

Consumo de API de Conversión en React

```
import { useEffect, useState } from 'react';
```

```
import './App.css';
```

useEffect y useState son hooks de React:

useState se utiliza para manejar el estado local del componente.

useEffect se usa para ejecutar funciones secundarias (side effects), como hacer peticiones a APIs.

Se importa el archivo CSS App.css para aplicar estilos.

Definición del componente principal

jsx

Copiar

Editar

```
function App() {  
  const [monedas, setMonedas] = useState([]);
```

Se define el componente funcional App.

Se crea una variable de estado monedas inicializada como un arreglo vacío.

setMonedas es la función que actualiza el estado monedas.

Efecto para obtener datos de la API

jsx

Copiar

Editar

```
useEffect(() => {
```

```

fetch('https://localhost:3000/monedas/')
  .then((res) => {
    return res.json();
  })
  .then((data) => {
    console.log(data)
    setMonedas(data);
  })
}, []);

```

useEffect se ejecuta una sola vez al montar el componente (debido al arreglo vacío []).

Se hace una petición fetch a <https://localhost:3000/monedas/>, que es una API local.

La respuesta se convierte en formato JSON.

El resultado se imprime en consola y luego se almacena en el estado monedas mediante setMonedas.

Renderizado del contenido

jsx

Copiar

Editar

```

return (
  <>
    <h1>Información de monedas</h1>

```

Se renderiza un encabezado de nivel 1 con el título "Información de monedas".

Tabla con la información

jsx

Copiar

Editar

```
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Origen</th>
      <th>Destino</th>
      <th>Valor</th>
    </tr>
  </thead>
  <tbody>
    {
      monedas.map((item, index) => {
        return (
          <tr key={index}>
            <td>{item.id}</td>
            <td>{item.origen}</td>
            <td>{item.destino}</td>
            <td>{item.valor}</td>
          </tr>
        )
      })
    }
  </tbody>
</table>
```

Se crea una tabla con encabezados para mostrar los datos de cada moneda: ID, origen, destino y valor.

Se usa `.map()` para iterar sobre cada elemento del arreglo monedas y se genera una fila (`<tr>`) por cada elemento.

Cada fila muestra los valores individuales usando `item.id`, `item.origen`, `item.destino` y `item.valor`.

Lista comentada

jsx

Copiar

Editar

```
<ul>
  {
    // paises.map( (item, index) => <li key={index}>{ item.name.common }</li>)
  }
</ul>
```

Exportación del componente

jsx

Copiar

Editar

```
export default App;
```

Se exporta el componente App para que pueda ser usado en otros archivos del proyecto.