,110,1);
  -moz-box-shadow: 23px 21px 45px 6px rgba(113,110,110,1);
  box-shadow: 23px 21px 45px 6px rgba(113,110,110,1);
}

.formu label {
  font-weight:bold;
}

.abs-center {
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
}
```

```

.videojuego {
  word-wrap: break-word;
  border: 1px solid black;
  border-radius:10px;
  -webkit-box-shadow: 3px 5px 5px 0px rgba(113,110,110,1);
  -moz-box-shadow: 3px 5px 5px 0px rgba(113,110,110,1);
  box-shadow: 3px 5px 5px 0px rgba(113,110,110,1);
}

.anuncio {
  word-wrap: break-word;
  border: 1px solid black;
  border-radius:10px;
  -webkit-box-shadow: 3px 5px 5px 0px rgba(113,110,110,1);
  -moz-box-shadow: 3px 5px 5px 0px rgba(113,110,110,1);
  box-shadow: 3px 5px 5px 0px rgba(113,110,110,1);
}

```

## cabecera\_head.php

```

<meta charset="UTF-8">
<title>Games Second Life</title>
<meta name="descripcion" content="Sitio web para la busqueda de j
uegos de segunda mano">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<link rel="stylesheet" href="css/estilo.css"/>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
eOJMYsd53ii+sc0/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+ph31jmwWH6WP0WintQrMb4s7Z0dauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>
<script src="//cdn.jsdelivr.net/npm/sweetalert2@10"></script>
<script src="js/jquery-3.5.1.min.js"></script>

```

## cabecera.php

```
<?php
include_once("traduccion.php");
include_once("conexion.php");

class PonerCompartir {

    const SI = true;
    const NO = false;

}
;

function alertaPositiva($frase, $urlDespues = ""){
    ?>
    <script>
        Swal.fire({
            title: '<?= $frase ?>',
            icon: 'success',
            confirmButtonText: '<?= getT("Ok"); ?>'
        }).then((result) => {
            if (result.isConfirmed) {
                if ('<?= $urlDespues ?>' != '')
                    window.location.href = '<?= $urlDespues ?>';
            }
        });
    </script>
    <?php
}

function alertaError($frase){
    ?>
    <script>
        Swal.fire({
            title: '"<?= $frase?>"',
            icon: 'error',
            confirmButtonText: '<?= getT("Cerrar"); ?>'
        })
    </script>
    <?php
}
```

```

function mostrarAnuncio($fila, $ponerCompartir = PonerCompartir::SI) {
    ?>

    <div class="col-sm-12 col-md-12 videojuego mx-2 py-2 text-
center">
        <h1><?= $fila['titulo'] . " " ?></h1>
        ">
        <h3><?= $fila['precio'] . '&nbsp;€' ?></h3>
        <?php
        if (array_key_exists("esfavorito", $fila)) {
            if ($fila["esfavorito"] == 1)
                echo "<h3>favorito</h3>";
            else
                echo "<h3>no favorito</h3>";
        }
        if ($ponerCompartir == PonerCompartir::SI) {
            echo "<a class=\"twitter-share-
button\" href=\"https://twitter.com/intent/tweet?original_referer=http%3A
%2F%2Flocalhost%2F&ref_src=twsrc%5Etfw&tw_p=tweetbutton&url=http%3A%2F%2F
localhost/tiendaTfg/anuncio.php?id=" . $fila['idanuncio'] . "\">.getT("C
ompartir en Twitter")."</a>";
            echo "<br>";
        }
        ?>
        <button type="button" class="btn btn-
primary" onclick="window.open('anuncio.php?id=<?= $fila['idanuncio'] ?>',
'_blank');"><?= getT("Ver detalles") ?></button>
    </div>

    <?php
}

function publicarAnuncio($titulo, $plataforma, $imagen, $descripcion, $id
, $estado, $caja, $manual, $extras, $precio) {
    $mysql = getConexionMySQL();

    if ($imagen["type"] != "image/jpeg") {
        $respuesta = "<span class = 'msnError'> Tiene que subir un archiv
o jpg</span>";
    } else {
        $rutaOrigen = $imagen["tmp_name"];
        $rutaDestino = "img/" . md5($imagen["tmp_name"]) . ".jpeg";
        $resultado = move_uploaded_file($rutaOrigen, $rutaDestino);
        if ($resultado) {
            $sql = "INSERT INTO anuncio VALUES (null, '$titulo', '$descri
pcion',"
                . " '$rutaDestino', NOW(), '$id', '$plataforma', '$es
tado', '$caja', '$manual', '$extras', '$precio',0)";

```

```

        $res = mysqli_query($mysql, $sql);

        if ($res) {
            alertaPositiva("Anuncio publicado correctamente");
        } else {
            alertaError("Hubo un error al subir el anuncio");
        }
    } else {
        alertaError("Hubo un error al subir el anuncio");
    }
}
}

function crearCategorias($idPlataformaSeleccionar) {
    $mysql = getConexionMySQL();
    $sql = "SELECT * FROM plataforma";
    $res = mysqli_query($mysql, $sql);
    $respuesta = '<select name="plataforma" class="form-
control" require>';
    $respuesta .= '<option value="">Seleccione una plataforma</option>';
    while ($fila = mysqli_fetch_array($res)) {
        $seleccionar = ($fila[0] == $idPlataformaSeleccionar) ? " selected" : "";
        $respuesta .= '<option value="' . $fila[0] . '" . $seleccionar .
'>' . $fila[1] . '</option>';
    }
    $respuesta .= '</select>';
    return printf($respuesta);
}

function crearEstados($estadoSeleccionar) {
    $opciones = array("Selecciona una opción", "Precintado", "Excelente",
"Muy bueno", "Regular", "Malo", "Muy malo");
    foreach ($opciones as $indice => $valor) {
        $seleccionar = (strcmp($valor, $estadoSeleccionar) == 0) ? " sele
cted" : "";
        echo "<option value=\"\" . $indice . "\"\" . $seleccionar . ">" . $
valor . "</option>";
    }
}

session_start();
$mensajeSpan = "";
if (array_key_exists("operacion", $_REQUEST)) {
    if (strcmp($_REQUEST["operacion"], "iniciarSesion") == 0) {
        $correo = $_REQUEST["correo"];
        $clave = $_REQUEST["clave"];
        $mysql = getConexionMySQL();

```

```

    $sql = "SELECT * FROM usuario WHERE correo = '" . $correo . "' AND
D pass = '" . $clave . "'";
    $resExiste = mysqli_query($mysql, $sql);
    $fila = mysqli_fetch_array($resExiste);
    if (mysqli_num_rows($resExiste) > 0) {
        $_SESSION["idusuario"] = $fila[0];
        $_SESSION["nombrecompleto"] = $fila[1] . " " . $fila[2];
        $_SESSION["correo"] = $correo;
        $_SESSION["idioma"] = $fila["idioma"];
        alertaPositiva("Has iniciado sesión correctamente");
    } else {
        alertaError("Algún dato introducido es incorrecto");
    }
} else if (strcmp($_REQUEST["operacion"], "registrar") == 0) {
    $correo = $_REQUEST["correo"];
    $nombre = $_REQUEST["nombre"];
    $apellido = $_REQUEST["apellido"];
    $telefono = $_REQUEST["telefono"];
    $clave = $_REQUEST["clave"];
    $direccion = $_REQUEST["direccion"];
    $dni = $_REQUEST["dni"];
    $idioma = $_REQUEST["idioma"];

    $mysql = getConexionMySql();
    $sqlExiste = "SELECT * FROM usuario WHERE correo = '" . $correo .
    "'";
    $resExiste = mysqli_query($mysql, $sqlExiste);
    if (mysqli_num_rows($resExiste) > 0) {
        $mensajeSpan = "<span class = 'msnError'> El usuario ya está
registrado </span>";
    } else {
        $sql = "INSERT INTO usuario() VALUES (null, '" . $nombre . "'
, '" . $apellido . "', '" . $telefono . "', '" . $correo . "', '" . $clav
e . "', '" . $direccion . "', '" . $dni . "', '" . $idioma . "')";
        $resultado = mysqli_query($mysql, $sql);
        if ($resultado) {
            alertaPositiva("Usuario registrado correctamente");
        } else {
            alertaError("Error al registrar el usuario");
        }
    }
}
} else if (strcmp($_REQUEST["operacion"], "editarPerfil") == 0) {
    //ACTUALIZAR PERFIL
    $correo = $_REQUEST["correo"];
    $nombre = $_REQUEST["nombre"];
    $apellido = $_REQUEST["apellido"];
    $telefono = $_REQUEST["telefono"];
    $clave = $_REQUEST["clave"];

```

```

$direccion = $_REQUEST["direccion"];
$dni = $_REQUEST["dni"];
$id idioma = $_REQUEST["idioma"];
$mysql = getConexionMySQL();
$sql = "UPDATE usuario SET nombre = '". $nombre."', correo = '". $co
rreo."', apellido = '". $apellido."', telefono = '". $telefono."', direccion
= '". $direccion."', dni = '". $dni."', idioma = '". $idioma.'" WHERE (idusu
ario = " $_SESSION["idusuario"]." )";
$resultado = mysqli_query($mysql, $sql);
if ($resultado) {
    alertaPositiva("Usuario actualizado correctamente");
}
else{
    alertaError("Error al actualizar el usuario");
}
} else if (strcmp($_REQUEST["operacion"], "borrarPerfil") == 0) {
    //BORRAR PERFIL
    $clave = $_REQUEST["clave"];
    $mysql = getConexionMySQL();
    $sql = "DELETE FROM usuario WHERE pass = '". $clave.'" AND idusuar
io = " $_SESSION["idusuario"];
    $resultado = mysqli_query($mysql, $sql);
    if ($resultado) {
        alertaPositiva("Usuario borrado correctamente", "salir.php");
    }
    else{
        alertaError("Error al borrar el usuario");
    }
} else if (strcmp($_REQUEST["operacion"], "publicarAnuncio") == 0) {
    publicarAnuncio($_REQUEST["titulo"], $_REQUEST["plataforma"], $_F
ILES["imagen"], $_REQUEST["descripcion"], $_SESSION["idusuario"], $_REQUE
ST["estado"], $_REQUEST["caja"]
        , $_REQUEST["manual"], $_REQUEST["extras"], $_REQUEST["pr
ecio"]);
} else if (strcmp($_REQUEST["operacion"], "actualizarAnuncio") == 0)
{
    $idanuncio = $_REQUEST["id"];
    $titulo = $_REQUEST["titulo"];
    $descripcion = $_REQUEST["descripcion"];
    $precio = $_REQUEST["precio"];
    $idPlataformaSeleccionada = $_REQUEST["plataforma"];
    $estadoSeleccionado = $_REQUEST["estado"];
    $caja = $_REQUEST["caja"];
    $manual = $_REQUEST["manual"];
    $extras = $_REQUEST["extras"];
    $imagen = $_FILES["imagen"];
    $rutaDestino = "rutaImagen";
    if(array_key_exists("type", $imagen)){
        if ($imagen["type"] == "image/jpeg"){

```

```

        $rutaOrigen = $imagen["tmp_name"];
        $rutaDestino = "img/" . md5($imagen["tmp_name"]) . ".jpeg
";
        $resultado = move_uploaded_file($rutaOrigen, $rutaDestino
);
        $rutaDestino = "".$rutaDestino.""
    }
}
$mysql = getConexionMySql();
$sql = "UPDATE anuncio SET titulo = '". $titulo."', descripcion = '
".$descripcion."', precio = '". $precio."', idplataforma = '". $idPlataform
aSeleccionada."', estado = '". $estadoSeleccionado."'
, caja = '". $caja."', manual = '". $manual."', extras = '". $ex
tras."', rutaImagen = ".$rutaDestino." WHERE (idanuncio = ".$idanuncio.")
";
$resultado = mysqli_query($mysql, $sql);
if ($resultado) {
    alertaPositiva("Anuncio actualizado correctamente");
}
else{
    alertaError("Error al actualizar el anuncio");
}
}
else if (strcmp($_REQUEST["operacion"], "cambiarclave") == 0){
    $claveActual = $_REQUEST["claveactual"];
    $claveNueva = $_REQUEST["clavenueva"];
    $claveNueva2 = $_REQUEST["clavenueva2"];
    $mysql = getConexionMySql();
    $sql = "UPDATE usuario SET pass = '". $claveNueva.'" WHERE (idusua
rio = ".$_SESSION["idusuario"].") AND (''. $claveNueva.'" = '". $claveNueva
2.'" ) AND (''. $claveActual.'" = pass)";
    mysqli_query($mysql, $sql);
    if (mysqli_affected_rows($mysql) == 1) {
        alertaPositiva("La contraseña se ha cambiado con exito");
    }
    else{
        alertaError("Error al cambiar la contraseña");
    }
}
}
?>
<header>
    <section class="container">
        <?php
            if (array_key_exists("idusuario", $_SESSION)) {
                ?>
                <nav class="navbar navbar-expand-lg navbar-light bg-light">
                    <div class="container-fluid">

```

```

        <a class="navbar-
brand" href="#"><h1 onclick="window.location = 'index.php';" style="curso
r: pointer;">Games Second Life</h1></a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-
collapse" id="navbarSupportedContent">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item">
                    <a class="nav-link active" aria-
current="page" href="index.php"><?= getT("Inicio"); ?></a>
                </li>
                <li class="nav-item">
                    <a class="nav-
link" href="contacto.php"><?= getT("Contacto"); ?></a>
                </li>
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-
toggle" href="#" id="navbarDropdown" role="button" data-bs-
toggle="dropdown" aria-expanded="false">
                        <?php echo $_SESSION["nombrecompleto"
]; ?>
                    </a>
                    <ul class="dropdown-menu" aria-
labelledby="navbarDropdown">
                        <li><a class="dropdown-
item" href="perfil.php?<?php echo $_SESSION["idusuario"]; ?>"><?= getT("A
dministrar anuncios"); ?></a></li>
                        <li><a class="dropdown-
item" href="registrar.php?idusuario=<?php echo $_SESSION["idusuario"]; ?>
"><?= getT("Editar perfil"); ?></a></li>
                        <li><a class="dropdown-
item" href="cambiarclave.php?idusuario=<?php echo $_SESSION["idusuario"];
?>"><?= getT("Cambiar contraseña"); ?></a></li>
                        <li><a class="dropdown-
item" href="ventas.php?<?php echo $_SESSION["idusuario"]; ?>"><?= getT("M
is ventas"); ?></a></li>
                        <li><a class="dropdown-
item" href="pedidos.php?<?php echo $_SESSION["idusuario"]; ?>"><?= getT("
Mis compras"); ?></a></li>
                        <li><a class="dropdown-
item" href="favoritos.php?<?php echo $_SESSION["idusuario"]; ?>"><?= getT
("Favoritos"); ?></a></li>

```

```

                <li><a class="dropdown-
item" href="borrarperfil.php?<?php echo $_SESSION["idusuario"]; ?>"><? = g
etT("Borrar perfil"); ?></a></li>
                <li><a class="dropdown-
item" href="salir.php"><? = getT("Cerrar sesión"); ?></a></li>
            </ul>
        </li>
    </ul>

    <form action="buscar.php" method="post" class="d-
flex">
        <input class="form-control me-
2" type="search" name="busqueda" id="busqueda" placeholder="Buscar" aria-
label="Search">
        <button class="btn btn-outline-
success" name="buscar" type="submit">Buscar</button>
    </form>
</div>
</nav>
<?php
} else {
    ?>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container-fluid">
            <a class="navbar-
brand" href="#"><h1 onclick="window.location = 'index.php';" style="curso
r: pointer;">Games Second Life</h1></a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-
collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active" aria-
current="page" href="index.php">Inicio</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-
link" href="contacto.php"><? = getT("Contacto"); ?></a>
                    </li>
                    <li class="nav-item">
                        <button type="button" class="btn btn-
primary" onclick="window.location = 'iniciar_sesion.php';">Iniciar Sesión
</button>
                    </li>

```

```

        <li class="nav-item">
            <button type="button" class="btn btn-
warning" onclick="window.location = 'registrar.php';">Registrar</button
>
        </li>
    </ul>

    <form action="buscar.php" method="post" class="d-
flex">
        <input class="form-control me-
2" type="search" name="busqueda" id="busqueda" placeholder="<?= getT("Bus
car"); ?>" aria-label="Search">
        <button class="btn btn-outline-
success" name="buscar" type="submit"><?= getT("Buscar"); ?></button>
    </form>
</div>
</nav>

<?php
}
?>
</section>
</header>

```

## index.php

```

<!DOCTYPE html>
<html>
    <head>
        <?php include("cabecera_head.php"); ?>
        <link rel="stylesheet" href="css/glide.core.min.css"></link>
        <script src="js/glide.min.js"></script>
    </head>
    <body>
        <?php
            include("cabecera.php");

            $mysql = getConexionMySQL();
            $sql = "SELECT * FROM anuncio WHERE (idusuariocomprador = 0)
ORDER BY fecha DESC LIMIT 6";
            $res = mysqli_query($mysql, $sql);
        ?>
        <div class="glide mt-5">
            <div class="glide__track" data-glide-el="track">
                <ul class="glide__slides">
                    <?php

```

```

        while ($fila = mysqli_fetch_array($res)) {
            echo "<li class=\"glide__slide\">";
            mostrarAnuncio($fila, PonerCompartir::SI);
            echo "</li>";
        }
    ?>

</ul>
</div>
<div class="glide__bullets" data-glide-el="controls[nav]">
    <button class="glide__bullet" data-glide-
dir="<">&lt;</button>
    <button class="glide__bullet" data-glide-
dir=">">&gt;</button>
</div>
</div>
<script>
    new Glide('.glide', {
        type: 'carousel',
        startAt: 0,
        perView: 3,
        autoplay: 2000
    }).mount();
</script>
</body>
</html>

```

## anuncio.php

```

<!DOCTYPE html>
<?php include("traduccion.php"); ?>
<html>
    <head>
        <?php
            include("cabecera_head.php");

            $id = $_REQUEST["id"];
        ?>
        <script>
            function corazon(operacion) {
                $.ajax({
                    url: 'favorito.php',
                    data: {operacion: operacion, idanuncio: <?= $id ?>},
                    type: 'GET',
                    dataType: 'json'
                }).done(function(json) {
                    //alert('corazón: ok');
                });
            }
        </script>
    </head>
    <body>
        <div id="anuncio">
            <div class="glide">
                <div class="glide__slides">
                    <div class="glide__slide">
                        <div class="glide__img">
                            <img alt="Anuncio 1" />
                        </div>
                        <div class="glide__caption">
                            <p>Anuncio 1</p>
                        </div>
                    </div>
                    <div class="glide__slide">
                        <div class="glide__img">
                            <img alt="Anuncio 2" />
                        </div>
                        <div class="glide__caption">
                            <p>Anuncio 2</p>
                        </div>
                    </div>
                </div>
                <div class="glide__bullets">
                    <div class="glide__bullet">&lt;</div>
                    <div class="glide__bullet">&gt;</div>
                </div>
            </div>
        </div>
    </body>
</html>

```

```

        var obj = jQuery.parseJSON(json);
        if (obj)
            $('#corazon').html('&#9829;');
        else
            $('#corazon').html('&#9825;');
    }).fail(function(xhr, status, errorThrown) {
        //alert('corazón: error');
        console.log(xhr);
        console.log(status);
        console.log(errorThrown);
    });
}

$(document).ready(function() {
    corazon('consultar');
});

function clickBorrar() {
    Swal.fire({
        title: '¿Borrar?',
        showDenyButton: true,
        showCancelButton: false,
        confirmButtonText: 'Sí',
        denyButtonText: 'No'
    }).then((result) => {
        if (result.isConfirmed) {
            window.location.href = 'borraranuncio.php?idanuncio=
io=<?= $id ?>';
        }
    });
}

</script>
</head>
<body>
    <?php
        include("cabecera.php");

        $idUserio = 0;
        if (array_key_exists("idusuario", $_SESSION))
            $idUserio = $_SESSION["idusuario"];
        $mysql = getConexionMySQL();
        $sql = "SELECT * FROM anuncio WHERE idanuncio = ".$id;
        $res = mysqli_query($mysql, $sql);
        if ($fila = mysqli_fetch_array($res)) {
            echo '<div class="row mt-5 justify-content-center">';
            echo '<div class="col-3 anuncio text-center">';
            echo "<h1>".$fila['titulo']."</h1>";
            echo "<p>".$fila['estado']."</p>";
        }
    }
}

```

```

        echo "<img src=\"\".$fila['rutaImagen'].\"\" width=\"100%\"
alt=\"\".$fila['titulo'].\"\">";
        echo "<br>";
        if ($fila['caja'] == 1)
            echo getT("Contiene la caja original");
        else
            echo getT("Sin la caja original");
        echo "<br>";
        if ($fila['manual'] == 1)
            echo getT("Contiene el Manual");
        else
            echo getT("Sin manual");
        echo "<br>";
        if ($fila['extras'] == 1)
            echo getT("Contiene todos los extras");
        else
            echo getT("Sin extras");
        echo "<br>";
        echo "<h3>".getT("Descripción")."&nbsp;  </h3>".$fila['des
cripcion'];
        echo "<h3>".$fila['precio']."&nbsp;  €</h3>";
        if ($fila["idusuariocomprador"] == 0){
            if ($idUserario == $fila["idusuario"]){
                echo "<a class=\"btn btn-
primary\" href=\"publicar.php?idanuncio=".$id."\">".getT("Editar")."</a><
br/>";
                echo "<br>";
                echo "<a class=\"btn btn-
danger\" href=\"javascript:clickBorrar();\">".getT("Borrar")."</a>";
            }
            else
                echo "<a class=\"btn btn-
primary\" href=\"comprar.php?id=".$id."\">".getT("Comprar")."</a>";
        }
        echo "<div id=\"corazon\" onclick=\"corazon('modificar');
\" style=\"cursor: pointer;\"></div>";
        echo "</div>";
        echo "</div>";
    }

    else {
        echo getT("Anuncio no encontrado");
    }
?>
</body>
</html>

```

## publicar.php

```
<!DOCTYPE html>
<html>
  <head>
    <?php include("cabecera_head.php"); ?>
  </head>
  <body>
    <?php
      include("cabecera.php");

      $idAnuncio = 0;
      if (array_key_exists("idanuncio", $_REQUEST))
        $idAnuncio = $_REQUEST["idanuncio"];
      $titulo = "";
      $descripcion = "";
      $precio = "";
      $operacion = "publicarAnuncio";
      $idPlataformaSeleccionada = 0;
      $estadoSeleccionado = "";
      $caja = 0;
      $manual = 0;
      $extras = 0;
      if ($idAnuncio != 0) {
        $mysql = getConexionMySql();
        $sql = "SELECT * FROM anuncio WHERE idanuncio = ".$idAnun
cio;

        $resExiste = mysqli_query($mysql, $sql);
        $fila = mysqli_fetch_array($resExiste);
        if (mysqli_num_rows($resExiste) > 0) {
          $titulo = $fila["titulo"];
          $descripcion = $fila["descripcion"];
          $precio = $fila["precio"];
          $idPlataformaSeleccionada = $fila["idplataforma"];
          $estadoSeleccionado = $fila["estado"];
          $caja = $fila["caja"];
          $manual = $fila["manual"];
          $extras = $fila["extras"];
        }
        $operacion = "actualizarAnuncio";
      }
    ?>
    <div class="row justify-content-center mx-2 mt-5">
      <div class="formu col-sm-12 col-md-12 col-lg-6 col-xl-6 p-
4">
        <form action="index.php" method="post" enctype="multipart
/form-data">
```

```

        <div class="mb-3">
            <label class="form-label">Título</label>
            <input type="hidden" name="id" value="<?= $idAnun
cio; ?>" />
            <input type="text" class="form-
control" name="titulo" placeholder="Título" value="<?= $titulo ?>" />
        </div>
        <div class="mb-3">
            <label class="col-sm-2 control-
label">Plataforma</label>
            <div class="mb-3">
                <?php
                    crearCategorias($idPlataformaSeleccionada
);
                ?>
            </div>
        </div>
        <div class="mb-3">
            <label class="col-sm-2 control-
label">Estado del producto</label>
            <div class="mb-3">
                <select class="form-select" aria-
label="Seleccionar estado" name="estado">
                    <?php
                        crearEstados($estadoSeleccionado);
                    ?>
                </select>
            </div>
        </div>
        <div class="form-check">
            <input type="hidden" name="caja" value="0" />
            <input class="form-check-
input" type="checkbox" value="1" id="flexCheckDefault" name="caja" <?= ($
caja == 1) ? " checked" : ""; ?>/>
            <label class="form-check-
label" for="flexCheckDefault">
                Contiene caja original
            </label>
        </div>
        <div class="form-check">
            <input type="hidden" name="manual" value="0" >
            <input class="form-check-
input" type="checkbox" value="1" id="flexCheckDefault" name="manual" <?=
($manual == 1) ? " checked" : ""; ?>/>
            <label class="form-check-
label" for="flexCheckDefault">
                Contiene el manual
            </label>
        </div>

```

```

        <div class="form-check">
            <input type="hidden" name="extras" value="0" />
            <input class="form-check-
input" type="checkbox" value="1" id="flexCheckDefault" name="extras" <?=
($extras == 1) ? " checked" : ""; ?>/>
            <label class="form-check-
label" for="flexCheckDefault">
                Incluye los extras de la edición
            </label>
        </div>
        <div class="mb-3">
            <label class="form-label">Imagen</label>
            <input type="file" class="form-
control" name="imagen">
        </div>
        <div class="mb-3">
            <label class="form-label">Descripción</label>
            <textarea name="descripcion" id="form-
control" class="form-control" rows="5"><?= $descripcion; ?></textarea>
        </div>
        <div class="input-group mb-3">
            <label class="form-label">Precio</label>
            <span class="input-group-text">€</span>
            <input type="number" class="form-control" aria-
label="Monto" step="any" name="precio" value="<?= $precio; ?>"/>
        </div>
        <input type="hidden" name="operacion" value="<?= $ope
racion; ?>"/>
        <input class="btn btn-
primary" type="submit" name="submit" value="Aceptar"/>
    </form>
</div>
</div>
</body>
</html>

```

## borraranuncio.php

```

<!DOCTYPE html>
<html>
    <head>
        <?php include("cabecera_head.php"); ?>
    </head>
    <body>
        <?php
            include("cabecera.php");

```

```

        $idAnuncio = 0;
        if (array_key_exists("idanuncio", $_REQUEST))
            $idAnuncio = $_REQUEST["idanuncio"];
        if ($idAnuncio != 0) {
            $mysql = getConexionMySQL();
            $sql = "DELETE FROM `anuncio` WHERE idanuncio = ".$idAnun
cio;

            $resultado = mysqli_query($mysql, $sql);
            if ($resultado) {
                alertaPositiva("Anuncio borrado correctamente");
            }
            else{
                alertaError("Error al borrar el anuncio");
            }
        }
    }
    ?>
</body>
</html>

```

## borrarperfil.php

```

<!DOCTYPE html>
<?php include("traduccion.php"); ?>
<html>
    <head>
        <?php
            include("cabecera_head.php");
        ?>
        <script>
            function clickBorrar() {
                Swal.fire({
                    title: '¿Borrar perfil?',
                    showDenyButton: true,
                    showCancelButton: false,
                    confirmButtonText: 'Sí',
                    denyButtonText: 'No'
                }).then((result) => {
                    if (result.isConfirmed) {
                        $('form#formulario').submit();
                    }
                });
            }
        </script>
    </head>

```

```

<body>
  <?php
    include("cabecera.php");
  ?>
  <div class="row justify-content-center mx-2 mt-5">
    <div class="formu col-sm-12 col-md-12 col-lg-6 col-xl-6 p-4">
      <form action="index.php" method="post" id="formulario">
        <div class="form-group mb-3">
          <input type="hidden" name="operacion" value="borrarPerfil"/>
          <label for="formGroupExampleInput">Confirme la clave:</label>
          <input type="password" name="clave" class="form-control" placeholder="Introduzca su clave"/>
          <br>
          <button type="button" class="btn btn-danger" onclick="clickBorrar();">Borrar</button>
        </div>
      </form>
    </div>
  </div>
</body>
</html>

```

## buscar.php

```

<!DOCTYPE html>
<html>
  <head>
    <?php include_once("cabecera_head.php"); ?>
  </head>
  <body>
    <?php
      include_once("cabecera.php");

      $busqueda = $_REQUEST["busqueda"];
      $offset = 0;
      if (array_key_exists("offset", $_REQUEST))
        $offset = $_REQUEST["offset"];
      $idUserio = 0;
      if (array_key_exists("idusuario", $_SESSION))
        $idUserio = $_SESSION["idusuario"];
      $mysql = getConexionMySQL();
      $sql = "

```

```

        SELECT a.*, CASE WHEN afu.idusuario IS NOT NULL THEN 1 ELSE 0 END AS esfavorito
    FROM anuncio a LEFT JOIN anunciofavoritousuario afu ON (a.idanuncio = afu.idanuncio) AND (afu.idusuario = ".$idUsuario.")
    WHERE (a.titulo LIKE '%{$busqueda}%') AND (a.idusuariocomprador = 0) ORDER BY fecha LIMIT ".(NUM_ELEMENTOS_BUSQUEDA + 1)." OFFSET ".$offset;
    $res = mysqli_query($mysql, $sql);
    $num = mysqli_num_rows($res);
    if ($num > 0) {
        $i = 0;
        while (($fila = mysqli_fetch_array($res)) && ($i < NUM_ELEMENTOS_BUSQUEDA)) {
            mostrarAnuncio($fila, PonerCompartir::SI);
            $i++;
        }
        $offsetAnterior = $offset - NUM_ELEMENTOS_BUSQUEDA;
        $offsetSiguiente = $offset + NUM_ELEMENTOS_BUSQUEDA;
        echo "<br/>";
        if ($offset != 0)
            echo "<a class=\"btn btn-primary mx-2\" href=\"?offset=".$offsetAnterior."&busqueda=".$busqueda.\">Anterior</a>";
        if ($num > NUM_ELEMENTOS_BUSQUEDA)
            echo "<a class=\"btn btn-primary mx-2\" href=\"?offset=".$offsetSiguiente."&busqueda=".$busqueda.\">Siguiente</a>";
    }
    else
        echo '<h2>Sin resultados</h2>';
?>
<style>
    @media (max-width: 480px) {
        .videojuego img {
            width: 60% !important;
        }
    }
</style>
</body>
</html>

```

cambiarclave.php

```

<!DOCTYPE html>
<html>
<head>

```

```

    <?php include("cabecera_head.php"); ?>
</head>
<body>
    <?php include("cabecera.php"); ?>

    <div class="row justify-content-center mx-2 mt-5">
        <div class="formu col-sm-12 col-md-12 col-lg-6 col-xl-6 p-4">
            <form action="index.php" method="post">
                <div class="form-group mb-3">
                    <label for="formGroupExampleInput">Clave actual:</label>

                    <input type="password" class="form-control" id="formGroupExampleInput" name="claveactual" placeholder="Introduzca la contraseña actual">
                </div>
                <div class="form-group mb-3">
                    <label for="formGroupExampleInput">Clave nueva:</label>

                    <input type="password" class="form-control" id="formGroupExampleInput" name="clavenueva" placeholder="Introduzca la nueva contraseña">
                </div>
                <div class="form-group mb-3">
                    <label for="formGroupExampleInput">Repita clave nueva:</label>

                    <input type="password" class="form-control" id="formGroupExampleInput" name="clavenueva2" placeholder="Repita la nueva contraseña">
                </div>
                <input type="hidden" name="operacion" value="cambiarclave"/>
                <button type="submit" name="submit" value="Enviar" class="btn btn-warning">Cambiar Clave</button>
            </form>
        </div>
    </div>
</body>
</html>

```

## comprar.php

```

<!DOCTYPE html>
<html>
    <head>
        <?php include("cabecera_head.php"); ?>

```

```

    <script src="https://www.paypal.com/sdk/js?client-id=Ac5JBnAC-
Kn1XAoIM9FuetSh6RfBsYWAY3ZopfIb15Ap-
2sxNXm9V45RvNfKyKnhoB_417UR556IcBUA&currency=EUR"></script>
</head>
<body>
    <?php
        include("cabecera.php");
        $idAnuncio = $_REQUEST["id"];
        $idUserario = 0;
        if (array_key_exists("idusuario", $_SESSION))
            $idUserario = $_SESSION["idusuario"];
        $mysql = getConexionMySQL();
        $sql = "SELECT * FROM anuncio WHERE idanuncio = ".$idAnuncio;
        $res = mysqli_query($mysql, $sql);
        $num = mysqli_num_rows($res);
        if ($num > 0) {
            if ($fila = mysqli_fetch_array($res)) {
                echo '<div class="text-center">';
                echo "<div id=\"paypal-button-container\"></div>";
                echo "<script>paypal.Buttons({
                    createOrder: function(data, actions) {
                        return actions.order.create({
                            purchase_units: [{
                                amount: {
                                    value: '".$fila["precio"]."
                                }
                            }]
                        });
                    },
                    onApprove: function(data, actions) {
                        // This function captures the funds from the
transaction.
                        return actions.order.capture().then(function(
details) {
                            // This function shows a transaction succ
ess message to your buyer.
                            window.location = 'pagado.php?idanuncio="
.$idAnuncio.';
                        });
                    }
                }).render('#paypal-button-container');</script>";
                echo '</div>';
            }
        }
    ?>
</body>
</html>

```

## pagado.php

```
<!DOCTYPE html>
<html>
  <head>
    <?php include("cabecera_head.php"); ?>
  </head>
  <body>
    <?php
      include("cabecera.php");
      require_once('TCPDF/tcpdf.php');

      function generarFactura($fila) {
        // create new PDF document
        $pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE
_FORMAT, true, 'UTF-8', false);
        // remove default header/footer
        $pdf->setPrintHeader(false);
        $pdf->setPrintFooter(false);
        // set default monospaced font
        $pdf->SetDefaultMonospacedFont(PDF_FONT_MONOSPACED);
        // set margins
        $pdf-
>SetMargins(PDF_MARGIN_LEFT, PDF_MARGIN_TOP, PDF_MARGIN_RIGHT);
        // set auto page breaks
        $pdf->SetAutoPageBreak(TRUE, PDF_MARGIN_BOTTOM);
        // set image scale factor
        $pdf->setImageScale(PDF_IMAGE_SCALE_RATIO);
        // set font
        $pdf->SetFont('times', '', 11);
        // add a page
        $pdf->AddPage();
        // set some text to print
        $html = "<h1 style=\"text-
align: center;\">Factura N&ordm;".$fila["idanuncio"]."</h1>";
        $html .= "<table>";
        $html .= "<tr><td>Concepto</td><td>Descripción</td><td>Ca
ntidad</td><td>Importe</td></tr>";
        $html .= "<tr><td>".$fila["titulo"]."</td><td>".$fila["de
scripcion"]."</td><td>1</td><td>".$fila["precio"]." &euro;</td></tr>";
        $html .= "</table>";
        $html .= "<br/><br/><br/>";
        $html .= "Cliente:<br/>";
        $html .= $fila["nombrec"]." ".$fila["apellidoc"]."<br/>";
        $html .= $fila["direccionc"]."<br/>";
        $html .= "DNI: ".$fila["dnic"]."<br/>";
        $html .= "Teléfono: ".$fila["telefonoc"]."<br/>";
      }
    }
  }
}
```

```

        $html .= "Correo: ".$fila["correoc"]."<br/>";
        $pdf->writeHTML($html, true, false, true, false, '');
        // TODO: Escribir el cuerpo de la factura
        //Close and output PDF document
        $pdf->
>Output(__DIR__.'/facturas/factura_'.$fila["idanuncio"].'.pdf', 'F');
    }

    function getClaveCorrecta($idUserio, $clave) {
        $mysql = getConexionMySql();
        $sql = "SELECT * FROM usuario WHERE idusuario = ".$idUserio." AND pass = '".$clave."'";
        $res = mysqli_query($mysql, $sql);
        return (mysqli_num_rows($res) > 0) ? true : false;
    }

    function marcarComprado($idAnuncio, $idUserio) {
        $mysql = getConexionMySql();
        $sql = "UPDATE anuncio SET idusuariocomprador = ".$idUserio." WHERE (idanuncio = ".$idAnuncio.)";
        mysqli_query($mysql, $sql);
    }

    $idAnuncio = $_REQUEST["idanuncio"];
    $idUserio = 0;
    if (array_key_exists("idusuario", $_SESSION))
        $idUserio = $_SESSION["idusuario"];
    $mysql = getConexionMySql();
    $sql = "SELECT a.idanuncio, a.titulo, a.descripcion, a.precio, uc.nombre AS nombrec, uc.apellido AS apellidoc, uc.telefono AS telefonoc, uc.correo AS correoc, uc.direccion AS direccionc, uc.dni AS dnic, uv.nombre AS nombrev, uv.apellido AS apellidov, uv.telefono AS telefonov, uv.correo AS correov, uv.direccion AS direccionv, uv.dni AS dniv FROM anuncio a JOIN usuario uc JOIN usuario uv WHERE a.idanuncio = ".$idAnuncio." AND uc.idusuario = ".$idUserio." AND (a.idusuario = uv.idusuario)";
    $res = mysqli_query($mysql, $sql);
    $num = mysqli_num_rows($res);
    if ($num > 0) {
        if ($fila = mysqli_fetch_array($res)) {
            generarFactura($fila);
            marcarComprado($idAnuncio, $idUserio);
            // TODO: Enviar email a comprador y vendedor
            echo "<div class=\"container col-sm-12 col-md-12 mx-2 py-5 text-center\">";
                echo "<h2>Se ha enviado la factura a su correo electrónico</h2>";
                echo "<a class=\"btn btn-primary\" target=\"_blank\" href=\"facturas/factura_".$idAnuncio.".pdf\">Descargar factura</a>";
        }
    }
}

```

```

        echo "</div>";

        mail($_SESSION["correo"], "Fectura para comprador", "
http://localhost/tiendaTfg/facturas/factura_".$idAnuncio.".pdf");
        mail($fila["correov"], "Factura para vendedor", "http
://localhost/tiendaTfg/facturas/factura_".$idAnuncio.".pdf");
    }
}
?>
</body>
</html>

```

## conexion.php

```

<?php
include("config.php");

function getConnectionMySQL() {
    // Crear conexión
    $conn = mysqli_connect(SERVER, USER, PASS, BD);
    // Verificación
    if (!$conn) {
        die("Error al conectar a la BD: " . mysqli_connect_error());
    }
    mysqli_set_charset($conn, "utf8");
    return $conn;
}
?>

```

## config.php

```

<?php
define("SERVER", "localhost");
define("BD", "bd-tienda");
define("USER", "root");
define("PASS", "");
define("NUM_ELEMENTOS_BUSQUEDA", 3);
?>

```

## contacto.php

```
<!DOCTYPE html>
<html>
  <head>
    <?php include("cabecera_head.php"); ?>
  </head>
  <body>
    <?php
      include("cabecera.php");
    ?>
    <div class="abs-center embed-responsive embed-responsive-16by9">
      <iframe class="embed-responsive-
item" src="https://docs.google.com/forms/d/e/1FAIpQLSd_7bf9P13s_qgAQNKZc0
WDUnp70rwY84ryb6p_M7ZpbGKFg/viewform?embedded=true" width="640" height="
717" frameborder="0" marginheight="0" marginwidth="0">Cargando...</iframe>
    </div>
  </body>
</html>
```

## favorito.php

```
<?php
  session_start();
  include("conexion.php");

  function getEsFavorito($idAnuncio, $idUserario) {
    $ret = false;
    $sql = "SELECT * FROM anunciofavoritousuario WHERE idanuncio = "
.$idAnuncio." AND idusuario = ".$idUserario;
    $mysql = getConnectionMySQL();
    mysqli_query($mysql, $sql);
    if (mysqli_affected_rows($mysql) > 0)
      $ret = true;
    return $ret;
  }

  $operacion = $_REQUEST["operacion"];
  $idAnuncio = $_REQUEST["idanuncio"];
  $idUserario = $_SESSION["idusuario"];
  $esFavorito = getEsFavorito($idAnuncio, $idUserario);
  if ($operacion == "modificar") {
    if ($esFavorito)
```

```

        $sql = "DELETE FROM anunciofavoritousuario WHERE idanuncio =
        ".$idAnuncio." AND idusuario = ".$idUsuario;
        else
            $sql = "INSERT INTO anunciofavoritousuario VALUES (". $idAnuncio.", ". $idUsuario.")";
        $mysql = getConexionMySql();
        mysqli_query($mysql, $sql);
        $esFavorito = getEsFavorito($idAnuncio, $idUsuario);
    }
    echo json_encode($esFavorito);
?>

```

## favoritos.php

```

<!DOCTYPE html>
<html>
    <head>
        <?php include("cabecera_head.php"); ?>
    </head>
    <body>
        <?php
            include("cabecera.php");
        ?>
        <section class="admin">
            <div class="container" style="text-align: center">
                <button type="button" class="btn btn-
                primary" onclick="window.open('publicar.php');">Nuevo Anuncio</button>
                <div class="row">
                    <h2>Favoritos del usuario</h2>
                </div>
            </div>
        </section>

        <?php
            $idUsuario = 0;
            if (array_key_exists("idusuario", $_SESSION))
                $idUsuario = $_SESSION["idusuario"];
            $mysql = getConexionMySql();
            //$sql = "SELECT * FROM anunciofavoritousuario WHERE idusuario =
            ".$idUsuario." ORDER BY fecha";
            $sql = "SELECT a.* FROM anuncio a JOIN anunciofavoritousuario afu
            ON (afu.idusuario = ".$idUsuario.") AND (a.idanuncio = afu.idanuncio) OR
            DER BY fecha";
            $res = mysqli_query($mysql, $sql);
            $num = mysqli_num_rows($res);
            if ($num > 0) {

```

```

        while ($fila = mysqli_fetch_array($res)) {
            mostrarAnuncio($fila);
        }
    }
    else
        echo "<h3>No hay Favoritos</h3>"
    ?>

</body>
</html>

```

## iniciar\_sesion.php

```

<!DOCTYPE html>
<html>
    <head>
        <?php include("cabecera_head.php"); ?>
    </head>
    <body>
        <?php include("cabecera.php"); ?>

        <div class="row justify-content-center mx-2 mt-5">
            <div class="formu col-sm-12 col-md-12 col-lg-6 col-xl-6 p-4">
                <form action="index.php" method="post">
                    <div class="form-group mb-3">
                        <label for="formGroupExampleInput">Correo:</label>
                    >
                        <input type="text" class="form-
control" id="formGroupExampleInput" name="correo" placeholder="Introduzca
un Email válido">
                    </div>
                    <div class="form-group mb-3">
                        <label for="formGroupExampleInput2">Clave:</label>
                    >
                        <input type="password" class="form-
control" name="clave" id="formGroupExampleInput2" placeholder="Escriba la
contraseña">
                    </div>
                    <input type="hidden" name="operacion" value="iniciarS
esion"/>

                    <button type="submit" class="btn btn-
primary" >Iniciar Sesión</button>

                </form>
            </div>
        </div>

```

```

    </div>
  </body>
</html>

```

## pedidos.php

```

<!DOCTYPE html>
<html>
  <head>
    <?php include("cabecera_head.php"); ?>
  </head>
  <body>
    <?php
      include("cabecera.php");
    ?>
    <section class="admin">
      <div class="container" style="text-align: center">
        <button type="button" class="btn btn-
primary" onclick="window.open('publicar.php');">Nuevo Anuncio</button>
        <div class="row">
          <h2>Pedidos del usuario</h2>
        </div>
      </div>
    </section>

    <?php
      $idUserario = 0;
      if (array_key_exists("idusuario", $_SESSION))
        $idUserario = $_SESSION["idusuario"];
      $mysql = getConexionMySQL();
      $sql = "SELECT * FROM anuncio WHERE idusuariocomprador = ".$idUser
ario." ORDER BY fecha";
      $res = mysqli_query($mysql, $sql);
      $num = mysqli_num_rows($res);
      if ($num > 0) {
        while ($fila = mysqli_fetch_array($res)) {
          mostrarAnuncio($fila);
        }
      }
      else
        echo "<h3>No hay pedidos realizados</h3>"
    ?>

  </body>
</html>

```

## perfil.php

```
<!DOCTYPE html>
<html>
  <head>
    <?php include("cabecera_head.php"); ?>
  </head>
  <body>
    <?php
      include("cabecera.php");
    ?>
    <section class="admin">
      <div class="container" style="text-align: center">
        <button type="button" class="btn btn-
primary" onclick="window.open('publicar.php');">Nuevo Anuncio</button>
        <div class="row">
          <h2>Anuncios del usuario</h2>
        </div>
      </div>
    </section>

    <?php
      $idUserario = 0;
      if (array_key_exists("idusuario", $_SESSION))
        $idUserario = $_SESSION["idusuario"];
      $mysql = getConexionMySQL();
      $sql = "SELECT * FROM anuncio WHERE idusuario = ".$idUserario." AN
D idusuariocomprador = 0 ORDER BY fecha";
      $res = mysqli_query($mysql, $sql);
      $num = mysqli_num_rows($res);
      $numAnuncio = 3;
      if ($num > 0) {
        echo '<div class = "justify-content-center">';
        echo '<div class = "row mt-4">';
        $i = 0;
        while ($fila = mysqli_fetch_array($res)) {
          if (($i % $numAnuncio) != 0) {
            mostrarAnuncio($fila);
            $i++;
          }
          else {
            echo '</div>';
            echo '<div class = "row mt-4">';
            mostrarAnuncio($fila);
            $i++;
          }
        }
      }
    }
  }

```

```

    }
    }
    echo '</div>';
    echo '</div>';
  }
  else
    echo "<h3>No hay anuncios</h3>"
  ?>
</body>
</html>

```

## registrar.php

```

<!DOCTYPE html>
<html>
  <head>
    <?php include("cabecera_head.php"); ?>
  </head>
  <body>

    <script type="text/javascript">
      function validarEmail(elemento) {

        var texto = document.getElementById(elemento.id).value;
        var regex = /^[-\w.%+]{1,64}@(?:[A-Z0-9-
]
]{{1,63}}\.){{1,125}}[A-Z]{{2,63}}$/i;

        if (!regex.test(texto)) {
          document.getElementById("resultado").innerHTML = "Cor
reo inválido";
        } else {
          document.getElementById("resultado").innerHTML = "";
        }
      }
    </script>

    <?php include("cabecera.php");
    $idUserario = 0;
    if (array_key_exists("idusuario", $_REQUEST))
      $idUserario = $_REQUEST["idusuario"];
    $correo = "";
    $nombre = "";
    $apellido = "";
    $telefono = "";
    $direccion = "";
    $clave = "";

```

```

$dni = "";
$operacion = "registrar";

if ($idUsuario != 0) {
    // hacer consulta SQL para obtener el valor de titulo, de
    // descripción y precio que hay ahora
    $mysql = getConexionMySql();
    $sql = "SELECT * FROM usuario WHERE idusuario = ".$idUsua
    rio;

    $resExiste = mysqli_query($mysql, $sql);
    $fila = mysqli_fetch_array($resExiste);
    if (mysqli_num_rows($resExiste) > 0) {
        $correo = $fila["correo"];
        $nombre = $fila["nombre"];
        $apellido = $fila["apellido"];
        $telefono = $fila["telefono"];
        $direccion = $fila["direccion"];
        $clave = $fila["pass"];
        $dni = $fila["dni"];
    }
    $operacion = "editarPerfil";
}
?>
<div class="row justify-content-center mx-2 mt-5">
    <div class="formu col-sm-12 col-md-12 col-lg-6 col-xl-6 p-
    4">
        <form action="index.php" method="post">
            <div class="form-group mb-3">
                <label for="formGroupExampleInput">Correo:</label
                >
                <input type="text" id="email" id="formGroupExamp
                leInput" name="correo" class="form-
                control" placeholder="Correo Electronico" value="<?= $correo ?>" onkeyup=
                "validarEmail(this)">
                <a id='resultado' style="color:#FF0000;"></a>
            </div>
            <div class="form-group mb-3">
                <label for="formGroupExampleInput">Nombre:</label
                >
                <input type="text" name="nombre" class="form-
                control" placeholder="Nombre" id="formGroupExampleInput" value="<?= $nomb
                re ?>" />
            </div>
            <div class="form-group mb-3">
                <label for="formGroupExampleInput">Apellido:</lab
                el>
                <input type="text" name="apellido" class="form-
                control" placeholder="Apellido completo" id="formGroupExampleInput" value
                ="<?= $apellido ?>" />
            </div>
        </form>
    </div>
</div>

```

```

        </div>
        <div class="form-group mb-3">
            <label for="formGroupExampleInput">Teléfono:</label>
            <input type="text" name="telefono" class="form-control" placeholder="Número de teléfono" id="formGroupExampleInput" value="<?= $telefono ?>"/>
        </div>
        <div class="form-group mb-3">
            <label for="formGroupExampleInput">Dirección:</label>
            <input type="text" name="direccion" class="form-control" placeholder="Dirección completa" id="formGroupExampleInput" value="<?= $direccion ?>"/>
        </div>
        <div class="form-group mb-3">
            <label for="formGroupExampleInput">Contraseña:</label>
            <input type="password" name="clave" class="form-control" placeholder="Escriba la contraseña" id="formGroupExampleInput" value="<?= $clave ?>"/>
        </div>
        <div class="form-group mb-3">
            <label for="formGroupExampleInput">Dni:</label>
            <input type="text" name="dni" class="form-control" placeholder="Introduzca su Dni" id="formGroupExampleInput" value="<?= $dni ?>"/>
        </div>
        <div class="form-group mb-3">
            <label for="formGroupExampleInput">Idioma:</label>
            <select name="idioma" class="form-control" placeholder="Idioma" id="formGroupExampleInput">
                <option value="es">Español</option>
                <option value="en">English</option>
                <option value="de">Deutsch</option>
            </select>
        </div>
        <input type="hidden" name="operacion" value="<?= $operacion; ?>"/>
        <button type="submit" class="btn btn-warning" name="submit">Aceptar</button>
    </form>
</div>
</body>
</html>

```

## salir.php

```
<?php
    session_start();
    session_destroy();
    header("Location: index.php");
?>
```

## traduccion.php

```
<?php
    $T = array();
    $T["es"] = array();
    $T["en"] = array();
    $T["de"] = array();
    $T["de"]["Descripción"] = "Beschreibung";
    $T["en"]["Descripción"] = "Description";
    $T["en"]["Comprar"] = "Buy";
    $T["de"]["Comprar"] = "Kaufen";
    $T["en"]["Editar"] = "Edit";
    $T["de"]["Editar"] = "Bearbeiten";
    $T["en"]["Borrar"] = "Delete";
    $T["de"]["Borrar"] = "Löschen";
    $T["en"]["Inicio"] = "Home";
    $T["de"]["Inicio"] = "Home";
    $T["en"]["Contacto"] = "Contact";
    $T["de"]["Contacto"] = "Kontakt";
    $T["en"]["Administrar anuncios"] = "Manage ads";
    $T["de"]["Administrar anuncios"] = "Anzeige bearbeiten";
    $T["en"]["Editar perfil"] = "Edit profile";
    $T["de"]["Editar perfil"] = "Profil bearbeiten";
    $T["en"]["Cambiar contraseña"] = "Change Password";
    $T["de"]["Cambiar contraseña"] = "Passwort ändern";
    $T["en"]["Mis ventas"] = "Sales";
    $T["de"]["Mis ventas"] = "Meine Verkäufe";
    $T["en"]["Mis compras"] = "My orders";
    $T["de"]["Mis compras"] = "Meine Käufe";
    $T["en"]["Favoritos"] = "Favorites";
    $T["de"]["Favoritos"] = "Favoriten";
    $T["en"]["Borrar perfil"] = "Delete profile";
    $T["de"]["Borrar perfil"] = "Profil löschen";
    $T["en"]["Buscar"] = "Search";
    $T["de"]["Buscar"] = "Suchen";
```

```

$T["en"]["Anuncios del usuario"] = "User Ads";
$T["de"]["Anuncios del usuario"] = "Meine Anzeigen";
$T["en"]["Nuevo Anuncio"] = "New Ad";
$T["de"]["Nuevo Anuncio"] = "Anzeige erstellen";
$T["en"]["Ver detalles"] = "See details";
$T["de"]["Ver detalles"] = "Details ";
$T["en"]["Compartir en Twitter"] = "Share with twitter";
$T["de"]["Compartir en Twitter"] = "Auf Twitter teilen";
$T["en"]["Cerrar sesión"] = "Logout";
$T["de"]["Cerrar sesión"] = "Logout";
$T["en"]["Contiene la caja original"] = "Original box included";
$T["de"]["Contiene la caja original"] = "Originalverpackt";
$T["en"]["Sin la caja original"] = "Original box not included";
$T["de"]["Sin la caja original"] = "Ohne Originalverpackung";
$T["en"]["Contiene el Manual"] = "Instructions included";
$T["de"]["Contiene el Manual"] = "Enthält Spielanleitung";
$T["en"]["Sin el Manual"] = "Instructions not included";
$T["de"]["Sin el Manual"] = "Entählt keine Spielanleitung";
$T["en"]["Contiene todos los extras"] = "Extras included";
$T["de"]["Contiene todos los extras"] = "Enthält alle Extras";
$T["en"]["Sin extras"] = "Extras not included";
$T["de"]["Sin extras"] = "Enthält keine Extras";
$T["en"]["Precintado"] = "Precinted";
$T["de"]["Precintado"] = "Versiegelt";
$T["en"]["Excelente"] = "Excelent";
$T["de"]["Excelente"] = "Ausgezeichnet";
$T["en"]["Muy bueno"] = "Very Good";
$T["de"]["Muy bueno"] = "Sehr Gut";
$T["en"]["Regular"] = "Regular";
$T["de"]["Regular"] = "Gut";
$T["en"]["Malo"] = "Bad";
$T["de"]["Malo"] = "Schlecht";
$T["en"]["Muy malo"] = "Very bad";
$T["de"]["Muy malo"] = "Sehr schlechts";

function getT($frase) {
    global $T;
    $idioma = "es";
    if (array_key_exists("idioma", $_SESSION))
        $idioma = $_SESSION["idioma"];
    if (array_key_exists($idioma, $T)) {
        if (array_key_exists($frase, $T[$idioma]))
            return $T[$idioma][$frase];
        else
            return $frase;
    }
    else
        return $frase;
}

```

```
}  
?>
```

## ventas.php

```
<!DOCTYPE html>  
<html>  
  <head>  
    <?php include("cabecera_head.php"); ?>  
  </head>  
  <body>  
    <?php  
      include("cabecera.php");  
    ?>  
    <section class="admin">  
      <div class="container" style="text-align: center">  
        <button type="button" class="btn btn-  
primary" onclick="window.open('publicar.php');">Nuevo Anuncio</button>  
        <div class="row">  
          <h2>Ventas del usuario</h2>  
        </div>  
      </div>  
    </section>  
  
    <?php  
      $idUserario = 0;  
      if (array_key_exists("idusuario", $_SESSION))  
        $idUserario = $_SESSION["idusuario"];  
      $mysql = getConexionMySQL();  
      $sql = "SELECT * FROM anuncio WHERE idusuario = ".$idUserario." AN  
D idusuariocomprador != 0 ORDER BY fecha";  
      $res = mysqli_query($mysql, $sql);  
      $num = mysqli_num_rows($res);  
      if ($num > 0) {  
        while ($fila = mysqli_fetch_array($res)) {  
          mostrarAnuncio($fila);  
        }  
      }  
      else  
        echo "<h3>No hay ventas realizadas</h3>"  
    ?>  
  </body>  
</html>
```