EDUCACIÓN IT

Bootcamp de Full Stack

Bienvenidos a la clase N°41

- Layout
- Hooks:
  - useContext
  - Custom hook
- Combinando recursos

IT

EDUCACIÓN IT

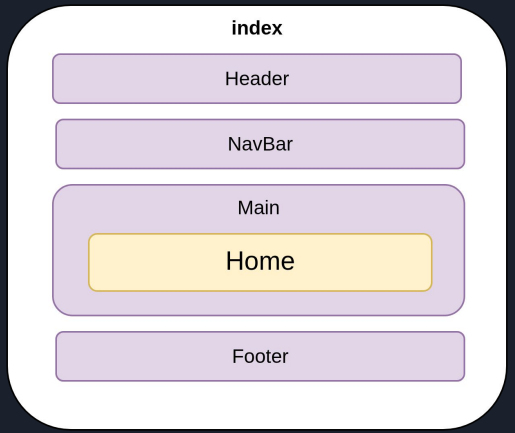
React

Layout

El **layout principal** es la estructura base de la interfaz de una aplicación. Generalmente incluye elementos persistentes en todas las vistas, como el encabezado (header), menú de navegación (navbar), pie de página (footer) y el área central (main) donde se renderiza el contenido de las páginas.

Árbol de directorios

```
src
├── components/
│   ├── buttons/
│   └── layout/
│       ├── header/
│       ├── navbar/
│       ├── main/
│       ├── footer/
│       └── Layout.jsx
├── pages/
│   └── home/
└── index.jsx
```



EDUCACIÓN IT

Hooks

useContext

El hook **useContext** permite acceder al valor de un contexto desde un componente funcional, sin necesidad de usar el componente <Context.Consumer>. Es útil para compartir datos globales como temas, usuarios, o configuración, sin pasar props manualmente por múltiples niveles.

Usos comunes:

- Acceder a datos globales como autenticación, idioma o tema.
- Compartir estado entre componentes sin prop drilling.
- Conectar componentes a un store global (useState).

Reglas

- Siempre debe usarse dentro del proveedor (<Context.Provider>) correspondiente.
- No reemplaza a un gestor de estado avanzado, pero es útil para datos simples y globales.

EDUCACIÓN IT

Hooks

Custom Hook

Un **custom hook** es una función reutilizable que comienza con use y permite encapsular lógica con estado o efectos, aprovechando los hooks de React. Se usa para abstraer comportamientos comunes entre componentes sin repetir código.

Usos comunes:

- Encapsular lógica de formularios, autenticación o peticiones HTTP.
- Compartir lógica de estado o efectos entre varios componentes.
- Mejorar la organización y legibilidad del código.

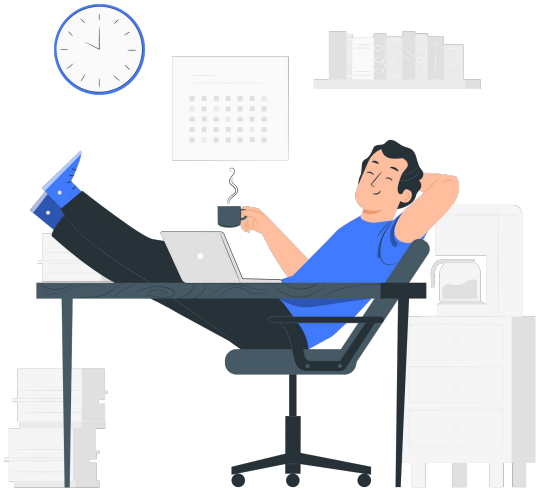
Reglas

- El nombre debe comenzar con use para que React lo identifique como hook.
- Puede usar otros hooks dentro (como useState, useEffect, etc.).
- Debe seguir las mismas reglas de los hooks.

EDUCACIÓN IT

BREAK

Descansemos 10 minutos



EDUCACIÓN IT

React

Combinando Recursos

- Una forma limpia y escalable de implementar un CRUD de productos en es combinando un custom hook (useProducts) con un contexto global (AppContext), utilizado dentro del proveedor (AppProvider).
- El contexto se encarga de mantener el estado compartido (como la lista de productos) y de exponer funciones como create, update, remove, entre otras.
- Por su parte, el hook useProducts simplifica el acceso al contexto y encapsula la lógica relacionada al dominio de productos.
- Esta estructura permite que cualquier componente acceda fácilmente al estado y acciones, evitando prop drilling y duplicación de lógica.

EDUCACIÓN IT

CIERRE DE CLASE

Continuaremos en la próxima clase