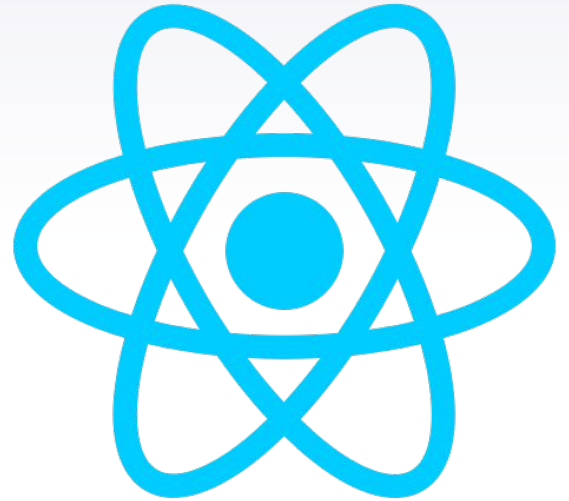




Consigna final de FrontEnd III



Propuesta 1

Carrito de compras

Crearemos un simulador de carrito de compras, en donde se debería experimentar la realización de compra de algún tipo de ítem.

En este proyecto se debería poder:

- Ver una lista de items
- Ver el detalle de cada uno de los items, cuando lo selecciono (Bonus)
- Poder guardar los items que deseo comprar en un carrito de compras
- Renderizar los items que quiero comprar dentro del carrito, a modo de lista
- Finalizar la compra, mostrando el total y los items comprados en un alert.

Para esto vamos a necesitar de las siguientes librerías

- React-router-dom
- axios

Propuesta 1

Como vamos a organizar nuestro proyecto

- Crear un sistema de rutas en donde contengamos:
 - una página de Inicio,
 - una lista de productos,
 - detalle de producto seleccionado (Bonus)
 - un carrito en donde muestren los productos seleccionados

Propuesta 1

En Inicio

- Debemos tener una NavBar o Header en donde nos debería dejar acceder a las diferentes rutas que creamos dentro del proyecto. Se pueden colocar los títulos de cada ruta e imprimirlos directamente en el Header, o se puede realizar una hamburguesa, la cual, al ser clickeada despliega los Links de redirección para cada ruta.
- Debajo del Header podemos colocar una portada de bienvenida y algún diseño que se nos ocurra que tenga que ver con la temática del shop.

Propuesta 1

En Lista de productos

- Debemos hacer un llamado a un endpoint de Mercado libre. Guardamos los resultados de esta endpoint dentro de un estado y renderizamos una Card por cada producto que traemos. (Tal como en el entregable)
- En cada Card, debería renderizarse imagen, título y precio del producto, sumado a dos botones de 'Agregar al carrito' y de 'Ver detalle'.
- Al clickear, agregar al carrito, debería guardarse el producto, dentro de la variable de carrito e ir sumando al total y a la cantidad de productos seleccionados.
- El stock de cada producto puede ser uno por producto o el stock que trae desde la api o bien, un número a decisión de ustedes.
- Dato importante: No confundir el componente Card con el componente del Detalle de cada Producto.

Propuesta 1

Para los productos vamos a necesitar

- Hacer un llamado a la API de Mercado libre:

```
'https://api.mercadolibre.com/sites/MLA/search?q=productos'
```

En donde "productos" serían los productos que vamos a vender en nuestra aplicación. Por ejemplo, si vamos a vender zapatillas, reemplazamos la palabra productos en la endpoint por zapatillas y la API nos traería como resultado una lista de zapatillas.

Hagamos de cuenta que es como si escribiéramos en el buscador de la página oficial de mercado libre y según lo que busquemos, nos va a dar una lista con esos productos que estamos buscando.

Propuesta 1

En Detalle del producto (Bonus)

- Podemos acceder a esta ruta, luego de haber clickeado “Ver detalle” en algún producto. Acá vamos a tener que usar `useParams`, para poder dirigirnos a la ruta en base al producto seleccionado.
- En este componente, debemos renderizar título, imagen y precio del producto y algunas características más, dependiendo de qué producto sea (opcional). Se puede extraer de “atributes” dentro de cada objeto que traemos.
- Acá también debería figurar un acumulador de stock para comprar en caso de que sea más de un producto disponible y un botón para agregar al carrito.

Propuesta 1

En Carrito

- Se debería renderizar la lista de los productos seleccionados y el total de precio a pagar.
- Finalmente, colocaremos un botón para finalizar compra, en donde debería darnos un alert o Sweetalert de los productos comprados, o simplemente un mensaje para el usuario en donde nos diga que finalizamos nuestra compra.

Propuesta 1

Cómo manejar los ítems seleccionados y el total

- Para esto vamos a necesitar un Context en donde encontraremos varios tipos de datos:
 - Un array a modo de carrito en donde guardaremos los items seleccionados para comprar
 - Un estado el cual cumplirá la función de acumular el total de precio a pagar para finalizar la compra.
 - Un loading global para usarlo en los momentos de carga de cada componente o página.
 - Metodos/Funciones para agregar items al carrito, sumar al total y borrar items que nos arrepentimos de comprar (Bonus)

Propuesta 1

A tener en cuenta

- Hagamos lo posible de todo lo propuesto, si algún componente no funciona, dejemoslo planteado como está y se irá revisando para poder arreglarlo.
- Recomendación: dejar los Bonus para el final de toda la consigna.
- En caso de querer utilizar alguna librería extra para resolver alguna parte de la consigna, bienvenida sea. (Ya sea Bootstrap, Material UI, Sweetalert, FontAwesome, Toastify)
- Tratemos de equilibrar el tiempo que le dedicamos al diseño y a lo funcional. Pensemos que lo más importante es que nuestro proyecto funcione, mientras que el diseño, con tener lo básico ya tendríamos un proyecto hecho.



Cualquier consulta nos
mantenemos al tanto!