**TO DO THE INVENTORY PROJECT**

**1. Estructura del Aplicativo**

* **Escaneo de Código de Barras:** Usa un lector de códigos de barras que envíe los datos al sistema. Al escanear, el sistema buscará en la base de datos y mostrará la información del artículo.
* **Consulta de Artículo:** Al escanear el código, el aplicativo mostrará las características del artículo, como nombre, descripción, oficina, y estado (si está registrado o no).
* **Historial de Escaneos:** Podrías agregar un historial para ver qué elementos han sido registrados y en qué momento.
* **Interfaz de Búsqueda Manual:** Para aquellos casos en los que el código no funcione o necesites una búsqueda específica.
* **Lector de Códigos de Barras**: Si tienes un lector físico, debería funcionar como una entrada de teclado. Si estás usando una cámara, podrías usar una librería de escaneo como **QuaggaJS** (JavaScript) o **ZBar** (Python).
* **PWA (Progressive Web App)**:
  + Si además de que se vea bien en el celular, quieres que tu aplicación web se sienta más como una aplicación nativa, puedes convertirla en una **PWA**. Esto te permite agregar funcionalidades como la posibilidad de instalar la aplicación en el teléfono, trabajar sin conexión a internet, y tener un rendimiento similar al de una aplicación móvil.
  + Las PWAs son aplicaciones web que pueden comportarse como aplicaciones nativas, pero no necesitas usar React Native para esto.

**4. Flujo de Funcionamiento**

1. **Escanear Código de Barras**: Al escanear, el lector envía el número al sistema.
2. **Consultar en la Base de Datos**: El sistema busca en la base de datos el artículo correspondiente al código.
3. **Mostrar Información del Artículo**: La interfaz muestra los detalles del artículo (nombre, descripción, oficina).
4. **Marcar como Registrado**: Cambia el estado del artículo a "registrado" en la base de datos y añade un registro al historial.
5. **Actualizar Pantalla**: Muestra un mensaje de éxito o de error según el resultado.

* **Componentes Principales**:
  + **Escáner**: Un componente para capturar el código de barras. Si usas una cámara, puedes integrar **QuaggaJS** o **html5-qrcode** para capturar y procesar el código.
  + **Vista de Artículo**: Muestra los detalles del artículo al escanear el código.
  + **Registro de Inventario**: Muestra una lista de artículos registrados y no registrados, con filtros para ver el historial.

**1. Captura y Envío de Datos al Escanear**

Cuando el lector de código de barras escanea un número:

* El lector simplemente coloca el número en el campo de entrada (como si lo escribieras en el teclado).
* Esto activa el evento onChange o onInput en el campo, que ejecuta la función de búsqueda y registro en el sistema.

**2. Búsqueda de Datos del Artículo y Registro Automático**

Una vez que se detecta el código de barras en el campo de entrada en tu componente BarcodeScanner, se hace lo siguiente:

* **Búsqueda de Artículo**: El código de barras se envía al servidor para obtener la información completa del artículo.
* **Registro Automático**: Si el artículo no está registrado, el servidor guarda automáticamente el estado del artículo como "registrado" y la fecha del registro.

1. **Referencia** (inputRef): La variable inputRef mantiene una referencia al campo de entrada. Esto te permite controlarlo directamente desde el código.
2. **Enfoque Automático**: En el useEffect, inputRef.current.focus() coloca el cursor en el campo de entrada cuando el componente carga por primera vez. Esto asegura que el campo esté listo para recibir el código al momento de escanear.
3. **Re-enfoque Después del Escaneo**: Cada vez que se completa el escaneo, el campo se limpia (setBarcode('')) y se vuelve a enfocar (inputRef.current.focus()), preparándose para el siguiente escaneo.