

# Proyecto de diseño de interfaces

En este proyecto se plantea realizar un diseño de un portal de productos electrónicos con cesta de compra permitiéndole al usuario seleccionar el producto o productos y posteriormente poder realizar la compra o eliminación del producto.


Como recursos se aportarán para su uso los documentos HTML y comun.css con los estilos CSS comunes.

- **Se importa una fuente personalizada:** *Rubik* desde Google Fonts.
- **Se definen colores reutilizables como variables CSS globales:**
  - --clr-main: verde oscuro (#45880e).
  - --clr-main-light: verde claro (#73ff00).
  - --clr-white: blanco cálido (#fff7e5).
  - --clr-gray: gris claro (#e2e2e2).
  - --clr-green: otro verde (#316b0a).
  - --clr-red: rojo oscuro (#961818).
- **Se aplican estilos básicos a todos los elementos para un diseño uniforme:** eliminación de márgenes y padding, uso de box-sizing: border-box y la fuente "Rubik" como predeterminada.
- Todas las medidas que se utilicen tendrán que ser relativas, en este caso mediante rem.
- Para el diseño responsivo se hará uso de flexbox y grid.

Por otra parte, se aporta una carpeta con las imágenes de los productos a utilizar y un archivo productos.json con la información de cada uno de los productos como su nombre, precio, imagen...

## Login

En esta interfaz, el usuario podrá introducir un nombre y contraseña para acceder al portal. Con el modo de facilitar la tarea y no ser necesario la implementación de un servidor u otras opciones el usuario podrá acceder con cualquier usuario o contraseña que cumpla las restricciones de formato requeridas.

The image shows a login form titled "Login" centered on a green background. The form has a light yellow background. It contains two input fields: the first is for the username, with "Martín" entered; the second is for the password, with asterisks indicating masked text. Below the password field, there is a red error message: "La contraseña debe tener entre 8 y 16 caracteres y solo puede contener letras, números y los caracteres +\$%&/().". At the bottom of the form, there are two buttons: a green "Entrar a CarbaShop" button and a dark green "Limpiar Datos" button.

Nótese que el login estará situado en el centro de la pantalla y tendrá que realizar las correspondientes validaciones que se indican a continuación.

### **Para el nombre de usuario.**

#### **Caso 1: El campo está vacío.**

- **Mensaje:** "Nombre obligatorio."
- Esto indica que el usuario no ha escrito nada en el campo del nombre. Es obligatorio rellenarlo.

#### **Caso 2: El formato del nombre es inválido.**

- **Mensaje:** "Nombre inválido."
- Esto ocurre si el nombre no cumple con el patrón especificado.
- Un nombre válido debe contener solo letras, incluyendo tanto letras mayúsculas (A, B, C, ...) como letras minúsculas (a, b, c, ...). También se aceptan letras con acentos o diacríticos, como Á, É, Í, Ó, Ú, Ü y Ñ, así como sus versiones en minúscula. Además, puede incluir espacios en blanco, lo que permite nombres compuestos como "Juan Carlos" o "María José". Sin embargo, no se permite

ningún otro tipo de carácter. Esto significa que no pueden incluirse números, como "123"; símbolos especiales, como @, #, \$, %; ni signos de puntuación, como puntos, comas o guiones.

### **Caso 3: El nombre supera el límite de caracteres.**

- **Mensaje:** "El nombre no puede tener más de 20 caracteres."
- **Por qué:** Limitar la longitud del nombre evita errores de almacenamiento o visualización.

### **Para la contraseña del usuario.**

#### **Caso 1: El campo está vacío.**

- **Mensaje:** "La contraseña es obligatoria."
- El usuario no ha ingresado ninguna contraseña. Este es un campo requerido.

#### **Caso 2: El formato de la contraseña es inválido.**

- **Mensaje:** "La contraseña debe tener entre 8 y 16 caracteres y solo puede contener letras, números y los caracteres ·\$%&/()."."
- La contraseña no cumple con los requisitos de seguridad o formato. Esos requisitos son:
  - Longitud: debe tener entre 8 y 16 caracteres.
  - Contenido: solo puede incluir letras, números y ciertos caracteres especiales como ·\$%&/().

Además, cuando el usuario se posicione sobre el botón correspondiente se modificará el color de este tal y como se indica en la imagen. En caso de superar todas las validaciones al pulsar en el botón "Entrar en CarbaShop" accedemos al portal.

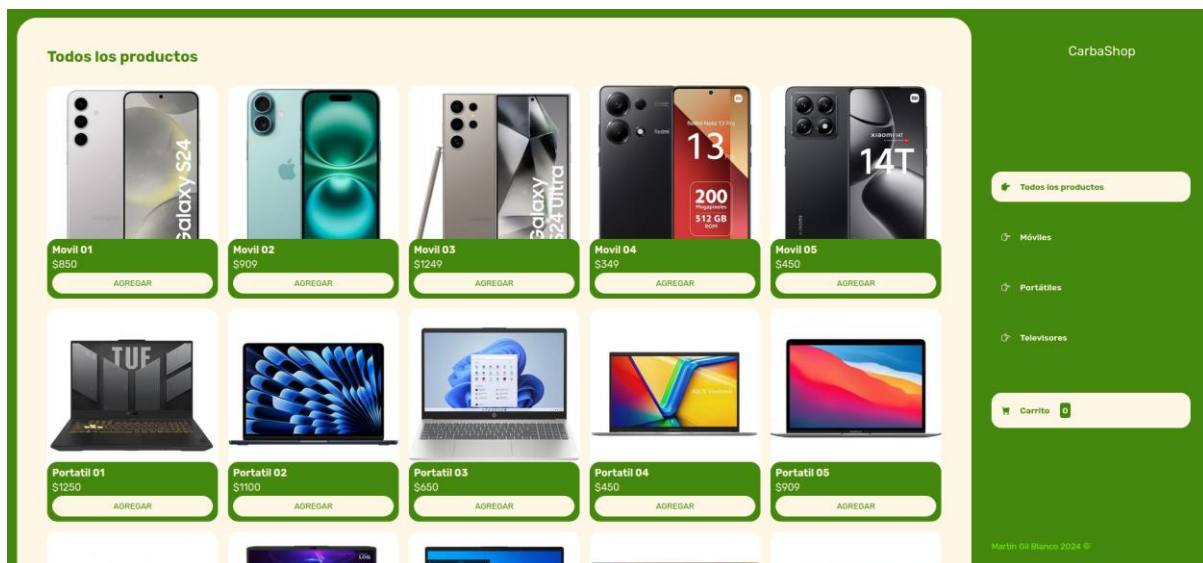
El botón "Limpiar datos" ha de limpiar todos los datos introducidos por el usuario en el formulario de login, así como las validaciones.

## CarbaShop

Una vez dentro del portal, el usuario podrá ver un listado de todos los productos tecnológicos que se ofrecen, los cuales se cargan de forma dinámica por javascript a través del archivo productos.json.

Existen pues dos secciones diferenciadas, por una parte, tenemos el apartado de productos donde se muestran los productos en función de la selección realizada por el cliente, así como el carrito de compra con el número de productos guardados, siendo las siguientes:

- Todos los productos, con la información de todos los productos tecnológicos ofertados.
- Móviles, donde únicamente se muestran los móviles.
- Portátiles, donde únicamente se muestran los portátiles.
- Televisores, donde únicamente se muestran los televisores.
- Carrito (número): Donde se indica el conjunto de productos reservados pasando a la vista de carrito si se pulsa en ella.

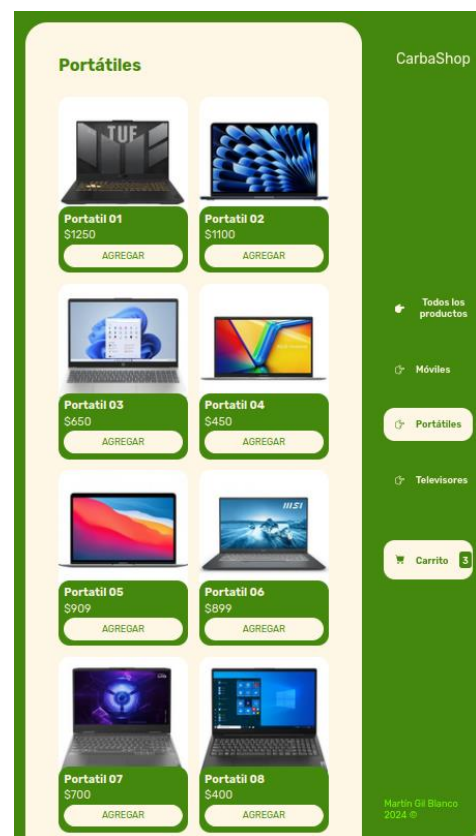
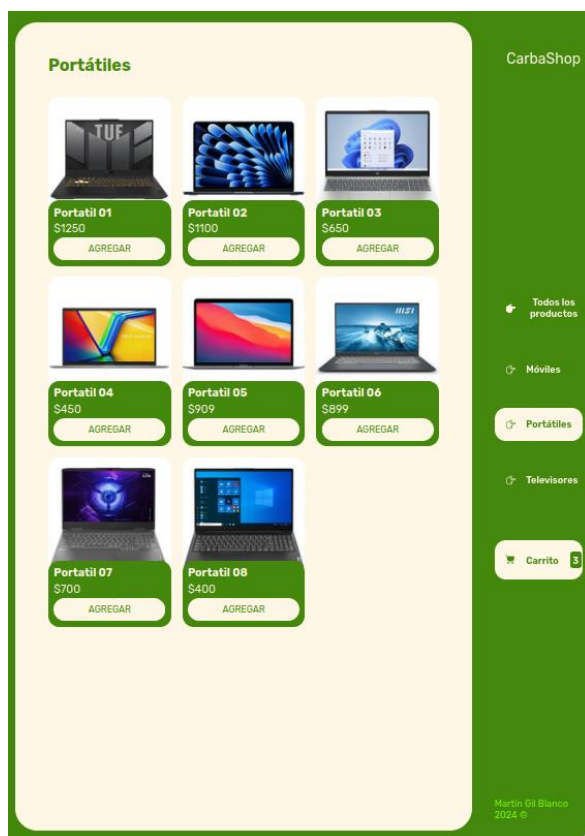


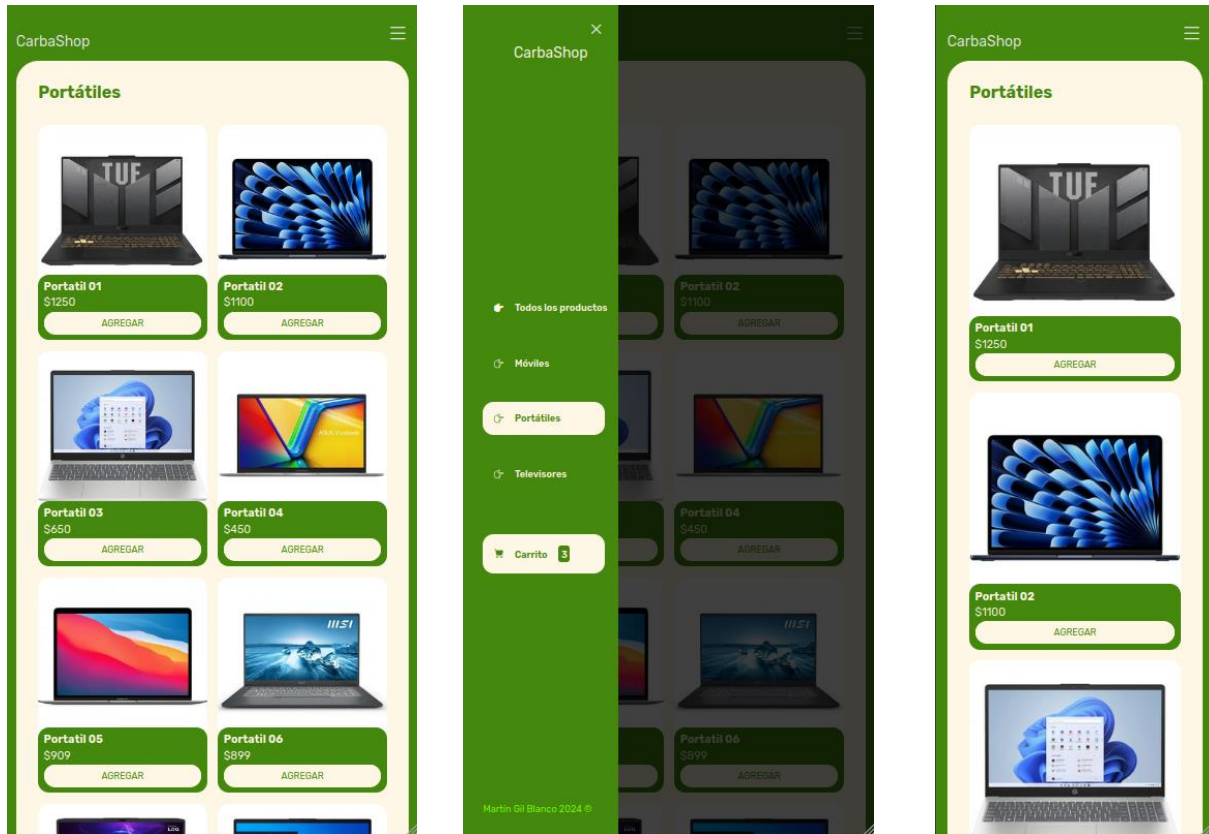
Además de lo anterior, el usuario puede pulsar en el texto de CarbaShop, sirviendo este para poder volver a la vista de login.

Cada uno de los productos tendrá el correspondiente botón de agregar, cada vez que pulsamos sobre este, se reserva un nuevo producto incrementado la cantidad de productos reservados. Vamos a suponer que reservados dos veces el primer teléfono y una vez el primer portátil. El número de productos en carrito será de tres como se muestra a continuación.



A mayores, los diseñadores han de adaptarse a diferentes tamaños de pantalla siendo los siguientes: 850px, 675px, 600px, 400px. Las siguientes imágenes corresponden con el comportamiento de la vista para los anteriores tamaños de pantalla.





Tenga en cuenta que para el menú lateral se hace uso de la sección HTML con class: close-menu.

## Carrito

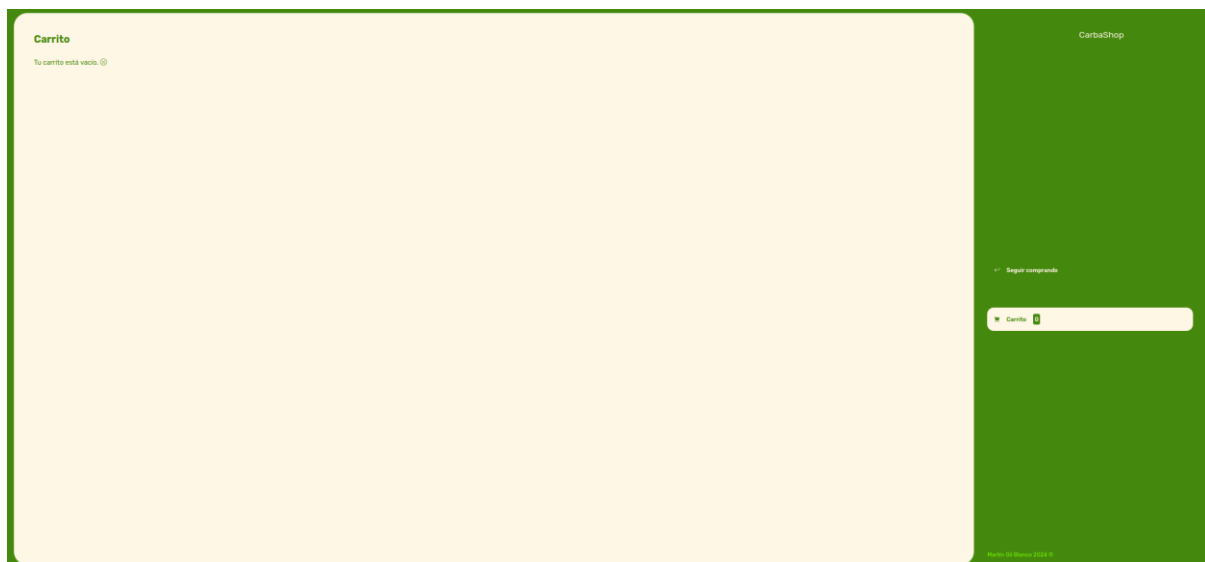
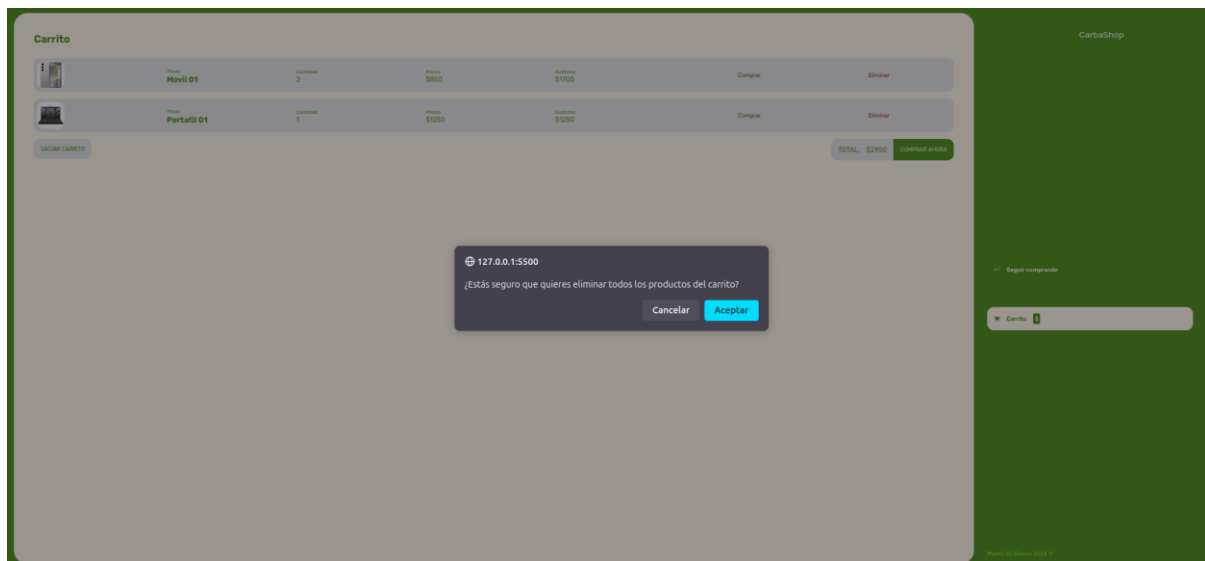
Una vez pulsamos en la cesta donde se nos indica que tenemos tres productos seleccionados nos aparecerá el histórico de los productos que hemos agregado. Para esto, podemos hacer uso del almacenamiento del navegador tanto almacenando en forma de string un objeto JSON que guarde los productos que tenemos agregados, así como el número de productos agregados.



Como podemos apreciar en esta vista vamos a tener lo siguiente.

- Una columna lateral a la derecha que nos permite ver la información del carrito o volver a la vista anterior de todos los productos.
- Una entrada por cada uno de los productos donde se aprecia:
  - Imagen.
  - Nombre.
  - Cantidad (número de productos que hemos agregado al carrito).
  - Precio del producto.
  - Subtotal (resultado de multiplicar la cantidad de productos por el precio para cada tipo de productos).
  - Botón de comprar.
  - Botón de eliminar.
- Botón de “Vaciar Carrito”.
- Entrada con precio total y botón de “Comprar ahora”.

Con el objetivo de no tener que implementar una lógica de negocio, los bototones, únicamente simularán el comportamiento, es decir, si nosotros pulsamos en cualquiera de ellos tendremos un mensaje de alerta de la acción a realizar. Supongamos que pulsamos en “Vaciar Carrito”.

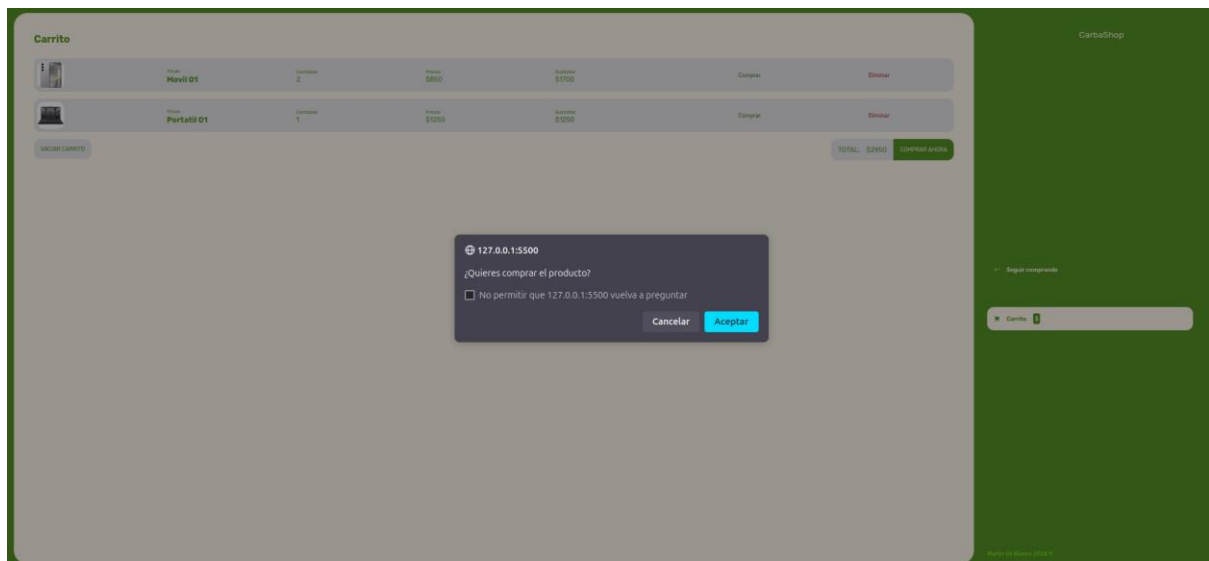


En caso de que no haya ningún producto o de que se hayan eliminado o comprado todos los productos el número de productos ha de ser cero y aparecerá el mensaje “Tu carrito está vacío :(”.

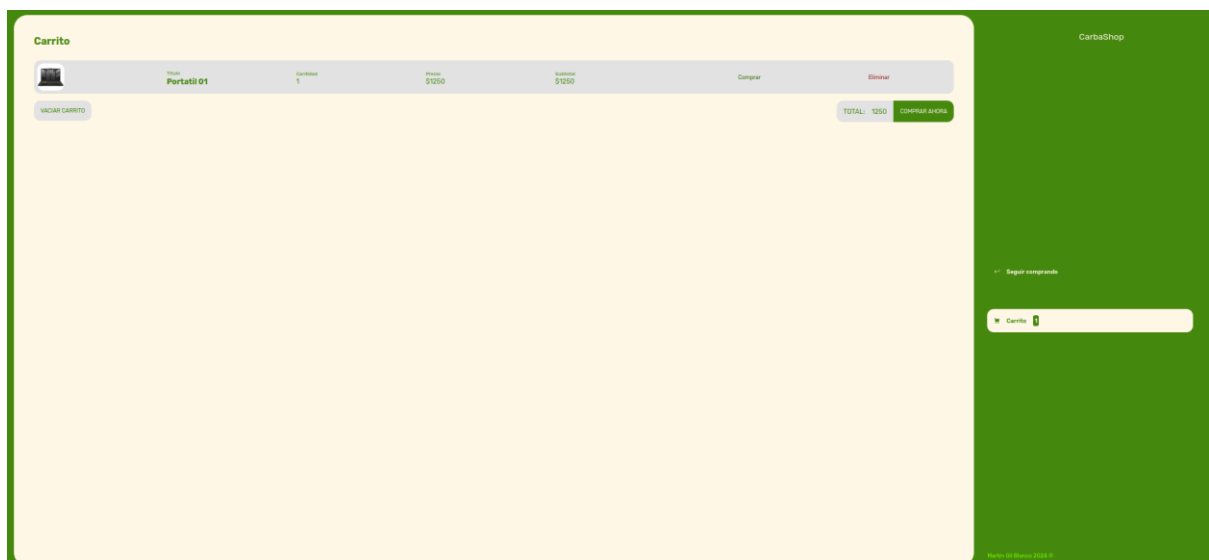
Del mismo modo que con el botón de “Vaciar carrito”, el botón “Comprar ahora” realizará el mismo comportamiento, pero tendremos el mensaje de alerta “¿Estás seguro que quieres comprar todos los productos por un total de \$2950?”.

Por otra parte, el usuario podría hacer uso de los botones de comprar o eliminar, supongamos que decide comprar los dos primeros teléfonos, en este caso, el número de productos se reducirá de tres a uno, así como el total que pasará de 2950 a 1250\$.

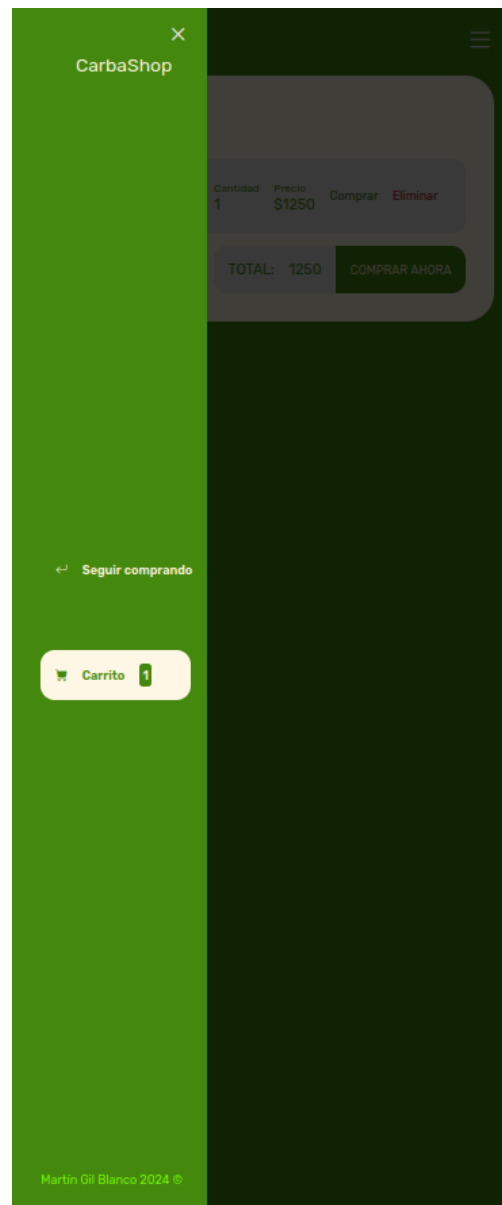




En caso de pulsar en el botón eliminar el mensaje sería: "¿Quieres eliminar el producto?".



Nuevamente, del mismo modo que antes la vista ha de poder adaptarse a los diferentes tamaños de pantalla, presentando un formato similar al siguiente.



Si el usuario pulsase en el nombre “CarbaShop” volveríamos a la vista de login, siendo necesario que cuando volvamos a entrar tengamos el carrito a cero. Esto es así ya que evitamos crear un servicio en base de datos donde guardemos usuarios y carrito asociado, a modo de únicamente centrarnos en el comportamiento de la interfaz y simplificar el desarrollo.

## Consideraciones finales

- El código del proyecto tendrá que estar subido en un repositorio único, siendo necesario realizar commits periódicamente, no se admitirá ningún proyecto que tenga menos de cinco commits y estos tendrán que realizarse en fechas diferentes. En caso contrario la calificación será un cero, con independencia del material entregado.
- Se hará uso de [github pages](#) para publicar la interfaz desarrollada.
- A parte de todo el código en formato ZIP con el formato. APELLIDO1\_APELLIDO2\_NOMBRE\_EV1.zip se tendrán que aportar las URL del repositorio y el page publicado. Se recomienda poner como público el repositorio posteriormente a la fecha de entrega para evitar copias.
- En caso de detectarse similitudes entre los compañeros, ambos tendrán un 0 como nota del trabajo.