

Memoria Grupo 5



Álvaro Abad de Donesteve

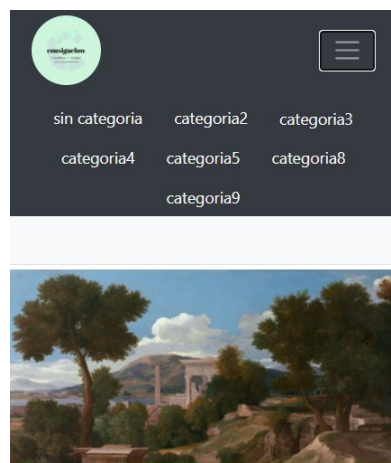
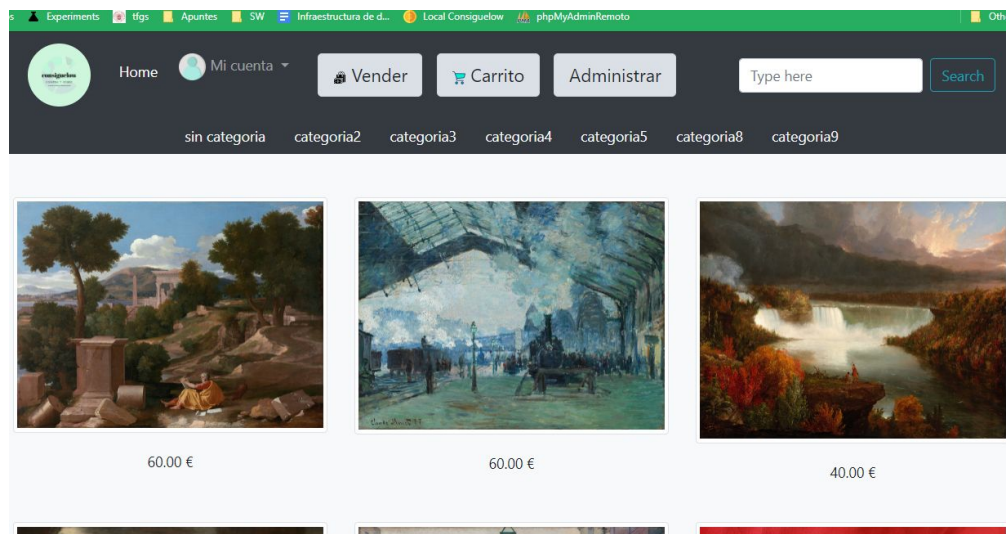
Nestor Antonio Marín Gómez

Introducción:

Consiguelow es el portal de compra venta de todo tipo de productos de particular a particular. Nuestra interfaz directa e intuitiva permite subir productos al portal para que sean expuestos a potenciales compradores, quienes podrán realizar el pedido directamente o añadir el producto al carrito para su posterior adquisición.

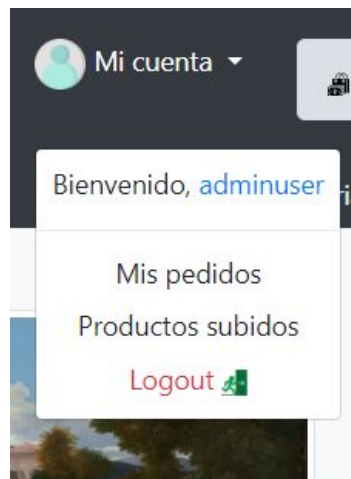
Descripción de la aplicación

La página principal de la aplicación presenta los productos subidos con su imagen y precio. La imagen, análogamente a ebay, es un enlace a la vista detallada del producto. A un usuario con la sesión iniciada no se le presentan sus propios productos en esta página. Los productos pedidos y pagados tampoco se muestran, a no ser que se cancele el pedido. Desde esta vista se puede pinchar en las fotos de los productos para dirigirse a su vista detallada o interactuar con el navbar.



En el navbar tenemos las siguientes opciones:

- Mi cuenta: desplegable con opciones para iniciar sesión o registrarse. En caso de ya haber iniciado sesión, las opciones pasan a ser estos enlaces:



- *nombre de usuario*: enlace al perfil del propio usuario. Muestra la información relativa al usuario e incluye enlaces a los pedidos y a los productos del mismo.
- *Mis pedidos* ([vistaPedidos.php](#)): enlace a los productos realizados por el usuario. Permite cancelarlos.
- *Productos subidos*([vistaProdsUsuario.php](#)): desde esta vista las imágenes enlazan a la vista detallada del producto, pero las opciones han cambiado respecto al acceso mediante enlaces de la portada u otros. *Comprar* y *Añadir al carrito* son sustituidos por la opción *Eliminar*.
- *Logout*

Sin loguearse:

- *Login*: formulario convencional, basado en el proyecto de 'estructura proyecto'. En el placeholder se incluyen por comodidad datos válidos del usuario con permisos de administrador que se pueden insertar mediante el sql. Incluye enlaces a la portada y al formulario de registro: ([¿Todavía no tienes cuenta? Regístrate aquí](#)).
- *Registro*: formulario de registro del usuario. Se ha incluido una validación en el cliente (también la hay en el servidor) del nombre de usuario. Pulsaremos Check para comprobar la disponibilidad:

Registro de usuario

Nombre de usuario:



Nombre completo:

Si se incluyen símbolos inválidos se muestra un mensaje:

Registro de usuario

Nombre de usuario:

El nombre de usuario sólo puede contener símbolos alfanuméricos

Mediante html también exige un formato para la tarjeta de crédito:

tarjeta credito:



Please match the requested format.

Se realizan distintas validaciones de expresiones regulares en el servidor para varios datos. En caso de error se controla y muestra al estilo del 'estructura-proyecto':

Registro de usuario

- Formato del dni incorrecto. Ejemplo 12345678Z
- El campo telefono no es válido

Si todo sale bien redirecciona a la portada con la sesión iniciada, hecho que podemos comprobar en el navbar anteriormente mencionado.

- *Vender (navbar)*: enlace al formulario de subida de productos.

Subir un producto

No has iniciado sesión, inicia sesión

[Login](#)

El precio puede contener decimales separados por un punto. La talla se ha representado en el código en una clase y cuenta con su propia lógica. Si no se especifica se guardará como '--'. La categoría tampoco es necesario especificarla, ya que se guardará como 'Sin categoría'. El desplegable de categorías muestra todas las cargadas de la base de datos. El resto de campos son obligatorios. Esta vista es responsive.

Subir un producto

Nombre del producto:

Descripcion

Precio del producto:

Unidades:

Talla -- ▾

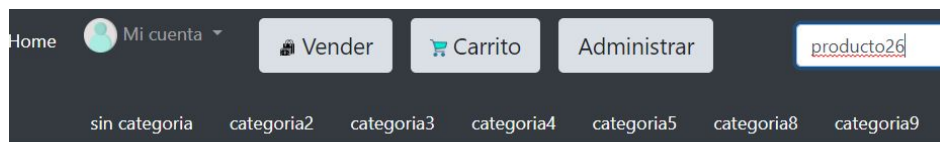
Categoría elige categoría ▾

Imagen

Choose File No file chosen

- Carrito: muestra todos los pedidos del usuario sin pagar. Si otro usuario compra uno de estos productos, desaparecerá del carro del usuario que lo contuviera. A la derecha incluye una factura del total de los productos del carrito. Desde esta vista podemos acceder a la caja para comprar el producto o eliminarlo del carrito, en cuyo caso mediante javascript se borra el html asociado al mismo en el listado y en la factura, que se recalcula. Las tarjetas de los productos incluyen enlaces al producto y al vendedor. Por último el botón de *Caja* del Listado de items, estaba pensado para la función de comprar varios productos, pero no ha llegado a implementarse.
- Administrar: este botón aparece en el navbar solo si el usuario cuenta con el rol de administrador insertado en la base de datos. Enlaza a una vista con accesos a funciones de gestión.

- [Añadir una categoría](#): formulario para insertar nuevas categorías. Al no tener productos no se mostrarán inicialmente en el navbar, para no navegar por categorías vacías, pero sí se cargarán en el formulario de subida de productos de vender.php.
- [Ver categorías ya existentes](#): permite eliminar categorías.
- [Administrar usuarios de la web](#): permite eliminar usuarios.
- [Gestionar los productos de la web](#): permite eliminar productos.
-
- El último elemento del navbar es el searchbar, de funcionalidad limitada, pues que consiste únicamente en una sencilla consulta sql.



Resultados del filtrado...

Resultados que coinciden con tu búsqueda: producto26

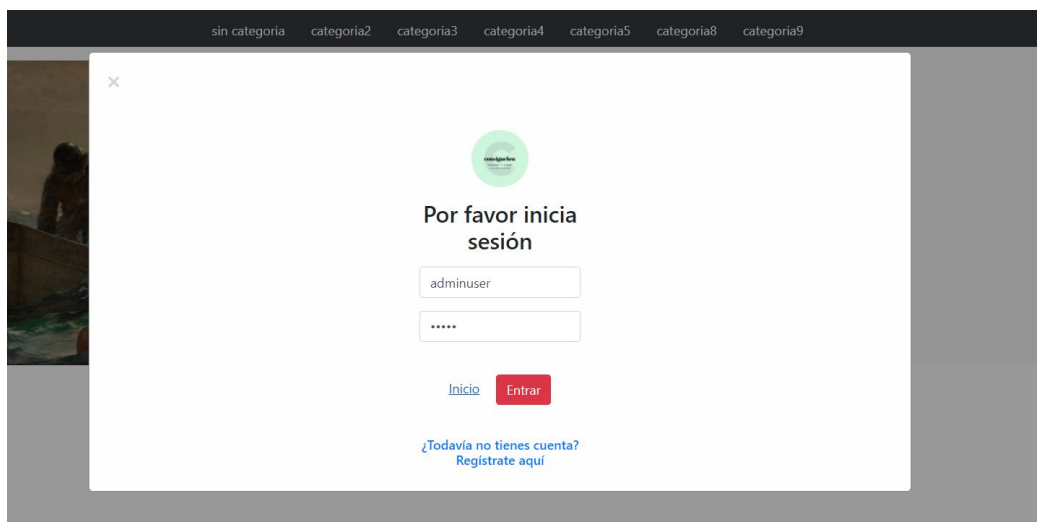


- En la zona inferior del navbar, se cargan y muestran las categorías con algún producto, aunque luego se pueda mostrar vacía por contener sólo productos del usuario actual. La vista enlazada a estos enlaces es similar a la portada.

- [productoDetalle.php](#):



Bajo el título del producto hay un enlace al perfil del vendedor. Comprar nos llevaría a la vista de caja donde realizamos el pedido. El botón adyacente varía mediante javascript, siendo inicialmente *Añadir al carrito*, y *Ver carrito* en caso de ya estar incluido en él. Si accedemos sin loguear, sólo aparece el botón comprar, que despliega también con javascript un modal para iniciar sesión:



- *Caja.php*: desde aquí pagaremos el pedido seleccionado. Es una vista responsive. Los checkbox no tienen funcionalidad. El apartado *Enviar a* puede convertirse mediante javascript en un formulario gestionado mediante Ajax para editar la información del comprador. Por último, para realizar el pago pulsaremos sobre el resumen de la factura en la parte derecha, *Confirmar y pagar*.

Enviar a

Álvaro abad

52906370E

calle falsa 123

691585787

Madrid

28022

1111222233334444

Editar

Nombre:

Álvaro abad

dni:

52906370E

direccion:

calle falsa 123

email:

correo@gmail.com

telefono:

691585787

ciudad:

Madrid

codigo postal:

28022

tarjeta credito:

1111222233334444

Confirmar

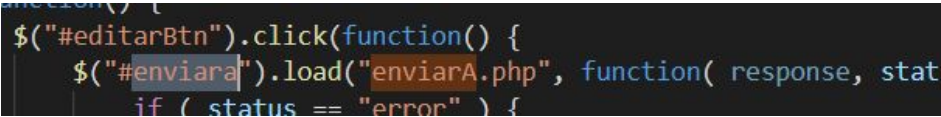
Cancelar

Modal de compra confirmada:



Arquitectura de la aplicación:

En el directorio principal se encuentran los scripts de vista y los de procesamiento de peticiones http:

- [adminProductos.php](#): vista para el administrador del listado de productos de la web.
- [adminUsuarios.php](#): vista para el administrador del listado de usuarios de la web.
- [anadeCategoria.php](#): vista para el administrador del formulario para añadir una categoría.
- **anadirPedido.php**: script de procesamiento de peticiones POST para añadir un producto. distingue entre compra sin estar en el carro, o estando incluido en el carro.
- **cancelaPedido.php**: script de procesamiento de peticiones post para eliminar pedidos.
- **cajusuariologueado**: vista que muestra el proceso de compra de un artículo que se quiere adquirir.
- **carro.usuarioLogueado**: vista del carrito de un usuario.
- **categoria.php**: vista de los productos de una categoría.
- **comprobarUsuario.php**: script de procesamiento de peticiones GET para comprobar en el registro si el nombre de usuario está disponible, al estilo del ejercicio 4.
- **eliminaCarritousuarioLogueado**: script de procesamiento de peticiones POST para eliminar productos del carro y descontar de la factura.
- **eliminaCat.php**: script de procesamiento de peticiones POST para eliminar categorías.
- **eliminaUsuario.php**: script de procesamiento de peticiones POST para eliminar usuarios.
- **enviarA.php**: crea y gestiona el formulario para editar la información del usuario en caja.php.


```
$("#editarBtn").click(function() {  
    $("#enviara").load("enviarA.php", function( response, stat  
    if ( status == "error" ) {
```
- **eliminaProducto.php**: script que procesa mediante peticiones POST la eliminación de un producto.
- **filtrar.php**: vista que procesa la búsqueda del searchbar y muestra el producto encontrado.
- **listaCategorias.php**: script de vista que solo le aparece al administrador, en la que se detallan todas las categorías existentes.
- **index.php**: página principal. Vista de los productos que no sean del usuario logueado ni estén pedidos y pagados.

- **vistaVendedor:** vista que se encarga de mostrar los artículos subidos por un vendedor en concreto.
- **vistaProdsUsuario:** script de vista que muestra los productos subidos por el usuario que tiene iniciada la sesión. Para que se pueda ver hay que pinchar en Mi Cuenta y luego en el nombre del usuario de la sesión.
- **vistaPedidos:** script de vista que muestra el numero de pedido y el articulo comprado.
- **vistaAdmin.php:** vista que solo le sale al usuario administrador, en la que le aparecen diferentes opciones para gestionar la web.
- **vender.php:** script de vista que muestra el html generado por *FormularioVender.php*
- **usuarioLogueado.php:** script que comprueba si estás logueado, y en caso contrario genera el formulario de login, que será recibido como respuesta por el javascript que lo solicita.
- **registro.php:** script que muestra el html generado por el formulario de registro.
- **productoDetalle.php:** script de vista que muestra detalladamente los artículos, indicando así su nombre, precio, talla, descripción y vendedor.
- **miPerfil.php:** script de vista en el que se ven los datos del usuario que tiene la sesión activa.
- **logout.php:** script que muestra el resultado de la acción de logout (cerrar la sesión).
- **login.php:** script que muestra el resultado de la acción de login (inicio de sesión) mediante el *FormularioLogin.php*.

En la carpeta includes están los archivos donde se declaran las clases de la aplicación, y también está la subcarpeta common :

- **Aplicacion.php:** estructura proyecto.
- **config.php:** estructura proyecto.
- **FormEditarInfo:** formulario que gestiona el cambio de la información personal de un usuario.
- **Categoria.php:** clase que se encarga de todas las acciones relacionadas con las categorías de la aplicación e interacciona con la base de datos.
- **Form.php:** clase padre de los formularios basado en estructura proyecto.
- **FormularioLogin.php:** formulario que se encarga de tramitar el login de un usuario.

- **FormularioRegistro.php**: formulario que hay que rellenar para procesar el registro de un usuario.
- **ImageUpload**: clase con la lógica de las imágenes subidas por el usuario. En el formulario de vender se envía el array FILES, cuyo valor es un atributo de esta clase. Contiene métodos para la validación segura de archivos como comprobar el tamaño, el tipo mime, cambio de nombre y formato. Al subir una imagen se crea un objeto imagen detallado a continuación. Si se intenta subir un archivo cuyo tipo mime no corresponde a una imagen no se efectuará la subida.
- **Imagen**: Clase para la gestión de objetos que representan imágenes. Representa información de la imagen en la bdd como el producto al que pertenece, el nombre en la bdd y el tipo mime. Implementa las operaciones de bdd. Se ha añadido una carpeta data/productos para alojar los archivos, con su correspondiente .htaccess
- **Usuario.php**: Este script se encarga de todas las operaciones que tengan que ver con el usuario.
- **Pedido.php**: Este script se encarga de todas las operaciones que tengan que ver con los pedidos.
- **FormEnviarA.php**: formulario Ajax que se encarga de la modificación de los datos de envío de un usuario.
- **FormularioCategoria.php**: formulario que se encarga de la creación de una nueva categoría que no exista ya.
- **FormularioVender.php**: formulario que gestiona la venta de un producto, comprueba también la correcta introducción de los datos.
- **Pedido.php**: clase que accede a la base de datos y gestiona todo lo relacionado con un pedido.
- **Producto.php**: clase que accede a la base de datos y gestiona los productos.
- **Usuarios.php**: clase que gestiona los usuarios.
- **Talla.php**: clase que representa las tallas.

Carpeta common: scripts comunes a todas las páginas.

- **cabecera.php**
- **footer.php**

La carpeta **img** contiene archivos de la web como iconos e imágenes.

La carpeta **data** está pensada para archivos subidos a la web, en este caso, en la subcarpeta **productos**, se encuentran las imágenes asociadas a los productos puestos en venta.

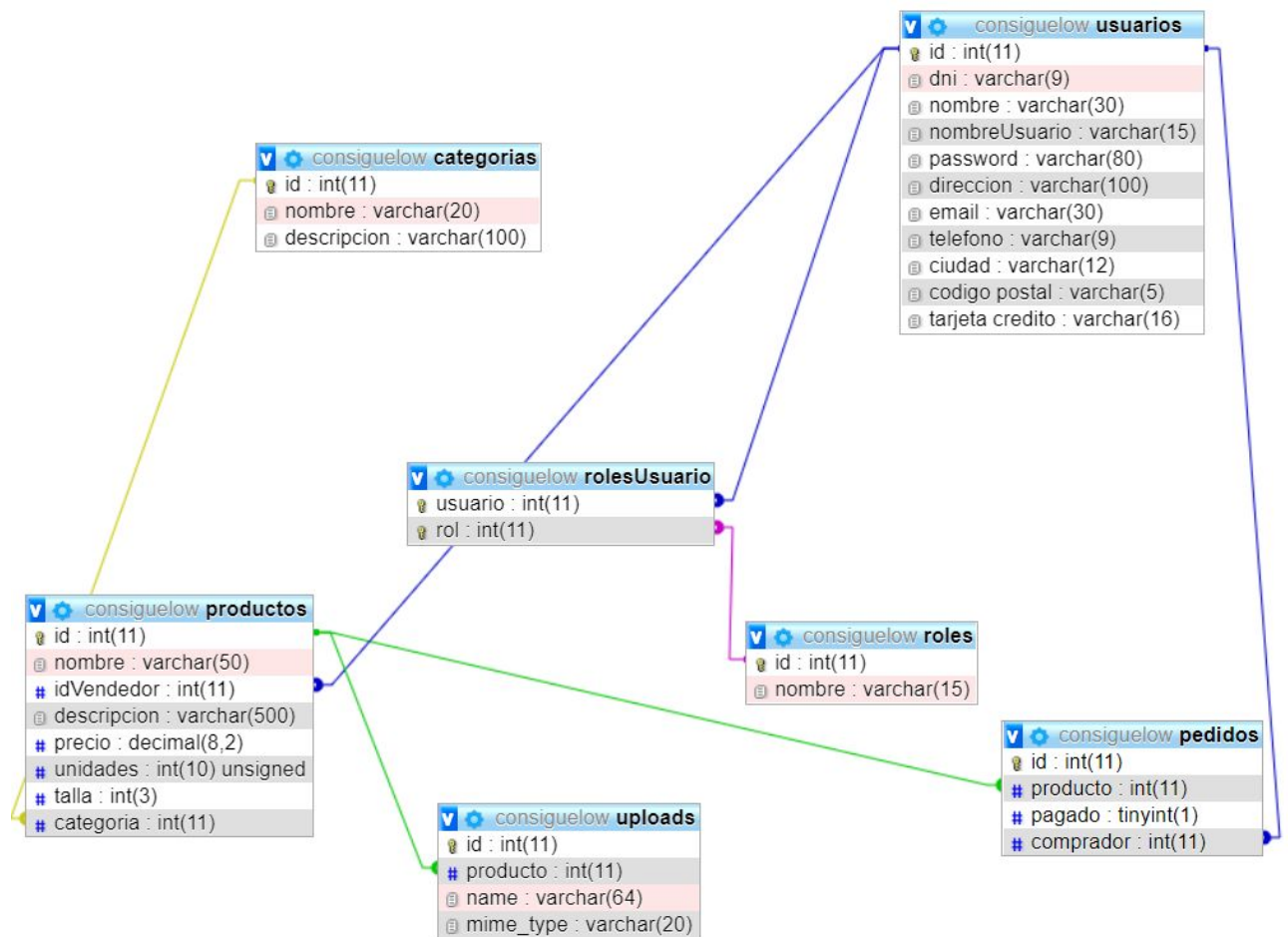
En la carpeta **css** además de los archivos de **Bootstrap**, se encuentra el archivo **modal.css** para los estilos del modal que se muestra si intentas comprar sin loguearte.

El directorio **js** contiene adicionalmente:

- **caja.js**: hace uso de **Jquery** para cargar en la vista de caja el formulario para editar la información del usuario, para enviar una petición a `anadirPedido.php`, y para crear el html del modal con el mensaje de éxito en la compra.
- **carro.js**: mediante **Jquery** realiza las funciones de eliminar del carro y recalcular la factura y de redireccionar a **caja.php**.
- **ejercicio4.js**: solicita a **comprobarUsuario.php** saber la disponibilidad de un nombre de usuario en el registro, y valida la expresión regular para comprobar caracteres inválidos.
- **productoDetalle.js**: realiza la función del botón *Añadir al carrito*, cambiándolo luego por *Ver carrito*. Pide a **usuarioLogueado.php** la comprobación para desplegar o no el modal del login. Implementa la función del botón Comprar (redirigir a *caja.php* con el id pertinente).

La carpeta **mysql** contiene:

- **datos.sql**: sentencias para poblar la base de datos. Importar con la base de datos creada. Desactivar la opción "Enable foreign key checks".
- **estructura.sql**: la database y el usuario se ha creado mediante la GUI del phpMyAdmin. Importar este script para crear las tablas. Desactivar la opción "Enable foreign key checks".
- **truncate.sql**: para vaciar las tablas de datos.
- **drop.sql**: para eliminar la base de datos.



Esquema de la base de datos.

Sobre el diseño de la base de datos:

Los productos del carro tienen el campo pagado a 0 para distinguirlos de los comprados, que lo tienen a 1. La tabla **uploads** guarda la información de las imágenes subidas por los usuarios. Los roles disponibles son: '1' para usuario normal, '2' para administrador.

Instrucciones de instalación:

La base de datos ha sido creada mediante la GUI de phpMyAdmin.

Las credenciales se encuentran en en config.php:

```
define('BD_NAME', 'consiguelow');

define('BD_USER', 'consiguelow');

define('BD_PASS', 'consiguelow');
```

```
/******LINUX******/
```

```
define('BD_HOST', 'vm05.db.swarm.test');  
  
define("F_PATH", R_PATH.'../data/productos/');
```

Importar **estrucutra.sql** para crear las tablas desactivando la opción Enable foreign key checks.

Se puede probar la web sin datos.

Para introducir los datos importar **datos.sql** desactivando la opción Enable foreign key checks.

Incluye 3 usuarios, con pedidos y productos. En el propio script

La contraseña para **adminuser** y **onlyuser** usuarios es '12345'

password_hash('\$2y\$10\$0eR.KhfTH5ybn/jlB86hwe/1nQeCKXk2RcLEjBscJbpUaF504kSOi',
PASSWORD_DEFAULT) == '12345'

Para **otrouser** es 'contrasenia'.

adminuser posee rol de administrador.

Importados los datos actualizamos la web y vemos en portada los 24 productos no comprados con sus respectivas imágenes (contenidas ya en el archivo). El phpMyAdmin se ha dejado con el usuario, la base de datos y las tablas creadas a falta de introducir los datos.