

03 - Tecnicatura Superior en Ciencias de Datos e Inteligencia Artificial

Componentes, Props, Integración de Tailwind CSS y Consumo de APIs con React

1. ¿Qué es un Componente en React?

Un **componente** en React es una función que devuelve **JSX** (una mezcla entre HTML y JavaScript) y permite **modularizar la interfaz** en partes reutilizables como botones, formularios o tarjetas de producto.

Ejemplo de un componente funcional:

```
const Saludo = () => {  
  return <h1>Hola, mundo</h1>;  
};
```

2. ¿Qué son las Props?

Las **props** (abreviación de *properties*) permiten que los componentes sean **dinámicos y reutilizables**, enviando datos desde el componente padre al hijo.

Ejemplo:

```
const Saludo = ({ nombre }) => {
```

```
return <h1>Hola, {nombre}</h1>;  
};
```

// Uso:

```
<Saludo nombre="Lucía" />
```

3. Integración de Tailwind CSS en un Proyecto Vite + React

1. Crear proyecto con:

```
npm create vite@latest
```

1. Instalar Tailwind:

```
npm install -D tailwindcss postcss autoprefixer  
npx tailwindcss init -p
```

1. Configurar `tailwind.config.js` :

```
content: ["/index.html", "/src/**/*.{js,jsx}"]
```

1. Importar en `src/index.css` :

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

4. Consumiendo Fake Store API con Axios en React

Fake Store API es una API REST gratuita para practicar con productos de e-commerce.

Instalación de Axios:

```
npm install axios
```

Uso en un componente:

```
import { useEffect, useState } from "react";
import axios from "axios";

const Productos = () => {
  const [productos, setProductos] = useState([]);

  useEffect(() => {
    axios.get("https://fakestoreapi.com/products?limit=20")
      .then(res => setProductos(res.data))
      .catch(err => console.error(err));
  }, []);

  return (
    <div className="p-4 grid grid-cols-1 md:grid-cols-4 gap-4">
      {productos.map(producto => (
        <div key={producto.id} className="border p-4 rounded shadow">
          <h2 className="font-bold">{producto.title}</h2>
          <p>${producto.price}</p>
        </div>
      ))}
    </div>
  );
};
```

```
export default Productos;
```

5. Estadísticas básicas: precios máximo, mínimo y promedio

Agrega lógica para calcular estadísticas básicas:

```
const precios = productos.map(p => p.price);  
const max = Math.max(...precios);  
const min = Math.min(...precios);  
const avg = (precios.reduce((a, b) => a + b, 0) / precios.length).toFixed(2);
```

```
<div className="p-4 bg-gray-100 mb-4 rounded">  
  <p><strong>Precio máximo:</strong> ${max}</p>  
  <p><strong>Precio mínimo:</strong> ${min}</p>  
  <p><strong>Precio promedio:</strong> ${avg}</p>  
</div>
```

6. Extra: Diseño con Tailwind + API

Aplicar utilidades de Tailwind para que el diseño sea responsive, ordenado y visualmente atractivo. Por ejemplo, botones con `hover`, tarjetas con `shadow`, o `grid` para disposición de productos.