Header
El componente Header proporciona la estructura y la apariencia visual del encabezado de la página. Incl uye el logotipo de Mercado Libre y el componente SearchBox para permitir búsquedas rápidas en la aplic ación.

Search box
El componente SearchBox proporciona una interfaz de búsqueda interactiva que permite a los usuarios in gresar un término de búsqueda y realizar una navegación a una ruta específica con ese término de búsq ueda como parámetro. Utilizando los hooks useState y useNavigate, se gestiona el estado del término de búsqueda y se realiza la navegación de manera eficiente

navigationContent
El componente NavigationContent se encarga de definir y renderizar las rutas de la aplicación utilizando r eact-router-dom. Dependiendo de la ruta actual, se renderiza el componente correspondiente, como Sear chView o Detail, con los parámetros adecuados.

ItemList
El componente ItemsList es responsable de renderizar una lista de artículos en la interfaz de usuario. Utili za la prop items para obtener los datos de los artículos y muestra solo los primeros cuatro elementos. Ca da elemento se renderiza utilizando el componente <Item>, que se importa desde el archivo "./Item".

Item
El componente Item se utiliza para mostrar la información de un artículo en la interfaz de usuario. Renderi za la imagen en miniatura, el precio, el indicador de envío gratuito, el título y el origen del artículo. Se utili zan componentes como <Link> de react-router-dom para crear enlaces a las páginas de detalle del artícu lo.

ItemDetail
El componente ItemDetail se utiliza para mostrar los detalles de un artículo, incluyendo la imagen, el títul o, la condición, el precio, el botón de compra y la descripción

Detail
El componente Detail utiliza el hook useParams de react-router-dom para obtener el parámetro id de la U RL, que corresponde al ID del artículo.

El estado item se inicializa como un objeto vacío utilizando el hook useState. Este estado se utilizará par a almacenar los datos del artículo.

El hook useEffect se utiliza para realizar una solicitud HTTP a la API de Mercado Libre cuando el ID del a rtículo cambia. Dentro del useEffect, se realiza la solicitud a la URL correspondiente al artículo utilizando el ID obtenido de useParams. Los datos de respuesta se convierten a JSON y se actualiza el estado item con los datos del artículo si no hay errores.

En caso de producirse algún error durante la solicitud, se muestra un mensaje de error en la consola.

El componente Detail retorna null si el estado item está vacío utilizando la función isEmpty de la bibliotec a lodash. Esto evita renderizar el componente ItemDetail antes de que se hayan obtenido los datos del ar tículo.

Si el estado item no está vacío, se renderiza el componente ItemDetail pasando el objeto item como prop .

Search

El estado items se inicializa como un arreglo vacío utilizando el hook useState. Este estado se utilizará p ara almacenar los resultados de la búsqueda.

El hook useEffect se utiliza para realizar una solicitud HTTP a la API de Mercado Libre cuando la prop se arch cambia. Dentro del useEffect, se realiza una solicitud a la URL correspondiente a la búsqueda utiliza ndo el término de búsqueda obtenido de la prop search. Los datos de respuesta se convierten a JSON y se actualiza el estado items con los resultados de la búsqueda si no hay errores.

En caso de producirse algún error durante la solicitud, se muestra un mensaje de error en la consola.

El componente Search retorna null si el estado items está vacío utilizando la función isEmpty de la bibliot eca lodash. Esto evita renderizar el componente ItemsList antes de que se hayan obtenido los resultados de la búsqueda.

Si el estado items no está vacío, se renderiza el componente ItemsList pasando el arreglo items como pr op.