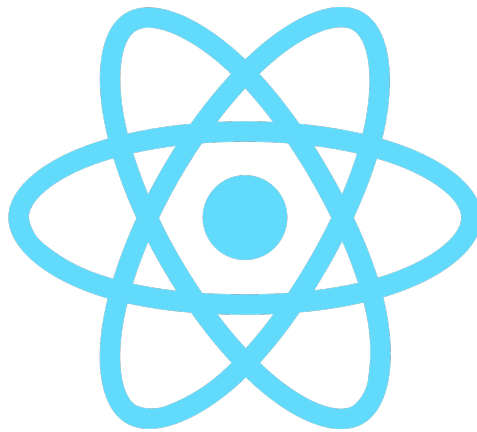




Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Universidad Pública de Navarra

DESPLIEGUE DE SERVICIOS MULTIMEDIA



Memoria de proyecto

REACT WEB APP

Iñigo Aranguren

Marzo 2020

Índice general

1. Estructura del proyecto	2
2. Componentes de la Tienda	3
3. Componentes de los Pedidos	4

1. Estructura del proyecto

La aplicación esta compuesta por un *Router* principal el cual permite llevar a cabo el enrutado a través de las diferentes páginas que componen la aplicación. Este componente *Router* se sitúa como la raíz del árbol de componentes puesto que será el que controle toda la aplicación.

Cabe destacar que el componente **App** principal controlará además de las posibles rutas de la aplicación la autenticación de los usuarios. Al cargarse el componente comprobará si existe en el *local storage* del navegador el token del usuario para cargar la sesión del mismo. En este componente se creará un contexto para que todos los componentes de la aplicación puedan acceder a él y tengan información sobre diferentes aspectos como por ejemplo la autenticación.

Dados los requisitos del proyecto, necesitábamos realizar 2 páginas principales en la aplicación: la página de los productos de la tienda y la página para revisar los pedidos ya realizados. Me referiré a estos dos componentes como *containers* puesto que son componentes que únicamente renderizan otros componentes más sencillos para formar la página que serán comentados a continuación. Adicionalmente, para llevar a cabo el sistema de autenticación se habrán creado dos *containers* auxiliares que serán el **Login** y **SignUp** para el inicio de sesión y registro respectivamente.

La página constará de 2 rutas principales: la ruta .../tienda en la que se encontrará el *container* **Tienda** y la ruta .../pedidos en la que se encuentra el *container* **Pedidos**.

Además, existirá un componente común a ambas páginas de la web que será el componente **MyNav**. Este componente será la barra de navegación superior que permitirá moverse por las secciones de la web. Será un *Nav* normal y corriente de *Bootstrap* pero con el añadido de los componentes *Link* necesarios para realizar el enrutado a través de *react-router*.

En las siguientes secciones se comentarán los componentes principales utilizados en cada uno de los *containers* mencionados.

2. Componentes de la Tienda

El componente **Tienda** carga los productos desde la base de datos de Firebase en su método *component-DidMount* y los almacena en el estado. Estos productos serán utilizados en el componente **ProductList** el cual formará una lista de otro componente creado llamado **Product**.

El componente **ProductList** recibe como propiedades un array de productos cargado desde la BBDD, un array que llevará la cuenta de los productos añadidos al carrito así como las funciones que serán llamadas por los botones + y - de cada producto. Este componente renderiza una lista(*list-group*) de componentes **Product**.

El componente **Product** recibe como propiedades los atributos del mismo guardados en la BBDD(imagen, precio, nombre ...) así como el número del mismo que ha sido añadido al carrito junto con las funciones de los botones - y +.

Además de la lista de productos, en el componente **Tienda** habrá otros componentes ocultos como los *Snackbar* y los *Modals*. Estos componentes se harán visibles cuando el usuario lleve a cabo ciertas acciones en esta página.

Por ejemplo los *Snackbar* serán visibles unos segundos para informar al usuario de productos añadidos o eliminados del carrito. Su visibilidad estará controlada a través del estado del componente principal **Tienda**.

Por otra parte, los *Modals* serán visibles cuando el usuario vaya a realizar el pedido. Habrá 3 *Modals* dados los requisitos del proyecto. Su visibilidad también será controlada a través del estado. Será en uno de estos *Modals* (**FormModal**) en el que se grabe el pedido en la BBDD con ciertos campos adicionales del usuario.

Cabe destacar que si el usuario no se encuentra autenticado y procede a realizar el pedido, será redirigido a la página de login para realizar el inicio de sesión correspondiente.

3. Componentes de los Pedidos

De igual forma que en el anterior, el componente **Pedidos** carga los pedidos almacenados en Firebase en su método *componentDidMount*. Estos pedidos se pasarán como propiedad del componente **PedidosList**. Los pedidos serán almacenados en Firebase con la siguiente estructura:

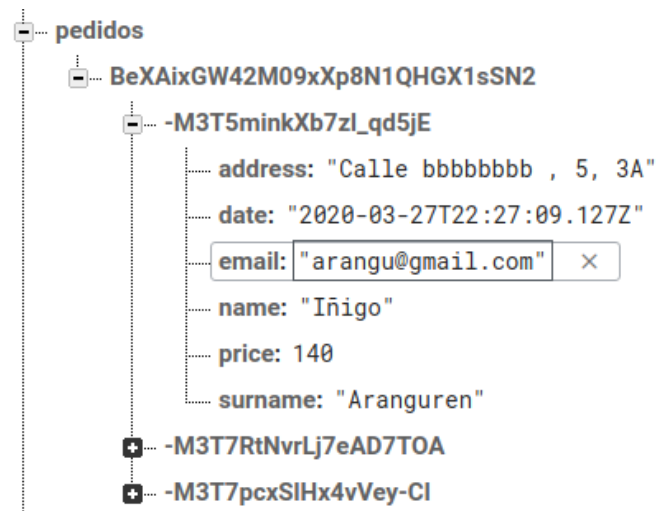


Figura 1: Estructura de los pedidos en la base de datos.

La rama de pedidos se dividirá en árboles inferiores que irán segmentados por el UID de Firebase. Por lo tanto, todos los pedidos que realice un usuario irán dentro de la colección de su UID y serán independientes a los realizados por otros usuarios. En la figura 1 se aprecia que el usuario con UID *BeXAixGW42M09xXp8N1QHGX1sSN2* ha realizado 3 pedidos que tendrán los IDs autogenerador por Firebase, y dentro de ellos se encontrará la información recogida desde el formulario implementado en la **Tienda**.

El componente **PedidosList** recibe como propiedad además del array de pedidos la función que implementará el botón de eliminar pedido. Este renderizará un componente *Accordion* propio *Bootstrap* al cual le pasará una lista de componentes **Pedido**.

El componente **Pedido** recibe como propiedades los atributos guardados del pedido en Firebase(nombre, apellido, precio, dirección, ...) así como la función que será utilizada para borrar el pedido de la BBDD.

En la página de pedidos también se han hecho uso de componentes invisibles como un *Snackbar* que aparecerá cuando el usuario elimine un pedido así como un *Modal* que permitirá confirmar al usuario la acción de eliminar un pedido.

Esta página no será accesible si el usuario no ha sido autenticado, puesto que serán los pedidos realizados por el usuario autenticado los mostrados en la sección.