

****

**PROYECTO INTEGRADO**

**Desarrollo de Aplicaciones Web**

**Raulmarket**

INDICE

PLANTEAMIENTO Y BRIEFING DEL PROYECTO ……………………………………………….. 3

HERRAMIENTAS Y/O TECNOLOGÍAS ………………………………………………………………. 4

FUNCIONALIDAD INTERNA ……………………………………………………………………………. 6

DESARROLLO DEL PROYECTO Y MOCKUP ………………………………………………………. 9

CODIFICACIÓN Y EXPLOTACIÓN DE RAULMARKET ……………………………………….. 20

PASARELA DE PAGO …………………………………………………….………………………………. 22

CONCLUSIONES FINALES ……………………………………..……………………………………… 23

REFERENCIAS Y BIBLIOGRAFÍA …………………………………………………………………….. 24

**PLANTEAMIENTO Y BRIEFING DEL PROYECTO**

**Raulmarket** es una aplicación web que tiene el objetivo de mostrar una serie de productos (de todo tipo) de los cuales el cliente podrá ir eligiendo cada uno de ellos, almacenar en su carrito y posteriormente realizar la compra.

Dispone de una interfaz principal que pedirá introducir una cuenta que debe estar registrada en el sistema.

Aunque también nos ofrece la posibilidad de registrarnos en ella (siempre otorgando el rol de usuario normal).

Esta aplicación se compone de dos tipos de roles de usuario:

- **Usuario**, es el rol que se le otorgará al usuario predeterminado, que tendrá los permisos justos y necesarios para ver, almacenar en el carrito y realizar la compra de los productos.

- **Administrador**, es el rol que tienen ciertos usuarios privilegiados.

Podrán ver, añadir, editar y eliminar cada uno de los productos.

Además de poder añadir otro usuario al sistema (también con el rol de administrador) y eliminarlos.

**HERRAMIENTAS Y/O TECNOLOGÍAS**

Para la realización de este proyecto se han requerido varias tecnologías para las distintas partes de la aplicación.

**DISEÑO**

**CSS**: «hojas de estilo en cascada» (Cascading Style Sheets). Lenguaje que maneja el diseño y presentación de las páginas web.

**Bootstrap**: Framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Su principal objetivo es facilitar el proceso de desarrollo de los sitios web responsivos y orientados a los dispositivos móviles.

**FUNCIONALIDAD**

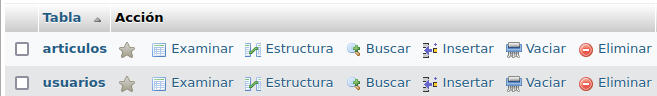
**Javascript:** Lenguaje de programación que convierte nuestra aplicación en una web más dinámica.

**Ajax:** Tecnología utilizada para crear páginas web asíncronas.

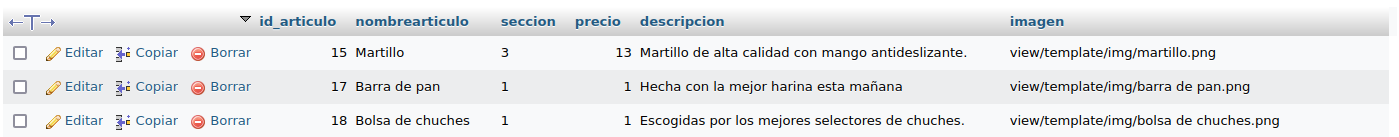
**Php:** Lenguaje de programación interpretado del lado del servidor.

**BASE DE DATOS**

**PhpMyAdmin**: Aplicación web que sirve para administrar bases de datos.

En **Raulmarket** no se utiliza una base de datos muy compleja, realmente se utilizan nada más que dos tablas:

En la tabla de **artículos** tendremos varias columnas, entre ellos el id\_articulo (código único que hará referencia a ese artículo en concreto), su nombre, la sección a la que pertenece, precio, descripción y la ruta de la imagen (almacenada en una carpeta).

****

****En la tabla de **usuarios**, podemos observar que tenemos ese id único para cada usuario, el rol (puede ser administrador o usuario), nombre, nombre de usuario y la contraseña encriptada (con MD5).

**FUNCIONALIDAD INTERNA**

Este proyecto sigue el patrón MVC (Modelo Vista Controlador), la idea de este patrón es separar nuestro sistema en tres capas.

**MODELO**

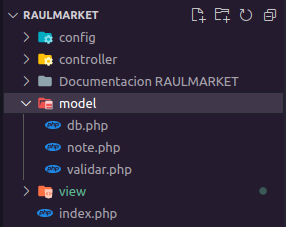
Se encarga de todo lo que tenga que ver con la persistencia e interacción con la base de datos.

En este caso, tenemos tres modelos:

**-db.php**: Contiene todos los datos relacionados con la interacción con la base de datos.

**- note.php**: Contiene todas las funciones de búsqueda, guardado, creación, etc, de los productos.

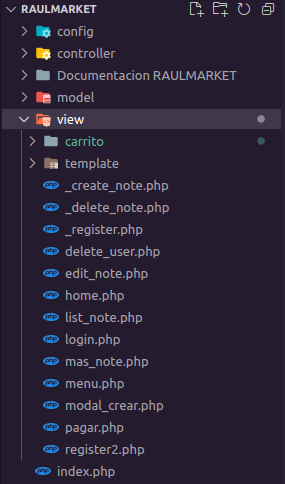
**- validar.php**: Contiene todas las funciones de gestión con los usuarios del sistema.



**VISTA**

Muestra al usuario todo lo que se pueda visualizar.

En este caso, disponemos de muchas vistas para las distintas funcionalidades de nuestra aplicación. Incluido las vistas que ya no utilizamos (tienen un “\_” delante).



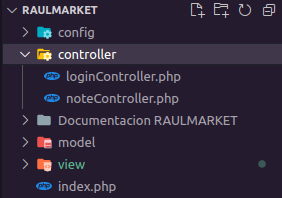
**CONTROLADOR**

Pide información al modelo y la implanta sobre la vista.

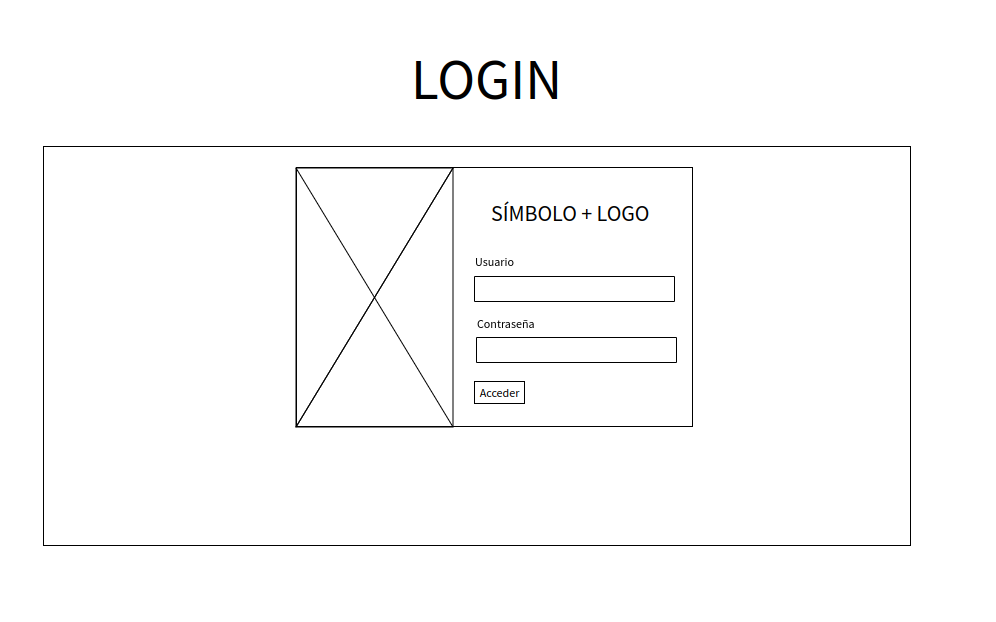
En este caso tenemos dos controladores:

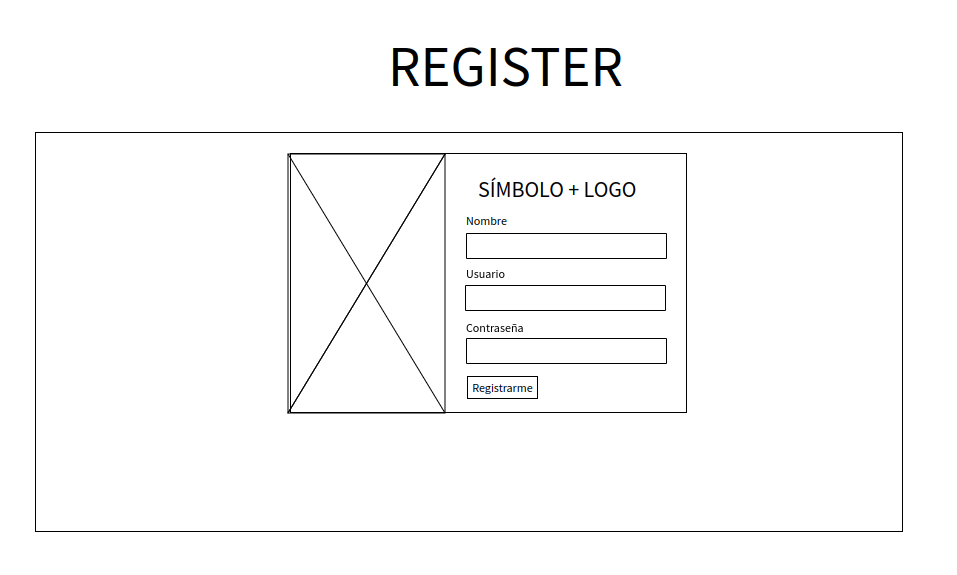
**- loginController.php**: Controla todo lo relacionado con la gestión de usuarios del login y register.

**- noteController.php**: Controla todo lo relacionado con la gestión de productos del sistema.



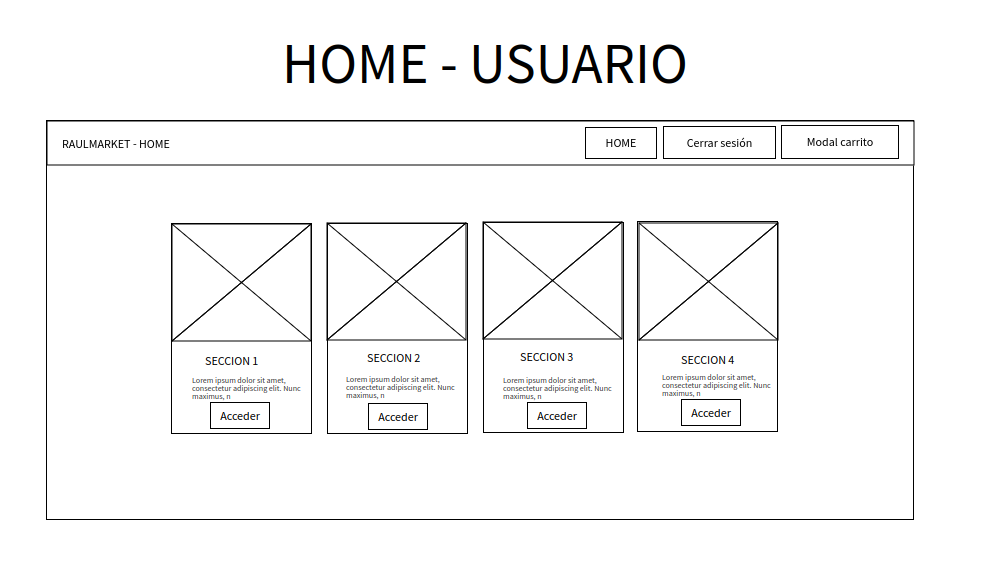
**DESARROLLO DEL PROYECTO Y MOCKUP**

****

****

La vista (login.php) muestra dos tipos de formularios, el de loguearse y la de registrarse.

**VISIÓN DEL USUARIO**

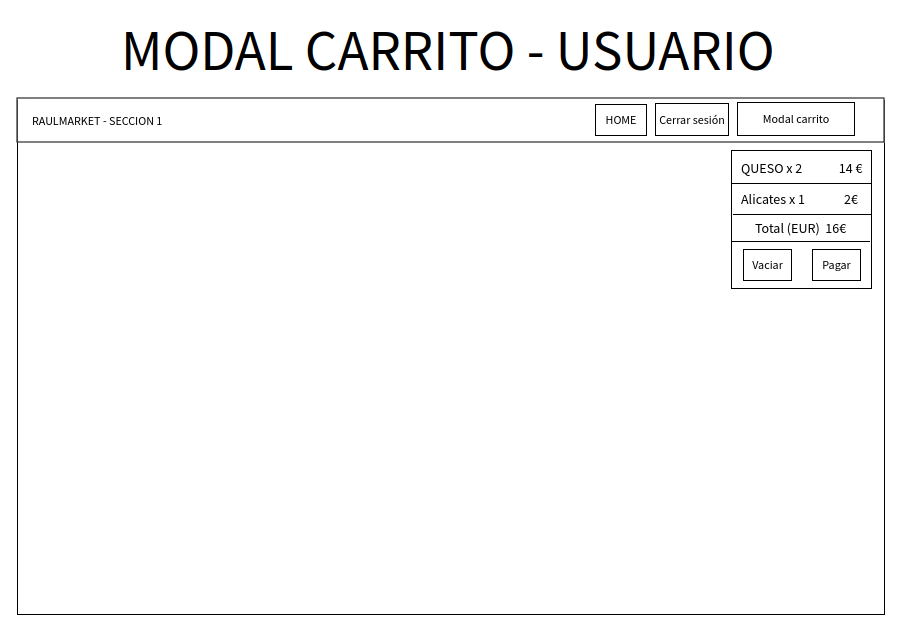
****

Una vez accedido, el usuario contará con un menú sencillo con varios botones y se situará en el **Home**:

- **Home**: Al hacer click redirige hacia la interfaz principal donde se muestras las distintas secciones de productos.

- **Cerrar** **sesión**: Botón que cierra la sesión del usuario y dirige a login.php.

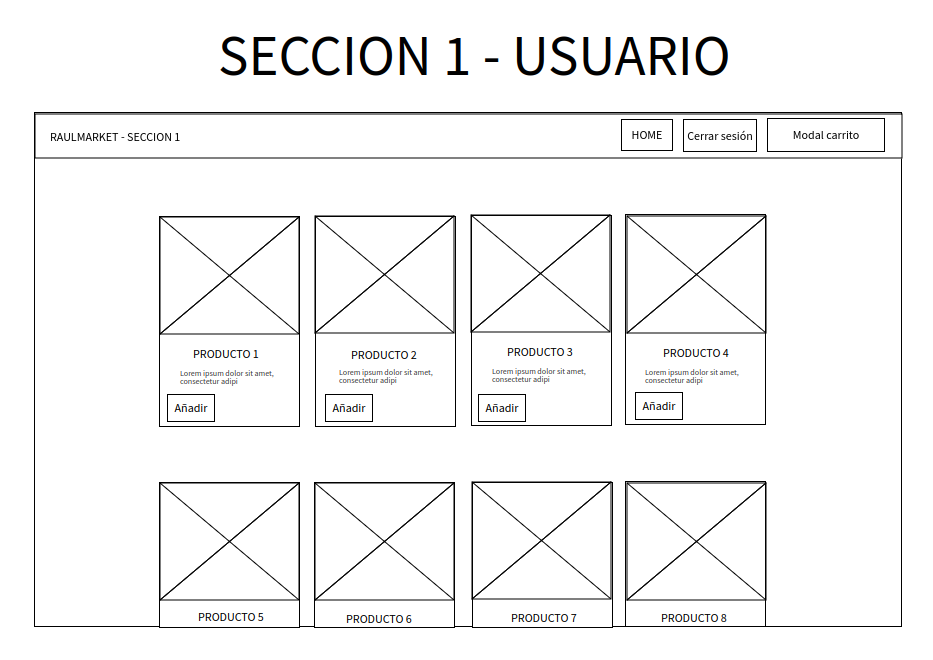
**- Modal** **carrito**: Modal que se actualizará automáticamente según se agreguen productos al carrito, mostrando también el total.

****

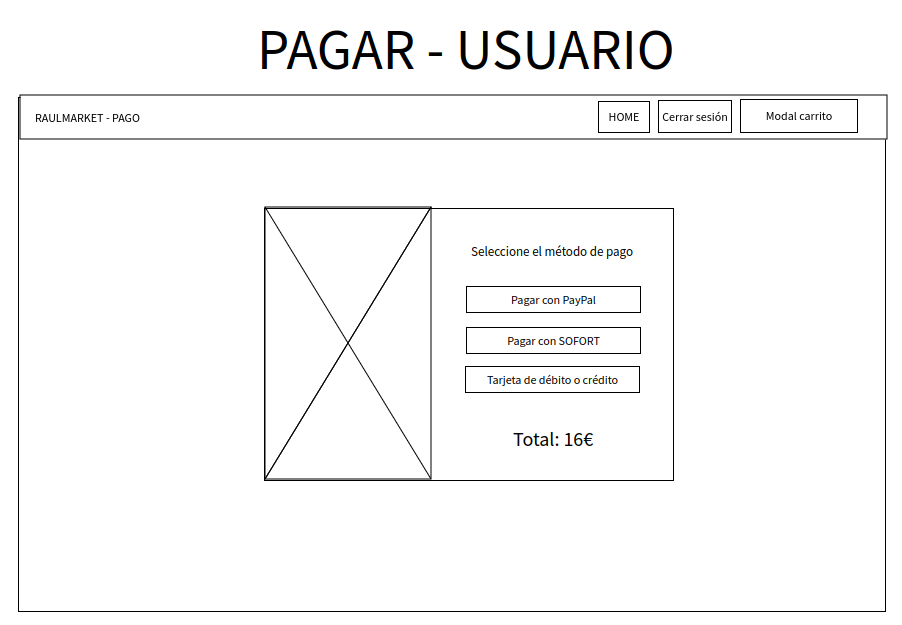
El modal del carrito muestra una lista de los productos que se van añadiendo, el recuento total de todos ellos y además dos botones:

- **Vaciar** **carrito**: Al hacer click se vaciará por completo el carrito.

- **Pagar**: Redirige a la página de pago donde se realiza el mismo.



Al acceder a cualquier sección se mostrarán los distintos tipos de productos con la opción de poder añadirlos al carrito.



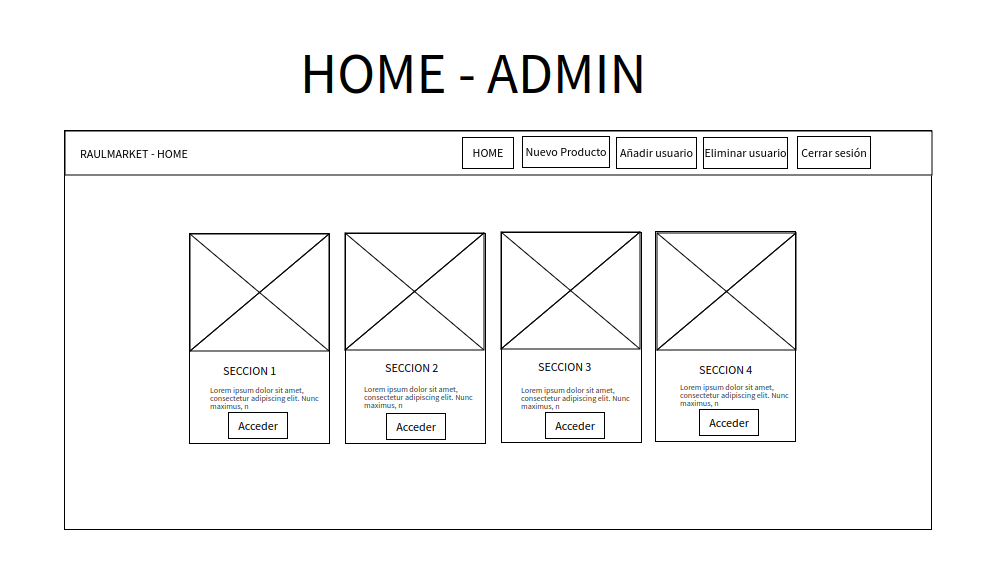
Vista (pagar.php) donde muestra al usuario el total a pagar con las distintas opciones de pago disponibles, todas ellas desarrollado por PayPal:

- PayPal

- SOFORT

- Tarjeta de débito o crédito

**VISIÓN DEL ADMIN**

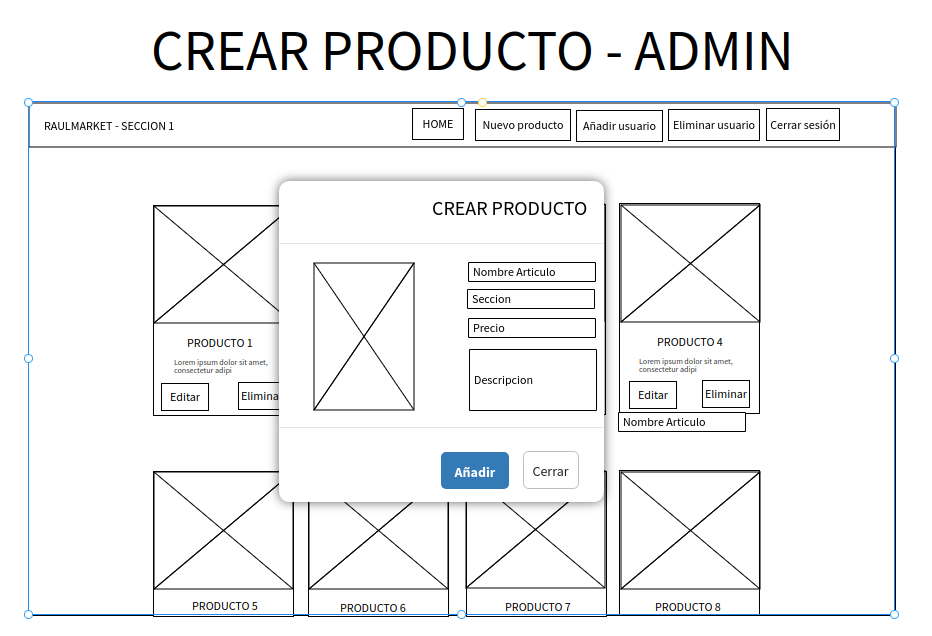
****

Una vez accedido, el admin contará con un menú más complejo que el usuario normal, dispondrá de las opciones mencionadas anteriormente (menos el carrito) junto a otras tres más y se situará en el **Home**:

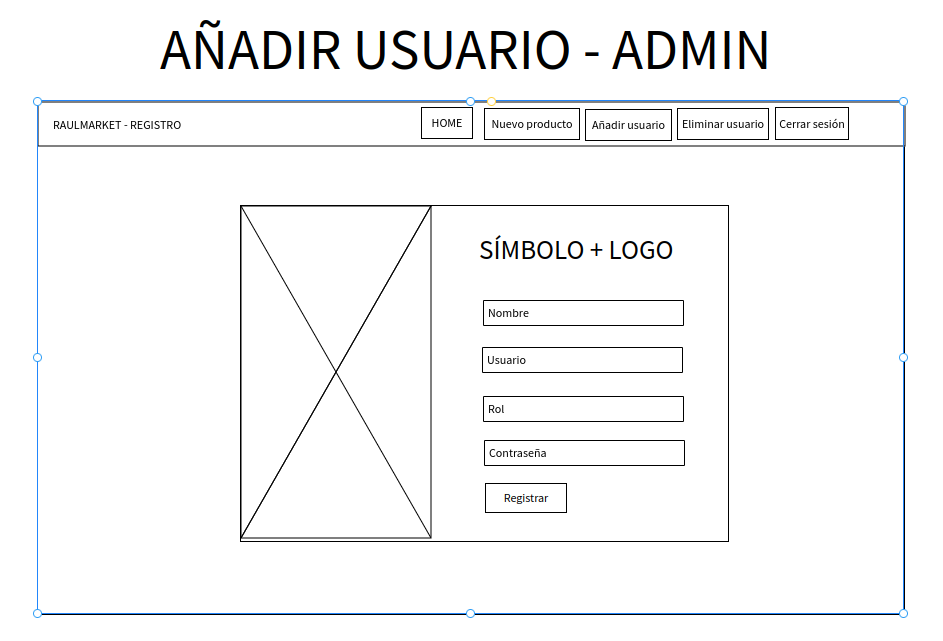
**- Crear producto**: Abre un modal con un formulario que añade productos.

**- Añadir usuario:** Redirige a “agregar2.php” que permite agregar usuarios con cualquier tipo de rol.

**- Eliminar usuario**: Redirige a “delete\_user.php”, donde muestra una tabla con todos los usuarios con la posibilidad de eliminarlos.

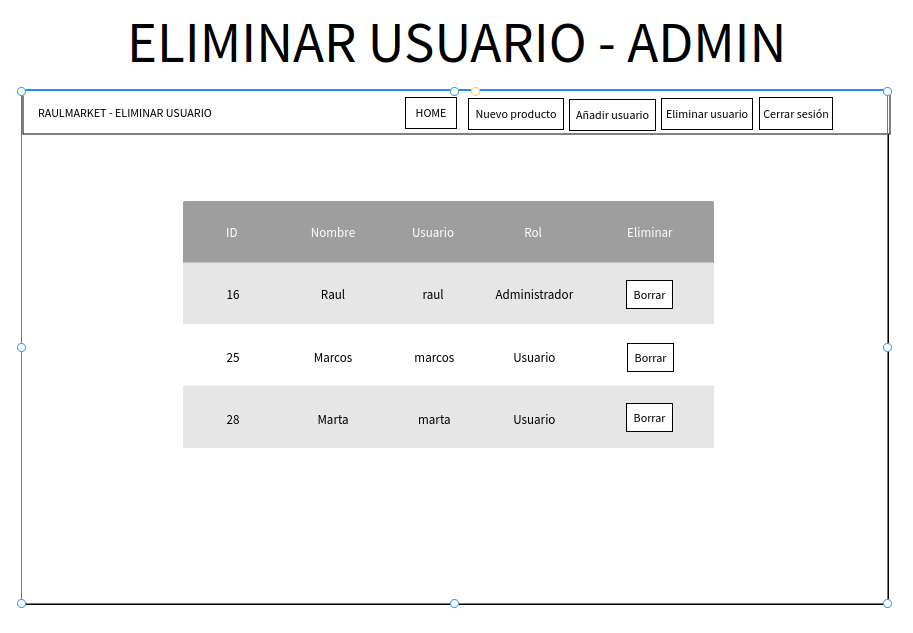


En la opción del menú de “Nuevo producto” nos muestra un modal que contiene un formulario con los datos del producto a incluir, entre ellos la sección, que muestra una lista de las secciones disponibles y donde añadir el producto.

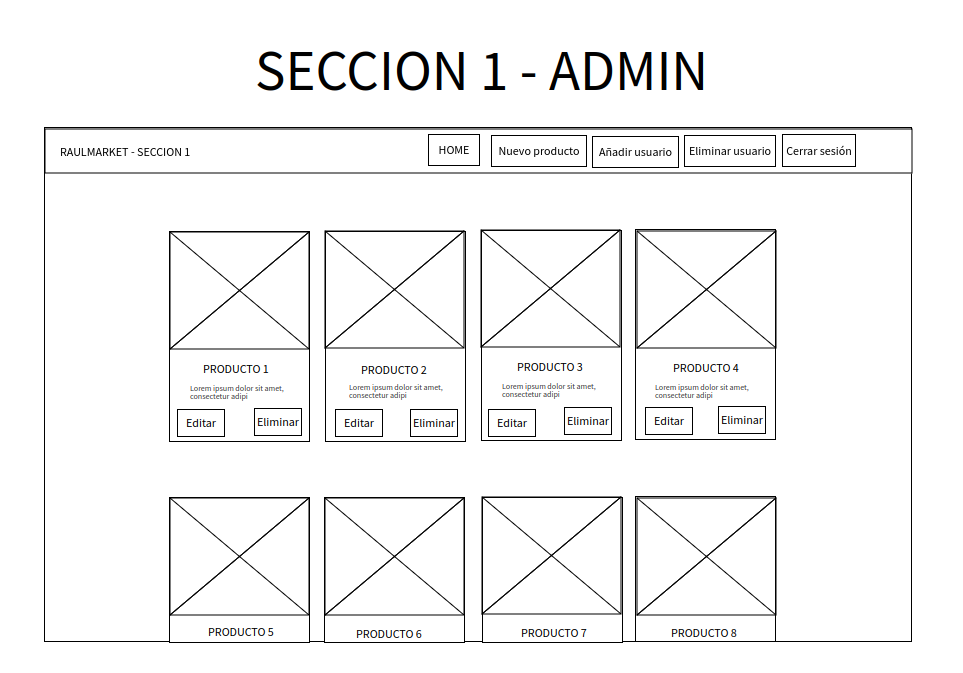


En la opción del menú de “Añadir usuario”, muestra un formulario con los datos a rellenar de ese usuario, entre ellos destaca la casilla de “Rol”, que ofrece la posibilidad de que el usuario sea tanto “**usuario**” como “**administrador**”.

Por lo que un usuario administrador solamente lo podrá ser gracias a otro administrador.



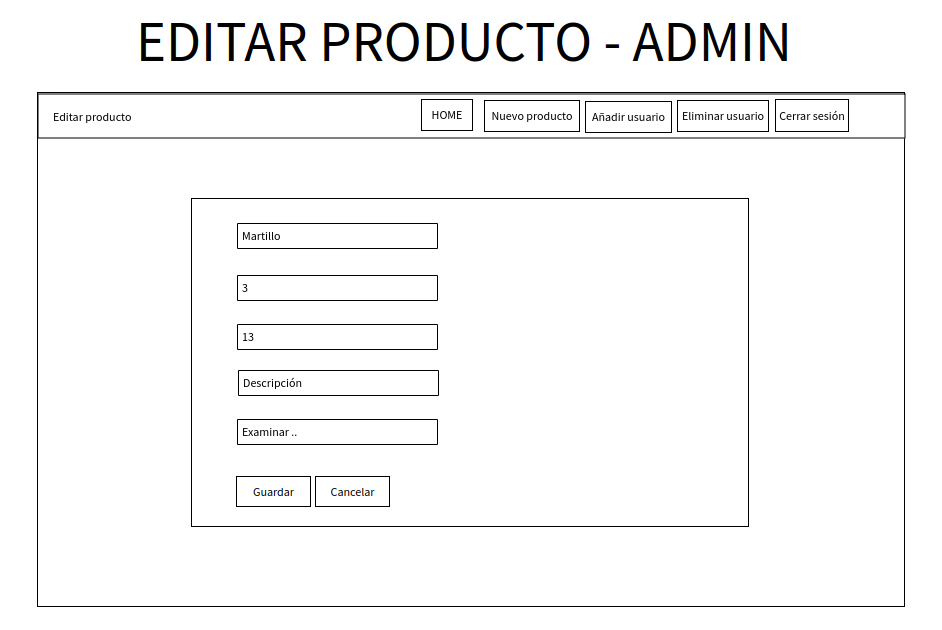
En la opción del menú de “Eliminar usuario”, muestra una tabla con todos los usuarios del sistema, ofreciendo a través de un botón el borrado de ese usuario.



En la sección ubicada, el administrador tendrá toda la gestión sobre esos productos:

- **Editar**: Redirige a “edit\_note.php”, donde nos muestra un formulario (con los datos rellenados de ese producto a editar) y ofreciendo la posibilidad de cambiarlos.

- **Eliminar**: A través de la tecnología AJAX, nos elimina ese producto de forma inmediata y mostrando el cambio realizado sin la necesidad de tener que recargar la página.



Al pulsar “editar” nos redirige a “edit\_note.php”, donde nos muestra un formulario con los datos rellenados de ese producto y con posibilidad de añadir una imagen.

**CODIFICACIÓN Y EXPLOTACIÓN DE RAULMARKET**

Como se ha explicado anteriormente, utilizamos el protocolo MVC y esto da pie a que nuestro archivo “**index.php**” actúe como emisor y receptor del controlador.

Este archivo ejecuta datos por defecto (página de login) y a raíz de aquí, recibirá otros datos que ejecutarán las distintas funciones del controlador junto a sus vistas.

**Controlador**

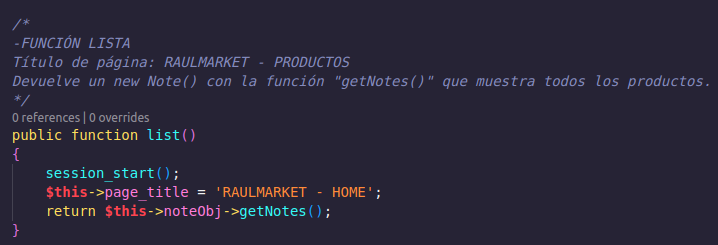
Un ejemplo podría ser la función “list()” de “noteController.php”.

Se puede observar como:

- Continuamos la sesión con “session\_start()”.

- Establece el título de la página.

- Devuelve el resultado de la función “getNotes()” (ubicada en el modelo note.php).

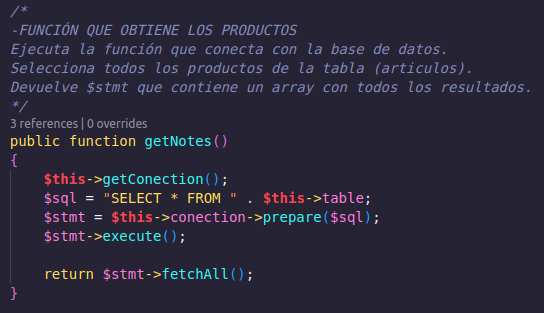


**Modelo**

Podemos ver que la función “getNotes()” realiza la interacción con la base de datos.

- Ejecuta otra función (la cual es el conector con nuestra base de datos).

- Guarda en una variable la sentencia a realizar (en este caso mostrar todos los datos).



- Prepara la sentencia y se ejecuta.

- Devuelve un array con todas las filas de la búsqueda realizada.

**PASARELA DE PAGO**

Para la realización de la pasarela de pago me he apoyado en el kit de herramientas (SDK) que ofrece el propio PayPal.

Primeramente se incluye el script :

Después, utilizamos el id “paypal-button-conteiner”, que nos crea la sección con las distintas opciones de pago que nos ofrece PayPal.

Y lo demás consiste en darle su funcionalidad y estilo.



**CONCLUSIONES FINALES**

Para la realización del proyecto tenía pensado en un principio extender y mejorar “Raulapp”, el proyecto realizado el curso anterior.

Sobretodo por el hecho de haber obtenido algo de experiencia en Laravel y Tailwind CSS.

Pero debido a la forma de trabajar en mi empresa (Área 14), usando código de forma nativa y usando el protocolo MVC, decidí hacerlo de esta manera y afrontar un nuevo proyecto por completo.

En relación a la funcionalidad, he incluido multitud de tecnologías y herramientas utilizadas hoy en día, como pueden ser el uso de modales, tecnología Ajax (para definir comunicaciones con el servidor sin recargar páginas), uso de roles de usuario, carrito de compra, o incluso la utilización de un SDK para la realización del pago por parte de un cliente.

Optando también por una interfaz intuitiva y visual, he usado Bootstrap, como framework (el más utilizado en la actualidad).

Por último, he perfeccionando el diseño para que llegue a ser “responsive”, de modo que sería apto para entorno móvil y Tablet.

**REFERENCIAS Y BIBLIOGRAFÍA**

- <https://www.jairogarciarincon.com/clase/programacion-orientada-a-objetos-en-php/patron-mvc>

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

- [https://www.hostinger.es/](https://www.hostinger.es/tutoriales/que-es-ajax)

- <https://stackoverflow.com/>

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

- <https://www.lawebdelprogramador.com/foros/JavaScript/>

- <https://getbootstrap.com/docs/4.0/utilities/>

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

- <https://lenguajejs.com/javascript/peticiones-http/ajax/>