



CIFO

Proyecto final

Desiré Marrón Carmona

Proyecto: eCommerce utilizando HTML, CSS y vanilla JS

Índice

Índice	Página 00
Contexto de la aplicación	Página 01 - Página 07
Tecnologías utilizadas	Página 08
Proceso de desarrollo	Página 09 - Página 12
Puntos fuertes y débiles	Página 13
Futuras líneas de desarrollo	Página 14

1. Contexto

Este proyecto se ha desarrollado para el curso “**Desarrollo web Front End**”, impartido de forma telemática en el CIFO de Hospitalet de Llobregat.

El objetivo del curso es desarrollar aplicaciones web utilizando los lenguajes **HTML5** y **CSS3**, así como **JavaScript** y algunos de los frameworks más utilizados en el mercado web actualmente: **React**.

Durante el curso hay mención a la accesibilidad, la semántica, el uso de AJAX y JSON, CORS, jQuery y las nuevas funcionalidades de ECMA6.

2. Sobre el proyecto

El objetivo del proyecto era **utilizar y consumir una API de terceros utilizando HTML5, CSS3 y únicamente vanilla JavaScript**.


Es decir, **para este proyecto no se ha hecho uso de ningún framework web actual (React, Angular, Vue...)**, tan sólo se han utilizado librerías externas minimalistas para utilidades muy específicas, como la librería [CrispyToastJS](#) para crear pequeños toasts y [Bulma CSS](#) como framework CSS.

La elección de API era totalmente libre, así como el tipo de proyecto. En mi caso, quería hacer una **página de eCommerce**.

3. Estructura del proyecto

- Estructura de carpetas del proyecto
 -  **CSS**
 -  main.css
 -  **JS**
 -  adminPanel.js
 -  main.js
 -  users.js
 -  cart.js

Contexto de la aplicación

-  **PAGES**
 -  adminPanel.html
 -  cart.html
 -  login.html
-  index.html
-  Readme.md

- **Main.css**

En este fichero he añadido todos los estilos personalizados CSS que he incluido en el proyecto.

- **adminPanel.js**

En este fichero he programado la lógica necesaria para la página que simula ser el adminPanel.js del eCommerce.

- **main.js**

Este fichero contiene la lógica necesaria para el Home de la página, aquí se hacen las llamadas principales a la API, así como cargar el Home o cargar las categorías que seleccione el usuario o usuaria.

- **users.js**

Este fichero contiene la lógica necesaria para recuperar un usuario concreto de la API, cargarlo en el formulario de login y añadir las credenciales necesarias en el localStorage para recuperarlas más adelante. Main.js tiene la lógica necesaria para borrar esas credenciales del localStorage cuando sea necesario.

- **cart.js**

Este fichero contiene la lógica necesaria para gestionar el carrito de la compra. En este caso, la API que he elegido tiene usuarios y cada usuario tiene asignado su carrito con productos al azar, de modo que cargo un carrito al azar y con esos datos relleno la página. Elegí un carrito que tuviera varios productos para que el resultado en la UI fuera más interesante y variado.

- **adminPanel.html**

Este fichero HTML contiene lo necesario para cargar lo que simula ser un Admin Panel de un eCommerce. Se ve así:

Contexto de la aplicación

Logout

Welcome, kate@gmail.com!

Admin Panel			
	Products	Categories	Users
	Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops		
	Mens Casual Premium Slim Fit T-Shirts		
	Mens Cotton Jacket		
	Mens Casual Slim Fit		
	John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet		
	Solid Gold Petite Micropave		
	White Gold Plated Princess		
	Pierced Owl Rose Gold Plated Stainless Steel Double		
	WD 2TB Elements Portable External Hard Drive - USB 3.0		
	SanDisk SSD PLUS 1TB Internal SSD - SATA III 6 Gb/s		
	Silicon Power 256GB SSD 3D NAND A55 SLC Cache Performance Boost SATA III 2.5		
	WD 4TB Gaming Drive Works with Playstation 4 Portable External Hard Drive		
	Acer SB220Q bi 21.5 inches Full HD (1920 x 1080) IPS Ultra-Thin		
	Samsung 49-Inch CHG90 144Hz Curved Gaming Monitor (LC49HG90DMNXZA) - Super Ultrawide Screen QLED		
	BIYLACLESEN Women's 3-in-1 Snowboard Jacket Winter Coats		
	Lock and Love Women's Removable Hooded Faux Leather Moto Biker Jacket		
	Rain Jacket Women Windbreaker Striped Climbing Raincoats		
	MBJ Women's Solid Short Sleeve Boat Neck V		
	Opna Women's Short Sleeve Moisture		
	DANVOUY Womens T Shirt Casual Cotton Short		

Captura de pantalla del fichero HTML "adminPanel.html"

Esta pantalla que simula ser un Admin Panel contiene:

- Un navbar con un botón para hacer Logout **(1)**

1 Logout

- Un texto que recoge el email del usuario y lo muestra por pantalla **(2)**

Welcome, kate@gmail.com! 2

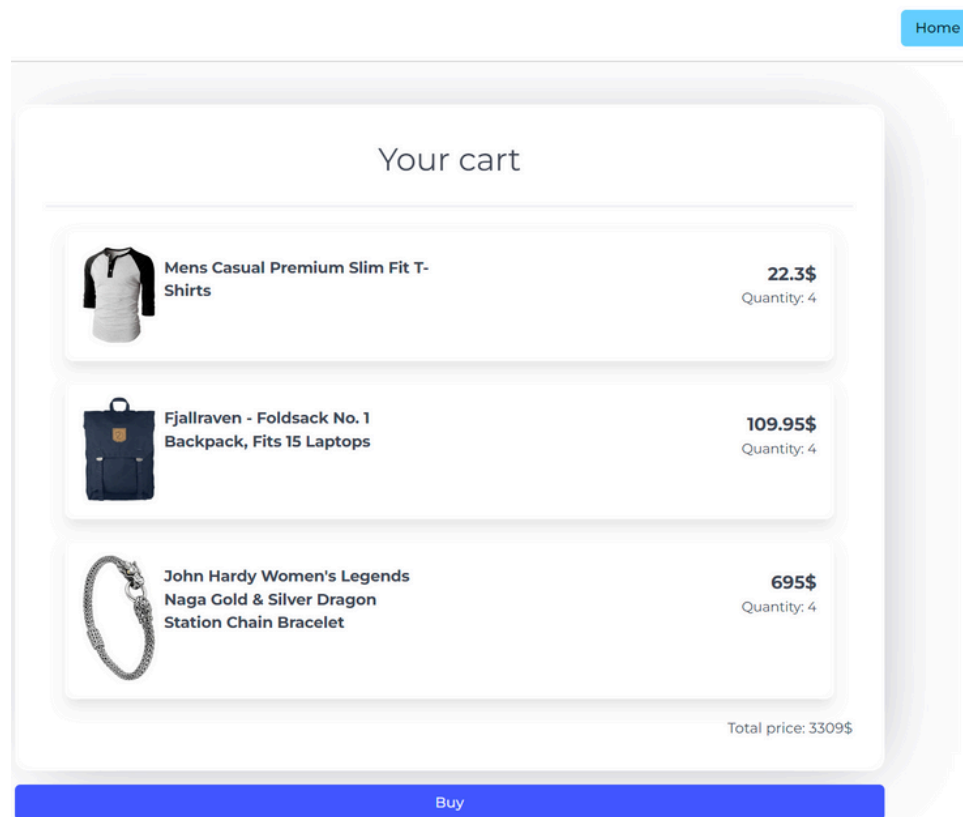
- Una tabla con 3 pestañas para cargar, respectivamente: productos, categorías y usuarios **(3)**

Admin Panel 3		
Products	Categories	Users
Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops		
Mens Casual Premium Slim Fit T-Shirts		

Contexto de la aplicación

- **cart.html**

Este fichero HTML contiene lo necesario para cargar lo que simula ser un carrito de compra de un eCommerce. Se ve así:



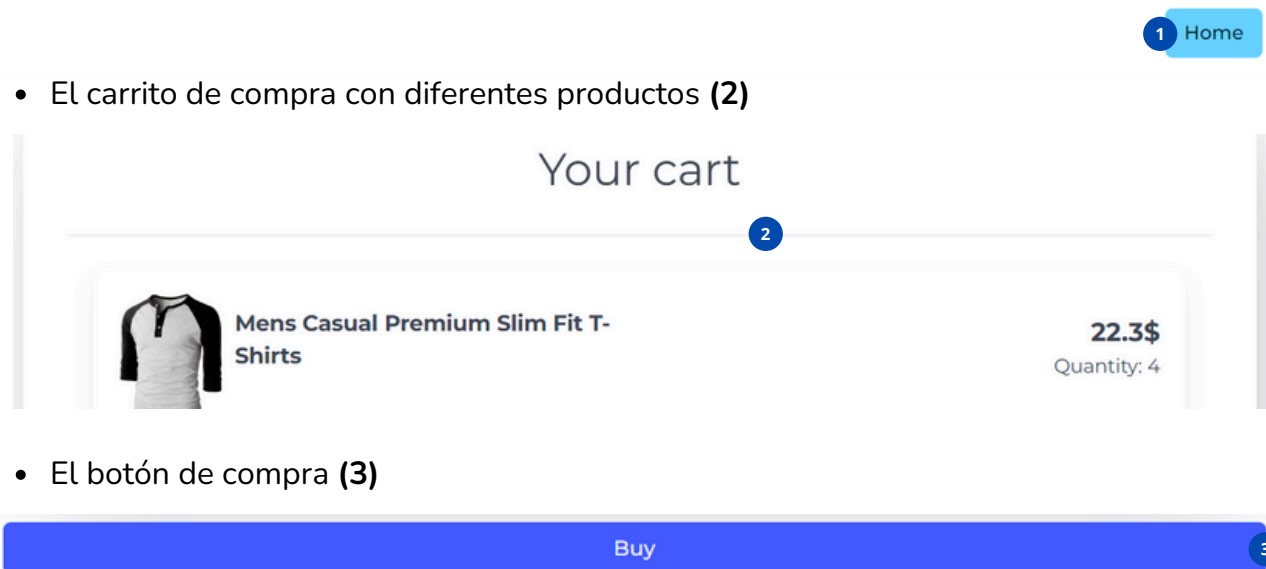
Captura de pantalla del fichero HTML "cart.html"

Esta pantalla que simula ser un carrito de compra contiene:

- Un navbar con un botón para volver al Home **(1)**

- El carrito de compra con diferentes productos **(2)**

- El botón de compra **(3)**



Contexto de la aplicación

- **login.html**

Este fichero HTML contiene lo necesario para cargar lo que simula ser una pantalla de Login. Se ve así:

Please use your user and password to enter the profile page

Log in

Sign in

Captura de pantalla del fichero HTML "login.html"

Esta pantalla que simula ser un carrito de compra contiene:

- Un formulario de Login cuyos campos ya se han rellenado mediante la lógica de la página **(1)**

- Dos botones. Uno de "log in" y otro de "sign in". **(2)**

Por ahora, sólo Login funciona. Una de las líneas de desarrollo a futuro sería desarrollar alguna lógica adicional para "Sign in" a nivel local, ya que la API que he utilizado de por sí no permite hacer modificaciones ni subidas al servidor.

Log in

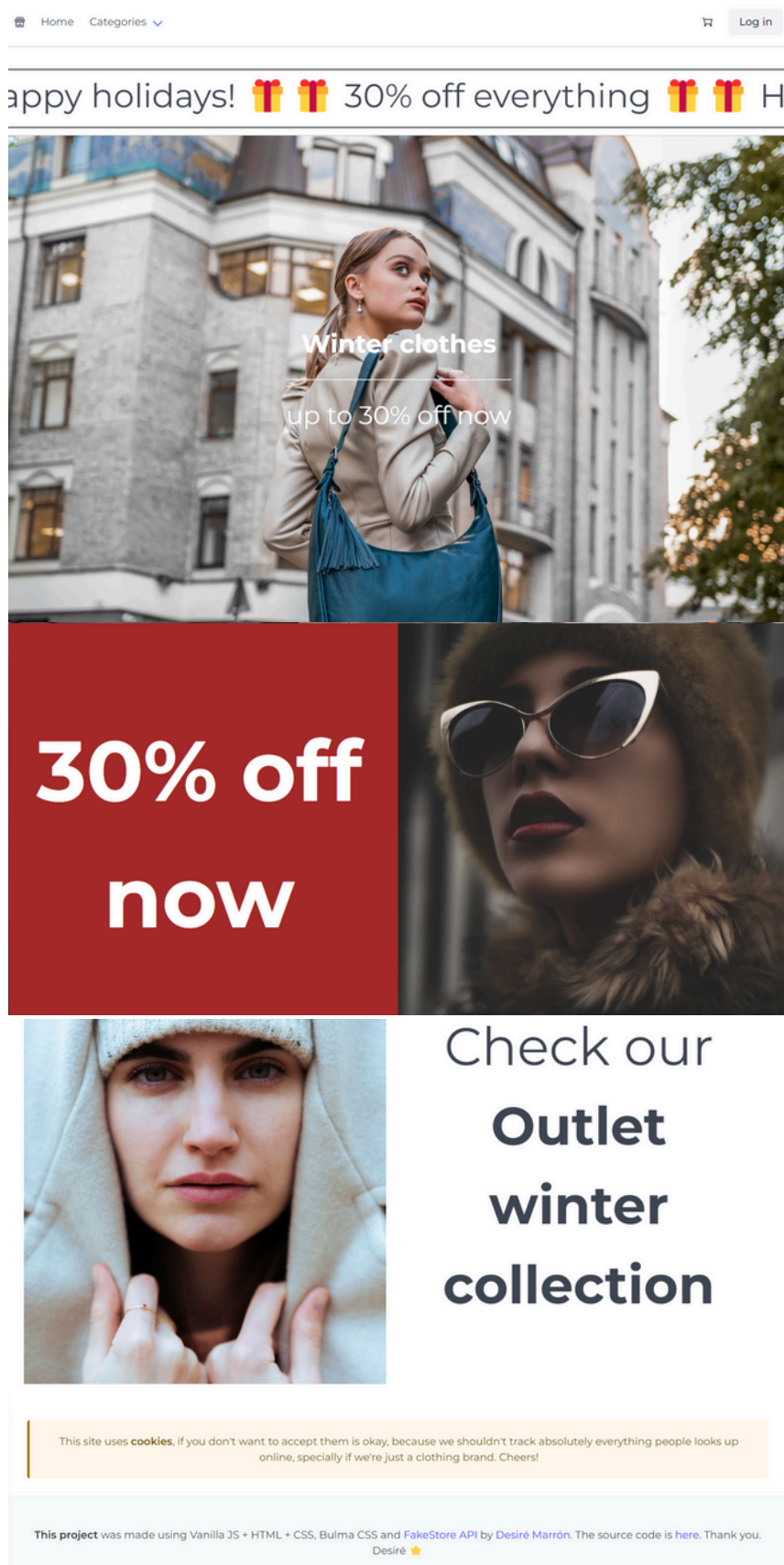
Sign in

- **index.html**

Finalmente, está el fichero index.html.

Este es el fichero HTML que hace a la vez de Home y también carga los productos de la categoría seleccionada por el usuario o usuaria. Se ve así:

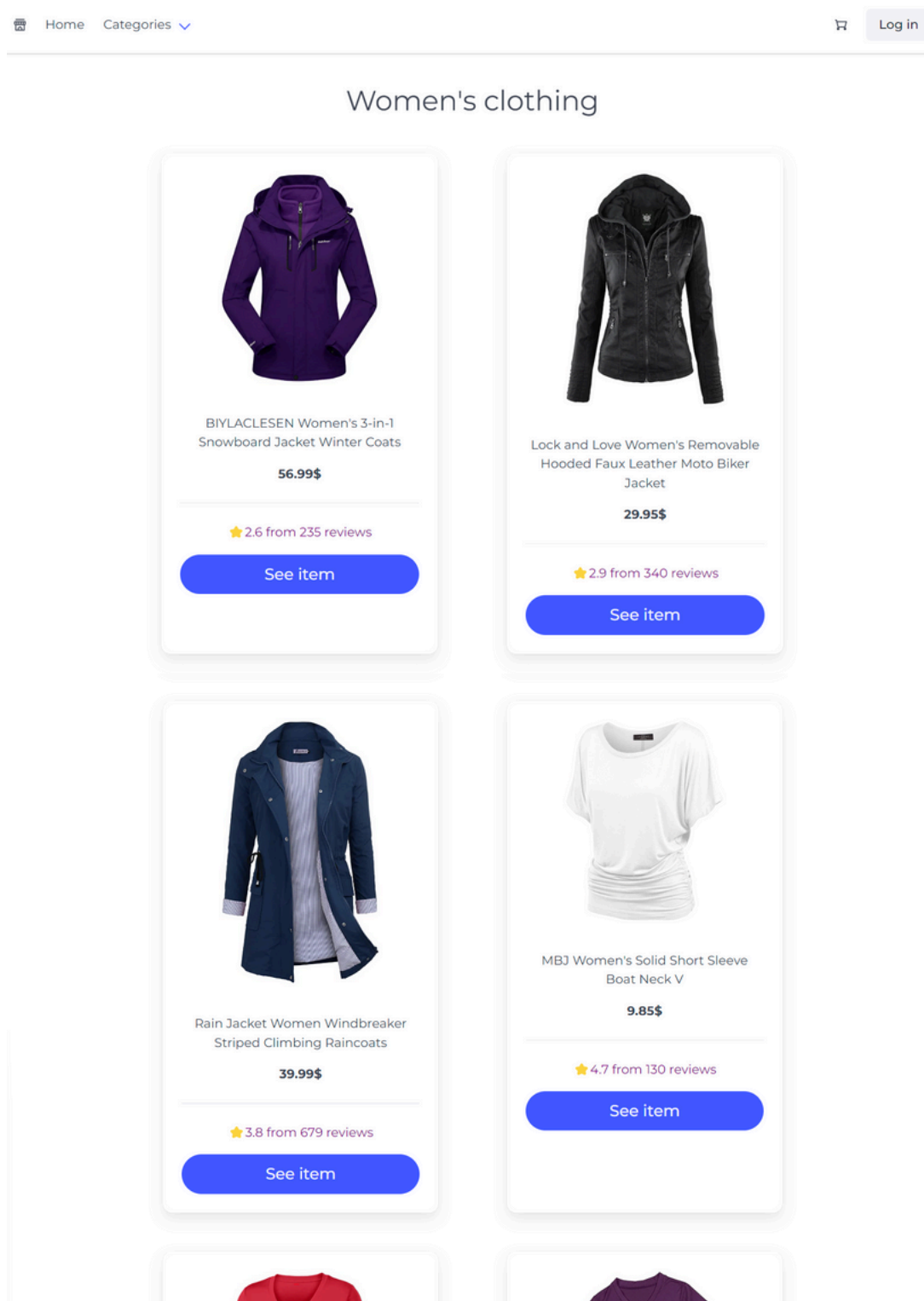
Contexto de la aplicación



Captura de pantalla del fichero HTML "index.html"

Contexto de la aplicación

Cuando una categoría es seleccionada desde el dropdown del navbar que dice “Categories”, la página cambia y se cargan las cartas de los productos de cada categoría. Se ve así:



Captura de pantalla del fichero HTML "index.html" tras seleccionar una categoría

1. Control de versiones

Para desarrollar este proyecto he utilizado [GitHub](#) y la [extensión de Git de Visual Studio Code](#).

[Aquí](#) está el repositorio público del proyecto desarrollado, así como la [lista de commits](#) del mismo.

2. IDE

Para desarrollar este proyecto he utilizado [Visual Studio Code](#) como editor de texto.

3. Lenguajes

Este proyecto debía desarrollarse utilizando: HTML5, CSS3, Vanilla JavaScript (*no se podía utilizar ningún framework web*).

4. Otros

Para este proyecto he utilizado **estilos CSS3 propios** (*main.css*), pero también me he apoyado en [BulmaCSS](#) para maquetar más rápidamente los componentes que necesitaba (*cards, tablas, paneles, carritos...*). Además he utilizado bastante la feature **Flexbox** de CSS3.

También he utilizado una **pequeña librería para crear toasts** de forma rápida y sencilla, llamada [CrispyToastUS](#), desarrollada por el usuario [raheelize](#) en Github y compatible con vanilla JavaScript.

Para este proyecto he utilizado la API [FakeStore API](#).

5. Bibliografía

Por supuesto, también he recurrido a páginas como [StackOverflow](#), [MDN](#), [Geeksforgeeks](#), [Dev.to](#) y [W3Schools](#), además de las guías de [Flavio Copes](#).

1. Funcionalidades deseadas

En este proyecto quería conseguir las siguientes funcionalidades para la página:

- Que el usuario/a pudiera crear una cuenta
- Que el usuario/a pudiera hacer login
- Que el usuario/a pudiera cargar productos clicando en su correspondiente categoría
- Paginar los productos añadiendo límites y rangos a la petición a la API
- Poder buscar productos a través de un buscador
- Poder mostrar los detalles de un producto seleccionado por el usuario/a
- Que el usuario/a pudiera ejercer de admin
 - modificando, borrando o creando categorías
 - modificando, borrando o creando usuarios
 - modificando, borrando o creando productos para una o varias categorías
- Crear un carrito de compra que el usuario/a pudiera gestionar
 - borrar o añadir productos
 - realizar una compra (*simulación*)

Leyenda

- Objetivo cumplido totalmente
- Objetivo cumplido parcialmente
- Objetivo no cumplido

2. Proceso por etapas

● Etapa 1

Objetivo(s):

- Pedir las categorías a la API y mostrarlas en la web. ■

Acciones:

- Creé el índice y un fichero simple de .js para testear la API
- Comprobé qué tipo de datos me devolvía la API para: categorías y productos
- Conseguí rellenar de forma automática las categorías de la página web con las categorías que me devolvía la API

● Etapa 2

Objetivo(s):

- Recuperar todos los productos de una categoría a través del input de usuario ■
- Mostrar todos los productos de esa categoría en la UI con sus datos. ■

Acciones:

- Creé un template en HTML y creé la funcionalidad de cargar todos los productos de una categoría y añadirlos a la UI cuando se clica en una categoría en el navbar
- Optimicé la lógica y añadí estilos visuales

Proceso de desarrollo

● Etapa 3

Objetivo(s):

- Mostrar los datos de un producto seleccionado por el usuario/a ■

Acciones:

- En la tercera etapa me centré en crear un modal en el que se cargasen todos los datos de un producto seleccionado a través de la web

● Etapa 4

Objetivo(s):

- Consultar los datos de usuario ■
- Simular un login de usuario ■
- Simular la funcionalidad de un Admin en un Admin Panel de eCommerce ■
- Consultar y mostrar por pantalla productos, categorías y email de usuarios ■
- Gestionar un logout ■

Acciones:

- Creé la lógica para recuperar usuarios
- Creé las páginas de Login y Admin Panel
- Creé la lógica para recuperar los datos de productos, categorías y email de usuarios para la página de AdminPanel y mostrarlos en una tabla

● Etapa 5

Objetivo(s):

- Crear página carrito de la compra ■
- Simular una interacción total con el carrito de compra (modificar-añadir-borrar productos del carrito) ■
- Conseguir el precio total del carrito ■
- Crear la página de Home ■
- Gestionar correctamente el journey desde Home - Categoría seleccionada - Login - Logout -> Volver al home ■
- Simular una compra en el carrito ■

Acciones:

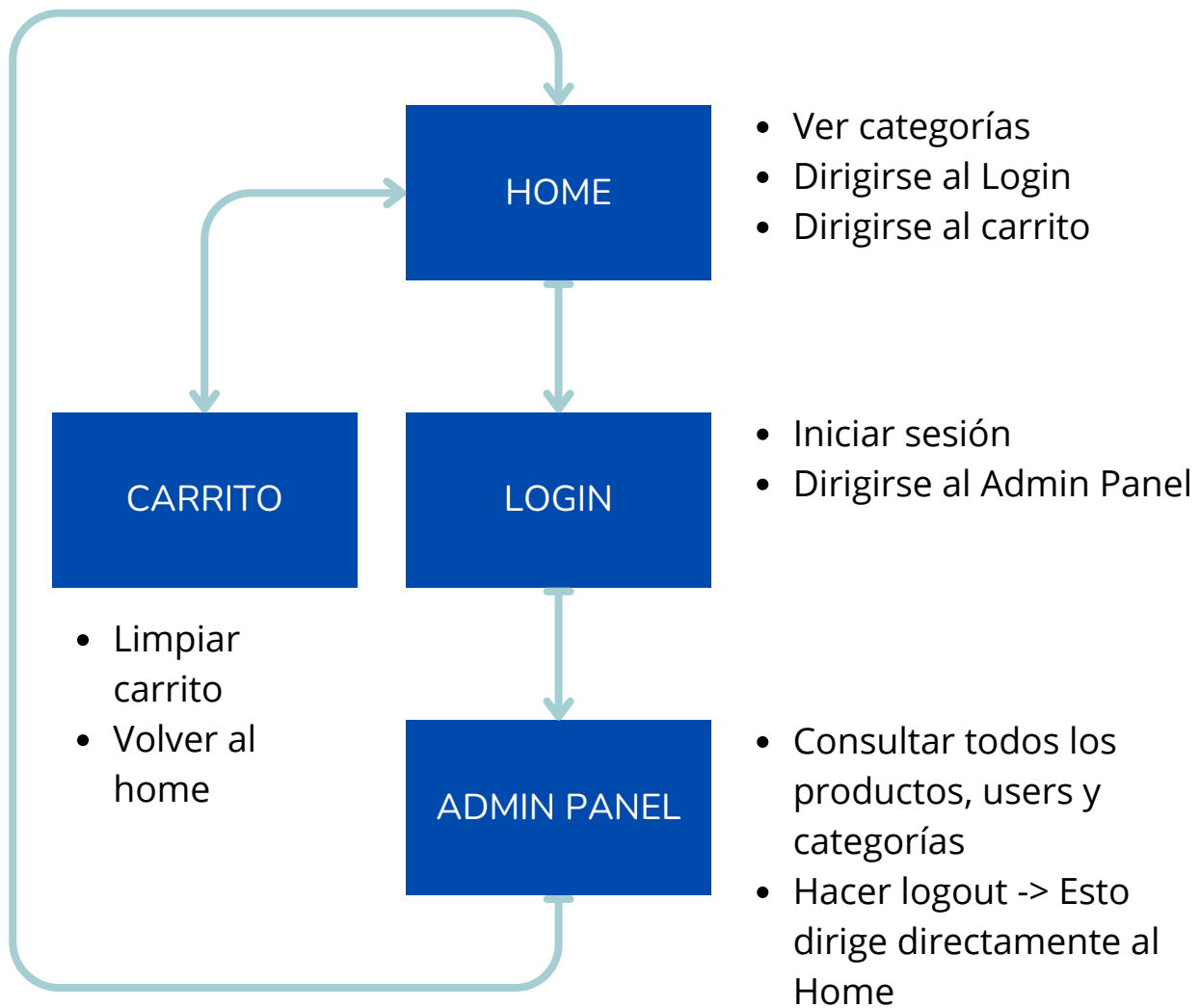
- Creé el carrito de la compra y su lógica
- Creé y estilicé el home
- Creé la lógica para gestionar mostrarlo/esconderlo cuando se va una categoría al home

3. Funcionalidades implementadas

Teniendo en cuenta las funcionalidades deseadas:

- Que el usuario/a pudiera crear una cuenta
 - Esta funcionalidad no se ha implementado todavía.
- Que el usuario/a pudiera hacer login
 - Esta funcionalidad **ha sido implementada parcialmente**. Actualmente cuando ingresas a Login, la página recupera un email y contraseña de la API y el botón de “Login” te redirige al Admin Panel.
- Que el usuario/a pudiera cargar productos clicando en su correspondiente categoría
 - Esta funcionalidad **ha sido implementada como se esperaba**.
- Paginar los productos añadiendo límites y rangos a la petición a la API
 - Esta funcionalidad no se ha implementado todavía.
- Poder buscar productos a través de un buscador
 - Esta funcionalidad no se ha implementado todavía.
- Poder mostrar los detalles de un producto seleccionado por el usuario/a
 - Esta funcionalidad **ha sido implementada como se esperaba**.
- Que el usuario/a pudiera ejercer de admin
 - modificando, borrando o creando categorías
 - modificando, borrando o creando usuarios
 - modificando, borrando o creando productos para una o varias categorías
 - Esta funcionalidad **ha sido implementada parcialmente**. Actualmente el usuario/a puede hacer interacciones simples con el panel de Admin, pero sólo puede visualizar datos, no puede editarlos, borrar o crear nuevos todavía.
- Crear un carrito de compra que el usuario/a pudiera gestionar
 - Esta funcionalidad ha sido **implementada parcialmente**. Sí existe un carrito, el cual recupera el carrito de un usuario y muestra los productos por pantalla, calculando también su precio final. El usuario/a puede clicar en el botón “Buy” y la lista se borrará. Obviamente, al recargar la página, los productos aparecerán de nuevo ya que no es una compra real.
 - borrar o añadir productos
 - realizar una compra (*simulación*)

4. Web map



1. Puntos fuertes

- Ya que la API es pequeña y devuelve pocos datos, la página carga muy rápidamente los datos para mostrar por pantalla.
- La estructura de la web es sólida, con partes claramente diferenciadas: home, espacio de categorías, carrito, pantalla de login, admin panel.
- Los elementos mostrados en la web son claramente visibles y el contraste es el adecuado.
- Las animaciones y decoraciones añadidas a la web no distraen ni son visualmente molestas.
- El uso de la página es sencillo y se aprende rápidamente.

2. Puntos débiles

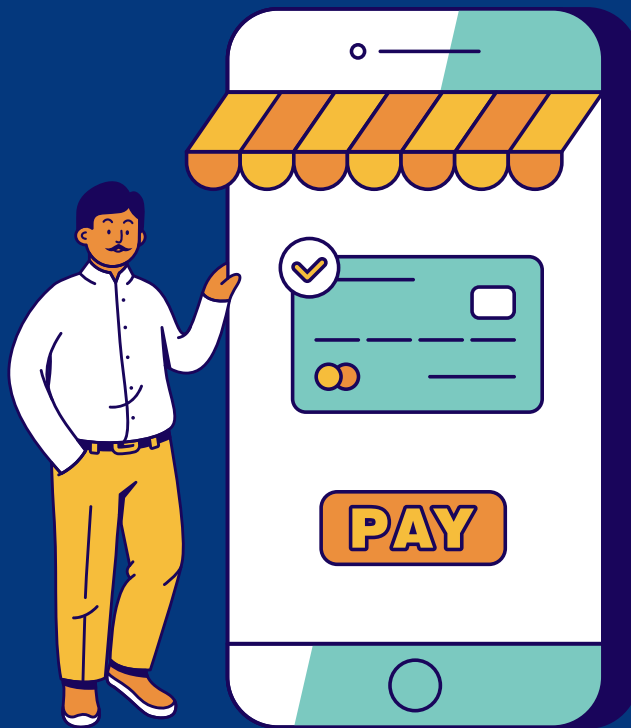
- Faltan partes importantes de la interacción de una web de este tipo (interacción completa del proceso de añadir un item al carrito, por ejemplo; interacción y proceso completo de registrar un usuario/a...)
- El estilo visual podría haberse pulido más.
- El código podría haberse documentado mejor y optimizarse más.
- No se visualiza correctamente en todos los dispositivos (responsive).

Futuras líneas de desarrollo

La intención es continuar con el desarrollo de este proyecto para completar, en la medida de lo posible, las funciones deseadas no implementadas o parcialmente implementadas.

Ciertamente no sé si este proyecto quedará como prototipo y a futuro o a finalizar el curso, decida rehacerlo con algún framework web (*Angular, React o Vue*) para mejorar su estructura y funcionamiento.

Sin duda creo que un proyecto como un eCommerce en un portafolio es altamente vistoso, así que no creo que abandone la idea incluso si no estoy del todo satisfecha con el resultado de este proyecto actual. Estoy segura de que al finalizar el curso tendré muchos más conocimientos con los cuales mejorar este y otros proyectos personales que realice.



CIFO

Proyecto final

Desiré Marrón Carmona