EDUCACIÓN IT

Bootcamp de Full Stack

Bienvenidos a la clase N°39

- Modularización
- Children
- Hooks
  - useState
- React Tools Developer
- Estructura de directorios
- Espacio de consultas

IT

EDUCACIÓN IT

React

Modularización

Componentes Internos

Son aquellos que se crean y usan dentro de un mismo módulo o aplicación. Generalmente son específicos para una funcionalidad o pantalla concreta y no están diseñados para ser reutilizados fuera de ese contexto.

Componentes Externos

Son componentes desarrollados para ser reutilizados en diferentes proyectos o módulos. Pueden ser parte de librerías, frameworks o paquetes independientes, y están diseñados para ser genéricos, configurables y fáciles de integrar.

EDUCACIÓN IT

React

Children

El **children** es una prop especial que representa el contenido que se incluye entre las etiquetas de apertura y cierre de un componente. Permite que un componente actúe como contenedor y renderice otros elementos o componentes dentro de su estructura.

Características:

- Todos los componentes reciben **props.children** automáticamente, sin necesidad de declararlo.
- Su valor puede ser un nodo React (elemento, texto, número), un array, una función o null.
- Es una forma de composición que permite mayor flexibilidad, que es diferente a pasar props específicas.

EDUCACIÓN IT

Hooks

Concepto

Los **Hooks** son funciones especiales introducidas en **React 16.8** que permiten usar estado, efectos, contexto y otras funcionalidades en componentes funcionales, sin necesidad de usar clases.

Características:

- Facilitan la reutilización de lógica entre componentes.
- Contribuyen a un código más limpio y organizado.
- Permiten dividir la lógica según su comportamiento.

Reglas:

1. Solo pueden usarse dentro de componentes funcionales o custom hooks.
2. Deben llamarse siempre en el nivel superior del componente o hook.
3. No se deben usar dentro de condicionales, bucles ni funciones.
4. Siempre comienzan con el prefijo use.

EDUCACIÓN IT

Hooks

useState

El hook **useState** permite agregar y gestionar **estado** local en componentes funcionales. Devuelve el valor actual del estado y una función para actualizarlo.

Usos comunes:

- Guardar el valor de inputs, selects o textareas.
- Controlar la visibilidad de elementos.
- Llevar conteos o cantidades.
- Gestionar cualquier estado interno del componente.

Reglas

- Cada llamada a **useState** es independiente y conserva su propio valor.

Sintaxis

```
const [hasChanges, setHasChanges] = useState(true);
```

EDUCACIÓN IT

React

React Tools Developer

¿Qué es **React Developer Tools**?

Es una extensión oficial para navegadores (Chrome, Firefox, Edge, etc.) que permite inspeccionar la estructura interna de una aplicación React en tiempo real. Es una herramienta clave para desarrolladores, ya que facilita la depuración, el análisis de rendimiento y la comprensión del estado y props de los componentes.

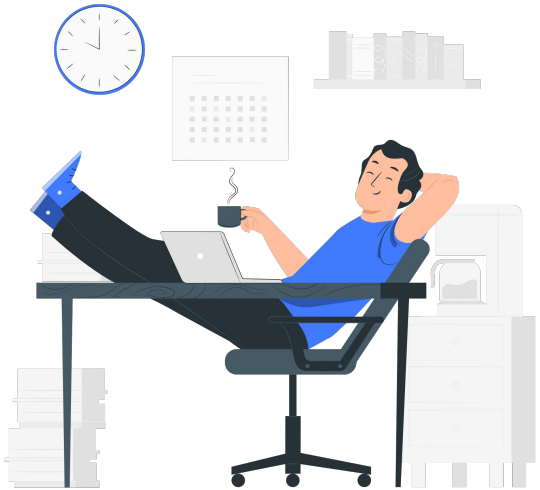
Uso:

- Visualizar la jerarquía de componentes como un árbol interactivo.
- Buscar componentes por nombre en la estructura de la app.
- Inspeccionar props, state, hooks y contextos de cada componente.
- Editar valores de props y state para probar distintos escenarios.
- Medir el rendimiento de renderizado de cada componente (Profiler).
    - Ver qué componente causa un renderizado y cuándo ocurre.
    - Identificar rápidamente componentes que se vuelven a renderizar innecesariamente.

EDUCACIÓN IT

BREAK

Descansemos 10 minutos



EDUCACIÓN IT

React

Estructura De Directorios

La **estructura de directorios** refiere a la forma en que se organizan carpetas y archivos dentro de un proyecto. Una estructura clara y lógica facilita el mantenimiento, la escalabilidad y la localización de recursos.

Un **árbol de directorios** es una representación jerárquica que muestra cómo se organizan las carpetas y archivos en un sistema.

```
src
├── assets/
│   ├── images/
│   └── icons/
├── components/
│   └── buttons/
├── pages/
│   └── home/
├── scss
└── utils/
    └── paths.js
```

EDUCACIÓN IT

Repaso

Espacio de Consultas

Temas

1. Componentes
2. Props
3. PropTypes
4. Children
5. Eventos
6. Clases
7. Hooks
8. Estado
9. Sass
10. Estructura de carpetas

EDUCACIÓN IT

CIERRE DE CLASE

Continuaremos en la próxima clase