* Consulta realizada correctamente

¡Entendido! Con esta información, la forma de estructurar tu proyecto se vuelve más clara. Tu aplicación tendrá una **vista de lista**, una **vista de detalle** y una **función de exportación**. Aquí tienes una guía sobre cómo abordar cada parte, manteniéndolo simple con React y Vite.

**1. Estructura de Componentes**

Para replicar la interfaz que muestras, debes dividir la aplicación en componentes reutilizables y gestionar el estado de manera eficiente.

* **App.jsx**: Este será tu componente principal. Deberá gestionar el estado de la aplicación, como la lista de equipos y el equipo seleccionado para ver sus detalles. Usarás un **"estado" de React** (useState) para almacenar la lista de equipos y otro para controlar la vista actual (lista o detalle).

JavaScript

// Ejemplo de cómo App.jsx podría manejar el estado

const [equipos, setEquipos] = useState([]);

const [equipoSeleccionado, setEquipoSeleccionado] = useState(null); // null significa que se muestra la lista

* **ListaEquipos.jsx**: Este componente será tu vista principal, la que se parece a la foto. Su trabajo será:
  1. Recibir el array de equipos como una "prop".
  2. Renderizar una tabla con los encabezados de la foto (Device Name, Status, IP Address, etc.).
  3. Mapear sobre el array de equipos y, para cada uno, renderizar una fila de la tabla.
  4. Manejar el evento de clic en cada fila para que, al hacer clic, se muestre la información detallada del equipo.
* **DetalleEquipo.jsx**: Este componente se encargará de mostrar la información específica de un solo dispositivo.
  1. Recibirá el objeto del equipo seleccionado como una "prop".
  2. Renderizará los detalles del equipo en un formato fácil de leer (quizás una lista o una tarjeta).
  3. Incluirá un botón para regresar a la lista principal.

**2. Implementación de los Casos de Uso**

**Caso de Uso 1: Visualizar la lista de equipos**

Como vimos en la guía anterior, usarás un estado en App.jsx para almacenar la lista. Para que se parezca a la imagen, lo harás dentro de una tabla.

1. **En ListaEquipos.jsx**, crea la estructura HTML de la tabla (<table>, <thead>, <tbody>).
2. Dentro de <tbody>, usa el método map() para iterar sobre el array de equipos. Por cada equipo, crea una fila (<tr>) con las celdas (<td>) correspondientes.
3. A cada fila (<tr>), añádele un evento onClick que llame a una función en el componente padre (App.jsx) para cambiar el estado de equipoSeleccionado.

**Caso de Uso 2: Ver la información específica de un dispositivo**

Aquí usarás una renderización condicional. En el componente App.jsx, dependiendo del valor de equipoSeleccionado, renderizarás una vista u otra.

JavaScript

// En App.jsx

if (equipoSeleccionado) {

return <DetalleEquipo equipo={equipoSeleccionado} onBack={() => setEquipoSeleccionado(null)} />;

} else {

return <ListaEquipos equipos={equipos} onSelectEquipo={setEquipoSeleccionado} />;

}

Esto es lo que se conoce como **enrutamiento simple** o **manejo de vistas a nivel de estado**. Es perfecto para proyectos pequeños.

**Caso de Uso 3: Exportar la información a PDF**

Para esta funcionalidad, necesitarás una librería externa. react-to-pdf o jspdf son opciones populares. Para este caso, react-to-pdf es más sencillo porque te permite tomar un componente de React y convertirlo en PDF.

1. **Instala la librería**:

Bash

npm install react-to-pdf

1. **En tu componente DetalleEquipo.jsx**, importa la librería.
2. Crea un botón de "Exportar a PDF". Al hacer clic, activas la función de la librería, que toma una referencia al componente que quieres exportar y genera el archivo PDF.

JavaScript

// En DetalleEquipo.jsx

import React from 'react';

import { usePDF } from 'react-to-pdf';

const DetalleEquipo = ({ equipo, onBack }) => {

const { toPDF, targetRef } = usePDF({ filename: 'detalle-equipo.pdf' });

return (

<div>

<div ref={targetRef}>

<h2>{equipo.nombre}</h2>

<p>Estado: {equipo.estado}</p>

{/\* Más información del equipo aquí \*/}

</div>

<button onClick={onBack}>Regresar</button>

<button onClick={() => toPDF()}>Exportar a PDF</button>

</div>

);

};

Esta guía te da un plan de acción para construir la aplicación. Comenzar con la estructura de componentes te permitirá manejar cada caso de uso de manera modular y sencilla, que es la filosofía de React.