Práctica Sesión 11: Cliente React para la API de Empleados Sesión 10

Cliente React para consumir una API REST de empleados

Crear un **cliente web en React** que se conecte a una **API REST hecha con Node.js + Express**, obtenga la lista de empleados y la muestre en pantalla con estadísticas básicas.

Descripción de la aplicación a desarrollar

La aplicación React debe conectarse a la API:

GET http://localhost:3000/empleados

La API devuelve un array JSON de este estilo:

```
[
  { "id": 1, "nombre": "Luis", "salario": 2000 },
  { "id": 2, "nombre": "Marta", "salario": 2400 },
  { "id": 3, "nombre": "Diego", "salario": 1800 }
]
```

El cliente React debe:

1. Conectarse a la API

- Usar fetch para hacer una petición GET al endpoint /empleados.
- Mostrar errores si la API no responde.

2. Gestionar estados en React

Utilizar useState para guardar:

- La lista de empleados recibida.

- Un texto de error si ocurre algún fallo.
- Un indicador de carga (cargando = true mientras la API responde).

Usar useEffect para llamar a la API **solo al cargar la página**.

3. Mostrar la información en pantalla

La app debe mostrar:

- Un título: **Lista de empleados**
- Mientras se carga:
  "Cargando empleados..."
- Si hay error:
  Un texto rojo con el mensaje.
- Si la carga es correcta:
  - Total de empleados.
  - Suma de salarios.
  - Salario medio.
  - Una tabla o lista con:
    - ID
    - Nombre
    - Salario

Requisitos técnicos

API Node.js + Express (servidor backend)

Debe incluir:

- Una ruta GET /empleados.
- Respuesta JSON.

- Activar **CORS** con:

```
const cors = require("cors");

app.use(cors());
```

¿Qué es CORS?

CORS significa:

Cross-**O**rigin **R**esource **S**haring
(Intercambio de Recursos de Orígenes Cruzados)

Es **un sistema de seguridad de los navegadores** que *bloquea peticiones HTTP* hechas desde una página web a un **servidor con diferente origen**.

Por esta razón debemos modificar el archivo de la semana pasada y añadirlo.

React (puerto 5173) llama a la API (puerto 3000), lo que supone tener orígenes distintos

Cliente React (frontend)

Debe cumplir:

- Uso de useState.
- Uso de useEffect.
- Petición fetch() correctamente implementada.
- Mostrar información de manera clara.
- Calcular estadísticas básicas.

Pasos para ejecutar la práctica

1. Arrancar la API con Node:

node app.js

Debe verse:

API funcionando en puerto 3000

Y en el navegador:

http://localhost:3000/empleados

2. Arrancar el cliente React:

npm run dev

Abrir en navegador:

http://localhost:5173/

Solución paso a paso

Vamos a montar:

- Una **API Node + Express** (api-empleados)
- Un **cliente React con Vite** (cliente-react-empleados)
- Activaremos **CORS** y modificaremos app.js.

PARTE 1: Crear la API de empleados (Node + Express + CORS)

Puedes hacer uno nuevo o modificar el de la semana pasada

1. Crear la carpeta de la API

En **CMD o PowerShell**:

cd C:\Users\Usuario\Desktop

mkdir api-empleados

cd api-empleados

Ahora estás en:

C:\Users\Usuario\Desktop\api-empleados>

2. Inicializar el proyecto Node

Dentro de api-empleados:

npm init -y

Esto crea un package.json.

Esto si debes hacerlo:

3. Instalar las dependencias: express y cors

En la **misma carpeta**:

npm install express cors

Esto crea node_modules y añade ambas dependencias.

4. Crear el archivo app.js

En la carpeta api-empleados, crea o modifica el archivo app.js de la semana pasada:

```
// Importamos los módulos necesarios
const express = require('express');
const cors = require('cors');

// Creamos la aplicación express
const app = express();

// Activamos CORS para permitir peticiones desde el cliente React
app.use(cors());

// Datos de ejemplo: listado de empleados
const empleados = [
  { id: 1, nombre: 'Luis', salario: 2000 },
  { id: 2, nombre: 'Marta', salario: 2400 },
  { id: 3, nombre: 'Diego', salario: 1800 },
];

// Endpoint GET /empleados que devuelve el array en formato JSON
app.get('/empleados', (req, res) => {
  res.json(empleados);
});
```

```
// Ponemos la API a escuchar en el puerto 3000
const PORT = 3000;
app.listen(PORT, () => {
  console.log(`API funcionando en puerto ${PORT}`);
});
```

Fíjate en estas partes importantes:

- app.use(cors()); Esto es lo que permite que tu React (puerto 5173) pueda llamar a la API (puerto 3000).
- Ruta: /empleados
- Puerto: 3000

5. Arrancar la API

En la terminal (situado en C:\Users\Usuario\Desktop\api-empleados):

node app.js

Deberías ver:

API funcionando en puerto 3000

Déjala **abierta**. Esa ventana de terminal tiene que seguir ejecutándose.

6. Comprobar que la API funciona

Abre el navegador y entra en:

http://localhost:3000/empleados

Si todo va bien, deberías ver el JSON:

[
{ "id": 1, "nombre": "Luis", "salario": 2000 },
{ "id": 2, "nombre": "Marta", "salario": 2400 },
{ "id": 3, "nombre": "Diego", "salario": 1800 }
]

Si llegas hasta aquí, la **API está perfecta**.

PARTE 2: Crear el cliente React con Vite

Vite es una herramienta moderna para crear proyectos web (como React, Vue, Svelte…).
Sirve para arrancar, desarrollar y construir aplicaciones mucho más rápido que herramientas anteriores como *Create React App*.

Esto va en **otra carpeta distinta**.

7. Crear el proyecto React con Vite

Abre **otra** terminal (nueva ventana o pestaña) y ejecuta:

cd C:\Users\Usuario\Desktop

npm create vite@latest cliente-react-empleados -- --template react

cd cliente-react-empleados

npm install

Ahora estás en:

C:\Users\Usuario\Desktop\cliente-react-empleados>

8. Modificar la carpeta src/

En la carpeta cliente-react-empleados:

1. Entra en src/ y **borra todo lo que haya** (por ejemplo App.jsx, main.jsx, etc.)
2. Pega los **dos archivos que te he dejado en el repositorio**:

- src/main.jsx
- src/App.jsx

PARTE 3: Arrancar todo junto

9. Arrancar la API (ventana 1)

En una terminal:

cd C:\Users\Usuario\Desktop\api-empleados

node app.js

Déjala ejecutándose.

10. Arrancar React (ventana 2)

En otra terminal:

cd C:\Users\Usuario\Desktop\cliente-react-empleados

npm run dev

Te mostrará algo como:

Local: http://localhost:5173/

Abre esa URL en el navegador.

11. ¿Qué deberías ver?

En http://localhost:5173/:

- Título: **Lista de empleados**
- Si está cargando: **Cargando empleados...**
- Luego:
    - Total de empleados
    - Suma de salarios
    - Salario medio
    - Tabla con Luis, Marta, Diego...