

Práctica 1

1.- Contenido de la aplicación

En este informe explicaré, página por página y siguiendo un flujo de aplicación lógico, que ha sido implementado, cómo y cuál es el estado final de cada una de las páginas.

1.1.- Home

El punto inicial lógico de cualquier página o aplicación web es su página de inicio o “home”. En esta práctica, al estar más centrada en un flujo de compra que en el diseño de la página en sí, no he dedicado más tiempo en la página de inicio que el que he necesitado para crearla.



Home works!

Comercio Electrónico | Práctica 1 | imolinava

La aplicación tiene compartida en toda ella el mismo header y footer desde el cual se puede buscar elementos, cambiar de idioma, ir a la página de inicio de sesión y ver el carrito.

Al acceder a la aplicación sin tener una sesión iniciada, crea un usuario invitado con un id `e_TIMESTAMP`. Este ID se utilizará para que cualquier persona, aunque no tenga usuario, pueda comprar.

1.2.- Inicio de sesión

El segundo paso lógico que haría un usuario es el de darse de alta en la aplicación para gestionar tanto sus compras actuales como sus futuras. Si hacemos click en el botón de *Login* de la cabecera nos vemos redirigidos a la página de inicio de sesión.

Email

Contrasenya

Iniciar sessió

Encara no tens conte?

[Regístrat](#)

Aquí encontramos un formulario el cual tiene una validación interna (formato de email válido y valores en ambos campos no vacíos) que si se cumple envía una petición al backend para que le devuelva el usuario vinculado a ese email (si la contraseña coincide). Por supuesto el usuario lo devuelve sin contraseña, de otro modo no sería seguro.


Si por contra no tenemos usuario, podemos hacer click en *Regístrat* para ir a la página de registro.

Nom

Cognoms

Email

Data de naixement

dd/mm/aaaa 

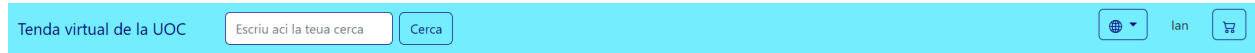
Contrasenya

Confirmar contrasenya

Regístrat

Igual que en el formulario de inicio de sesión, los inputs tienen que ser válidos para continuar con el registro de usuario. Por detrás, este comprueba que el email introducido no está en uso, y si no lo está crea un nuevo usuario.

En ambos casos, al recibir una respuesta de *success* del backend, la aplicación te redirecciona a la página principal y te carga en el `localStorage` del navegador el usuario y en las cookies compartidas el id del mismo.

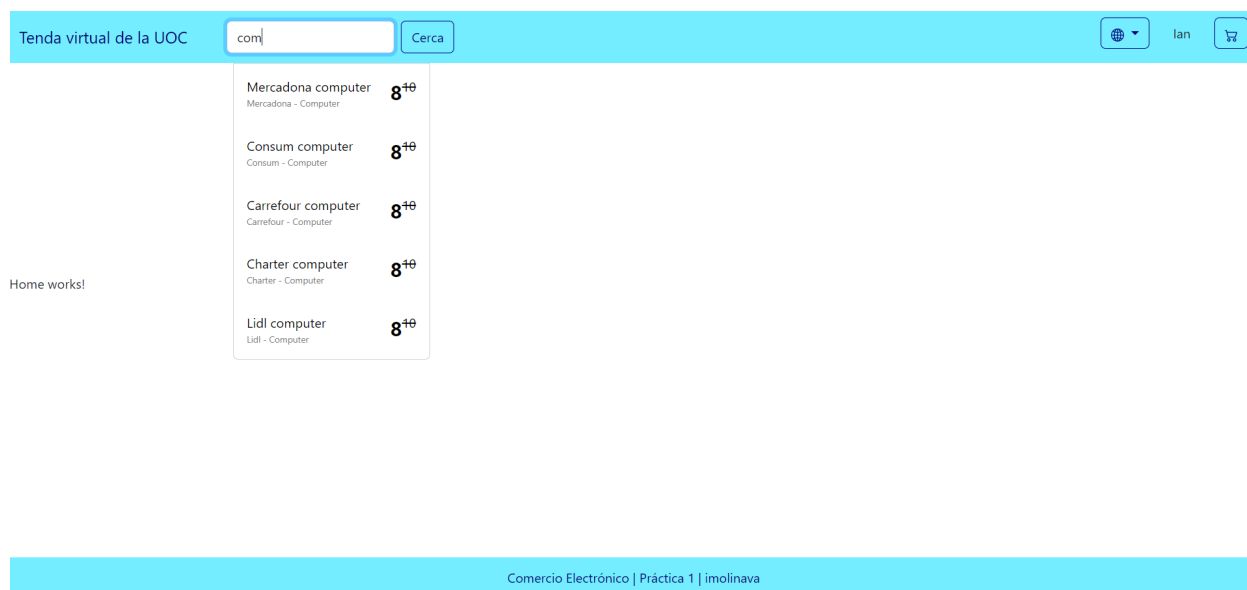


Home works!

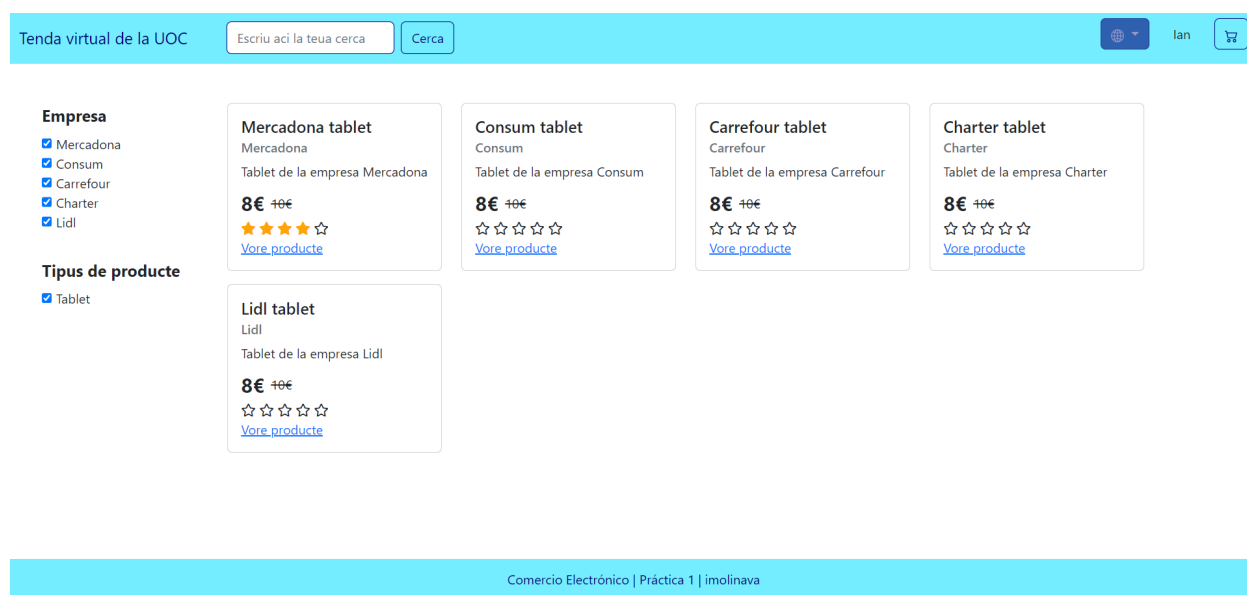
Decir que actualmente no se actualiza la cabecera automáticamente porque parece que la cookie tarda un poco en actualizarse, así que después de la redirección al iniciar sesión o registrarse hay que refrescar la página para que se vea reflejado el nombre del usuario en la cabecera.

1.3.- Búsqueda

Una vez el usuario ha iniciado sesión, lo que querrá es buscar productos. Para ello, desde la cabecera de la aplicación tiene un input en el que realizar búsquedas:



Este buscador busca cadenas de texto similares en los títulos de los productos (se podría ampliar a que buscara también en las descripciones). Una vez devueltos los resultados, se puede hacer click en uno de ellos para acceder a la página del producto, o hacer click en *Buscar* o pulsar la tecla *Enter* para acceder a la página de búsqueda de producto.





Esta página carga por URL el texto a buscar y devuelve el conjunto de productos que satisfacen esa búsqueda. Además, tiene en su columna izquierda un filtro dinámico que se carga con las empresas de los productos devueltos y los tipos de los productos devueltos.

Al interactuar con este filtro, actualizamos las vistas por medio de JavaScript, para evitar hacer más llamadas al backend de las necesarias.

Tenda virtual de la UOC

Cerca

lan

Empresa

- ☒ Mercadona
- ☐ Consum
- ☐ Carrefour
- ☒ Charter
- ☒ Lidl

Tipus de producte

- ☒ Tablet

Mercadona tablet

Mercadona

Tablet de la empresa Mercadona

8€ ~~10€~~

★★★★☆

[Vore producte](#)

Charter tablet

Charter

Tablet de la empresa Charter

8€ ~~10€~~

☆☆☆☆☆

[Vore producte](#)

Lidl tablet

Lidl

Tablet de la empresa Lidl

8€ ~~10€~~

☆☆☆☆☆

[Vore producte](#)



Comercio Electrónico | Práctica 1 | imolina

1.4.- Producto

Una vez encontrado el producto deseado, podemos acceder a su página personal haciendo click en *Ver producto*, o haciendo click en el producto mostrado en los resultados del buscador de la cabecera.

Tenda virtual de la UOC

Cerca

lan

Mercadona tablet

Mercadona - Tablet

8€ ~~10€~~

Tablet de la empresa Mercadona

★★★★☆

Quantitat

Añadir al carrito

Productos del mismo tipo

Consum tablet

Consum

Tablet de la empresa Consum

8€ ~~10€~~

★★★★☆

Carrefour tablet

Carrefour

Tablet de la empresa Carrefour

8€ ~~10€~~

★★★★☆

Charter tablet

Charter

Tablet de la empresa Charter

8€ ~~10€~~

★★★★☆

Lidl tablet

Lidl

Tablet de la empresa Lidl

8€ ~~10€~~

★★★★☆

☆☆☆☆☆
[Vore producto](#)

☆☆☆☆☆
[Vore producto](#)

☆☆☆☆☆
[Vore producto](#)

☆☆☆☆☆
[Vore producto](#)

Productos de la misma compañía

Mercadona computer
Mercadona
Computer de la empresa Mercadona
8€ ~~10€~~
☆☆☆☆☆
[Vore producto](#)

Mercadona phone
Mercadona
Phone de la empresa Mercadona
8€ ~~10€~~
☆☆☆☆☆
[Vore producto](#)

Mercadona TV
Mercadona
TV de la empresa Mercadona
8€ ~~10€~~
☆☆☆☆☆
[Vore producto](#)

Mercadona car
Mercadona
Car de la empresa Mercadona
8€ ~~10€~~
☆☆☆☆☆
[Vore producto](#)



Esta página muestra la información del producto, así como productos del mismo tipo y de la misma empresa. Destacar que no se ha implementado un sistema de recogida de imagen del producto, por lo que a lo largo de la aplicación los productos aparecerán sin imagen.

En esta misma página podemos añadir productos al carrito, modificando la cantidad de productos que queramos. Una vez seleccionada la cantidad y añadido, el carrito de la cabecera se actualizará de forma automática.

Tenda virtual de la UOC

Esriui aci la teua cerca

Cerca

 Ian 

Mercadona tablet

Mercadona - Tablet

8€ ~~10€~~

Tablet de la empresa Mercadona

★★★★☆

Quantitat

[Afezir a la cistella](#)

Productos del mismo tipo

Consum tablet
Consum
Tablet de la empresa Consum
8€ ~~10€~~
^ ^ ^ ^ ^

Carrefour tablet
Carrefour
Tablet de la empresa Carrefour
8€ ~~10€~~
^ ^ ^ ^ ^

Charter tablet
Charter
Tablet de la empresa Charter
8€ ~~10€~~
^ ^ ^ ^ ^

Lidl tablet
Lidl
Tablet de la empresa Lidl
8€ ~~10€~~
^ ^ ^ ^ ^

En este apartado, pese a que se ha añadido un valor de reseñas al producto, faltaría un apartado en el que los usuarios pudieran puntuar el producto y añadir comentarios. La base de datos está preparada para permitir esto, por lo que en una iteración futura se podría implementar.

1.5.- Carrito de la compra

Habiendo añadido al carrito los productos deseados, podemos observar el contenido del mismo haciendo click en el icono de carrito de la parte superior derecha de la página. Esto nos despliega un listado de todos los elementos que hemos añadido al carrito, en los que podemos hacer click para acceder de nuevo a la página del producto, y un botón que nos dirige a la página del carrito.

The screenshot shows the product page for 'Mercadona tablet'. The header includes the store name 'Tenda virtual de la UOC', a search bar with the text 'Escriu aci la teua cerca', and a 'Cerca' button. On the right, there's a user profile 'Ian' and a shopping cart icon with a red notification bubble. The product details section shows 'Mercadona tablet' with a price of 8€ (marked down from 10€), a description 'Mercadona - Tablet', and a rating of 4.5 stars. A quantity selector is set to 3, and there's a blue button 'Afegeir a la cistella'. A floating cart overlay on the right shows 'Mercadona tablet 8€' (marked down from 10€), 'Total: x3', and a button 'Ir al carrito'. Below the product details, a section 'Productes del mismo tipo' displays four other tablet products: 'Consum tablet' (8€), 'Carrefour tablet' (8€), 'Charter tablet' (8€), and 'Lidl tablet' (8€), each with a brief description and a rating.

Desde esta página, podemos realizar varias acciones. Modificar la cantidad de productos que tenemos en el carrito por medio de un desplegable, eliminar el producto del carrito o tramitar el pedido.

The screenshot shows the shopping cart page. The header is identical to the previous page. Below the header, a message says 'Benvingut Ian Molina!'. The cart items section displays 'Mercadona tablet' with a price of 8€ (marked down from 10€). To the right, there's a quantity selector set to 3 with a red trash icon, and a 'Total: 24€' label. A blue button 'Tramitar encomana' is at the bottom left. The footer of the page reads 'Comercio Electrónico | Práctica 1 | imolinava'.

1.6.- Tramitación de la compra

Haciendo click en el último elemento comentado, pasamos a la página de tramitación de la compra. Aquí volvemos a observar un listado de los productos que vamos a comprar, y si bajamos vemos un listado de métodos de pago y direcciones de envío vinculados al usuario.

Tenda virtual de la UOC

Escriu aci la teua cerca

Cerca

Ian

Resumen del pedido

Mercadona tablet

8€ ~~10€~~

Quantitat: x3

Total: 24€

Mètode de pagament

Paypal: ianmolina@papal.com

Afegir mètode de pagament

Direcció d'enviament

Street: C/Alacant, 20, 25

Afegir direcció d'enviament

Total: 24€

Tramitar encomana

Comercio Electrónico | Práctica 1 | imolinava

En ambos listados podemos añadir nuevos elementos.

Mètode de pagament

Paypal: ianmolina@papal.com

Afegir mètode de pagament

Tipus

Paypal

Usuari de paypal/número de tarjeta de crèdit

Guardar mètode de pagament

Direcció d'enviament

Street: C/Alacant, 20, 25

Afegir direcció d'enviament

Nom

Cognoms

Carrer

Codi postal

País

Telèfon

Email



Guardar direcció d'enviament

Una vez hemos añadido el método de pago y la dirección de envío, seleccionamos un valor de cada y tramitamos el pedido.

Tenda virtual de la UOC

Escriu aci la teua cerca

Cerca

lan

Resumen del pedido

Mercadona tablet

8€ ~~10€~~

Quantitat: x3

Total: 24€

Mètode de pagament

Paypal: ianmolina@papal.com

Afegir mètode de pagament

Direcció d'enviament

Street: C/Alacant, 20, 25

Afegir direcció d'enviament

Total: 24€

Tramitar encomana

Comercio Electrónico | Práctica 1 | imolinava

1.7.- Resumen del pedido

Al tramitar la compra, pasamos a una página en la que mostramos de nuevo un listado de los productos, el total y la fecha en la que se tramitó el pedido. Además, se puede acceder por URL a los pedidos por su ID, por supuesto comprobando que el usuario que ha iniciado sesión es el que está intentando acceder al pedido.

Tenda virtual de la UOC

Escriu aci la teua cerca

Cerca

lan

Mercadona tablet

8€ ~~10€~~

Quantitat: x3

Total: 24€

Fecha tramitació encomana: 2022-12-03 12:20:23

Total: 24€

Comercio Electrónico | Práctica 1 | imolinava

2.- Estructura de la aplicación

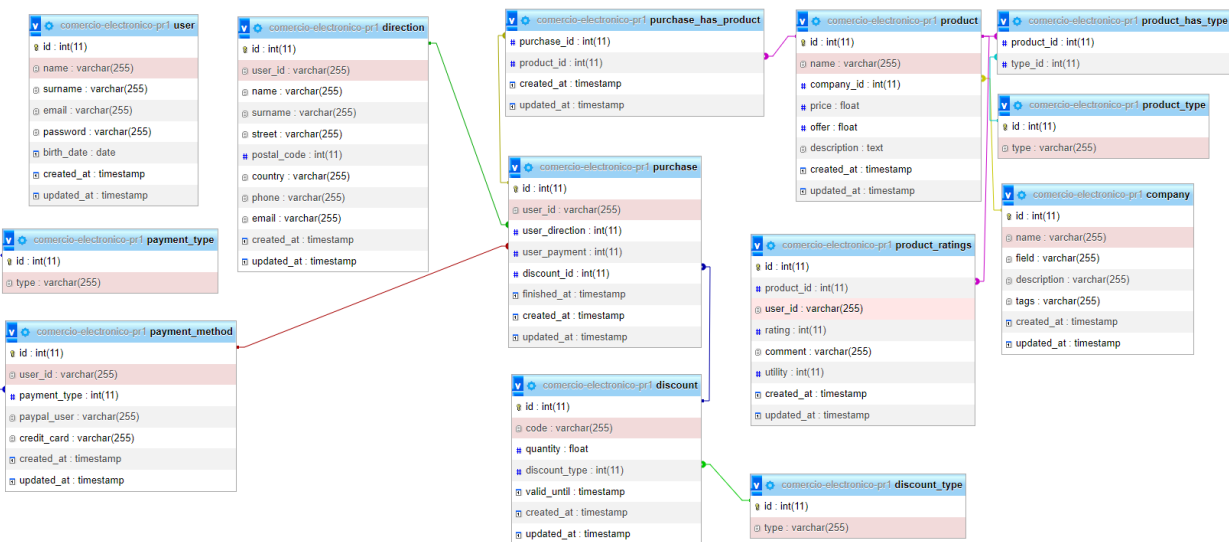
El proyecto está alojado en github en el [siguiente enlace](#), tiene un fichero README.md actualizado en el que se puede ver el estado de la aplicación y una explicación de la estructura de la misma, además de las tareas hechas y por hacer.

Está desarrollado con PHP para la parte de back, y HTML + CSS +JS para el front. La estructura de ficheros del proyecto sigue un patrón parecido al de Angular, en que tenemos, por cada página un mínimo de 3 ficheros, además de un cuarto si el componente en particular tiene llamadas a XMLHttpRequest:

- *componente.php*: controlador de la vista, se encargará de inicializar los datos necesarios.
- *componente.view.phtml*: contiene la vista del componente.
- *componente.styles.css*: en el que se incluyen los estilos que hacen efecto sobre la vista y se incorpora de forma automática en la cabecera del HTML.
- *componente.scripts.js*: en el que se incluye toda la lógica de la aplicación que se ejecuta sobre el navegador (control de formularios, interacciones de usuario, etc...).

BBDD

Base de datos MySQL con el siguiente diagrama:



El mismo objeto *purchase* se utiliza para el carrito de compra y compras finalizadas, la única diferencia es que el carrito de la compra no tiene fecha de finalización y las compras finalizadas sí.

El id del usuario no es una foreign key ya que no se están guardando los usuarios invitados, por lo que para guardar los métodos de pago y direcciones hacía falta acceder a un id que no estuviera apuntando a la tabla usuario.

Otra parte que ha faltado implementar en la práctica ha sido los descuentos. Desde un panel de control de administración se podría haber añadido los diferentes descuentos, que el usuario podría haber añadido a la hora de tramitar el pedido.

El fichero *database/comercio-electronico-pr1.sql* contiene toda la estructura de tablas y valores de la base de datos, preparado para importarlo y poder acceder a los datos.

Backend

Hay dos ficheros diferentes que vale la pena mencionar:

index.php

Se encarga de gestionar las funcionalidades transversales de la aplicación, tales como:

- Conexión a la base de datos
- Actualización de idiomas (cuyas traducciones se encuentran en los ficheros *internationalization/idioma.php*) en la variable `$_COOKIE`.
- Gestiona las rutas de la aplicación, redirigiendo dependiendo de la variable `$_SERVER['REQUEST_URI']` a un controlador u otro.

Ficheros *componente.php*

Son los ficheros sobre los que se hacen las llamadas XMLHttpRequest. Sirven para recibir llamadas POST o GET, realizar llamadas a la base de datos y devolver los resultados al componente.

Además se han generado modelos basados en la estructura de las diferentes tablas de la base de datos, aunque no se ha implementado la funcionalidad de que las respuestas de las peticiones a la base de datos por el backend se guarden en estos objetos.

Frontend

Desarrollado en HTML, CSS y JavaScript, con apoyo del gestor de librerías NPM, y con una estructura basada en Angular de MVP.

El fichero *page.php* es cargado por el gestor de rutas de *index.php* e incorpora su vista (*pages/page/page.view.php*). Una vez incluida esta, se importa en la misma su fichero JavaScript (*pages/page/page.scripts.js*) y por medio de la función *updateStyles* del fichero *pages/scripts.js* se actualiza en la etiqueta HEAD de la aplicación el link a los estilos contenidos en *pages/page/page.styles.css*.

El fichero *pages/scripts.js* contiene las funcionalidades:

- *updateStyles*: lee la ruta del componente y añade en la etiqueta HEAD de la aplicación la etiqueta LINK que apunta a los estilos del componente.
- *alert*: muestra una alerta de bootstrap, suele usarse en respuesta de error del backend.
- *createCookie*: genera una cookie.

Además se han generado diferentes módulos de JavaScript por funcionalidades agrupados en servicios:

- *http.service*: incluye los diferentes tipos de llamadas XMLHttpRequest (GET, POST, PUT) y devuelve una *Promise* a la que el usuario puede acceder cuando se resuelve.
- *loader.service*: genera un *overlay* con un icono de cargando cuando es llamado, suele ser llamado por el servicio anterior si recibe un parámetro específico como *true* para que lo muestre cuando empieza la llamada y lo esconda al finalizarla.
- *products.service*: utilizado para buscar productos desde el buscador de la cabecera.
- *shopping-cart.service*: gestiona todo el flujo del carrito, desde recibirlo al cargar la web, hasta modificar su contenido.
- *user-management.service*: se encarga de gestionar el inicio de sesión y registro de usuario.

3.- Comentario final

En general considero que la práctica ha ido bien. Me habría gustado tener más tiempo para finalizar implementaciones a las que no he llegado (como por ejemplo el panel de control para administradores desde donde comprobar la información de los usuarios, o la página de perfil de usuario desde donde actualizar su información), pero estoy contento con la forma en la que he abordado el desarrollo.

Al trabajar como desarrollador web frontend, más específicamente como desarrollador Angular, hacía mucho tiempo que no trabajaba con PHP, e incluso que no trabajaba con vanilla JavaScript para realizar una página completa de esta magnitud, así que me ha gustado el proyecto ya que he sido capaz de volver a trabajar con estas tecnologías, e incluso volcar los conocimientos que he ido desarrollando a lo largo de los años de experiencia en el sector.

La parte que más me ha costado, por supuesto, ha sido la parte de backend. Hace tiempo que no me encuentro en la situación de toma de decisiones respecto a modelos de datos, y en mis últimos trabajos las aplicaciones en las que he trabajado no han sido monolíticas (como es el caso de esta), sino que estaban separadas en frontend-backend, trabajando el backend únicamente como API y siendo el frontend el que se encarga de recibir los datos y trabajar con ellos.

Lo más interesante para mí a la hora de desarrollar la aplicación, ha sido la investigación de las comunicaciones entre ficheros y funcionalidades. Al haber ficheros que se llaman a veces en conjunto con la página, y otras veces como ficheros individuales (PHP que se carga al navegar a una página o al que se hace una petición XMLHttpRequest) han habido veces que he tenido que pegarme con el código para que la comunicación funcionara correctamente. Lo mismo ocurre con los ficheros JavaScript, que han tenido que ser importados como módulos para poder acceder a las funcionalidades que exportan.

En general, y como he dicho al principio, estoy contento con el resultado de la práctica y con el contenido de la misma, tengo ganas de ver como va la siguiente.