

Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas



Ingeniería Telemática

Routing y Consumo de APIs en React

Bautista Uribe Juan Manuel

Aplicaciones distribuidas.

4TM3

CONTEXTO:

1. Configuración del Sistema de Rutas (Routing)

Se instaló la librería fundamental para la gestión de rutas en React: `react-router-dom`.

- **Envoltorio del Router:** Se modificó el archivo `src/main.jsx` para envolver el componente principal (`<App />`) con `<BrowserRouter>` . Esto habilitó la funcionalidad de rutas para toda la aplicación.

2. Creación de Vistas y Navegación

Se creó una carpeta `src/pages` para alojar las vistas y se definieron los componentes básicos requeridos :

- **Home.jsx:** Página de inicio.
- **About.jsx:** Página de información.
- **Usuarios.jsx:** Contenedor inicial para la lista de usuarios.

Posteriormente, se transformó `src/App.jsx` en el gestor de la interfaz, definiendo:

- **Menú de Navegación:** Se implementaron los enlaces (`<Link>`) que permiten al usuario moverse entre las rutas sin recargar la página .
- **Definición de Rutas:** Se usaron los componentes `<Routes>` y `<Route>` para mapear las URLs a los componentes de página correspondientes (`/usuarios` a `<Usuarios />`, etc.) . Se incluyó una ruta `path="*"` para manejar páginas no encontradas (error 404) .

3. Conexión con API y Consumo de Datos (Hook `useEffect`)

El consumo de datos se implementó en el componente `src/pages/Usuarios.jsx` para cumplir con el requisito de listar usuarios desde una API.

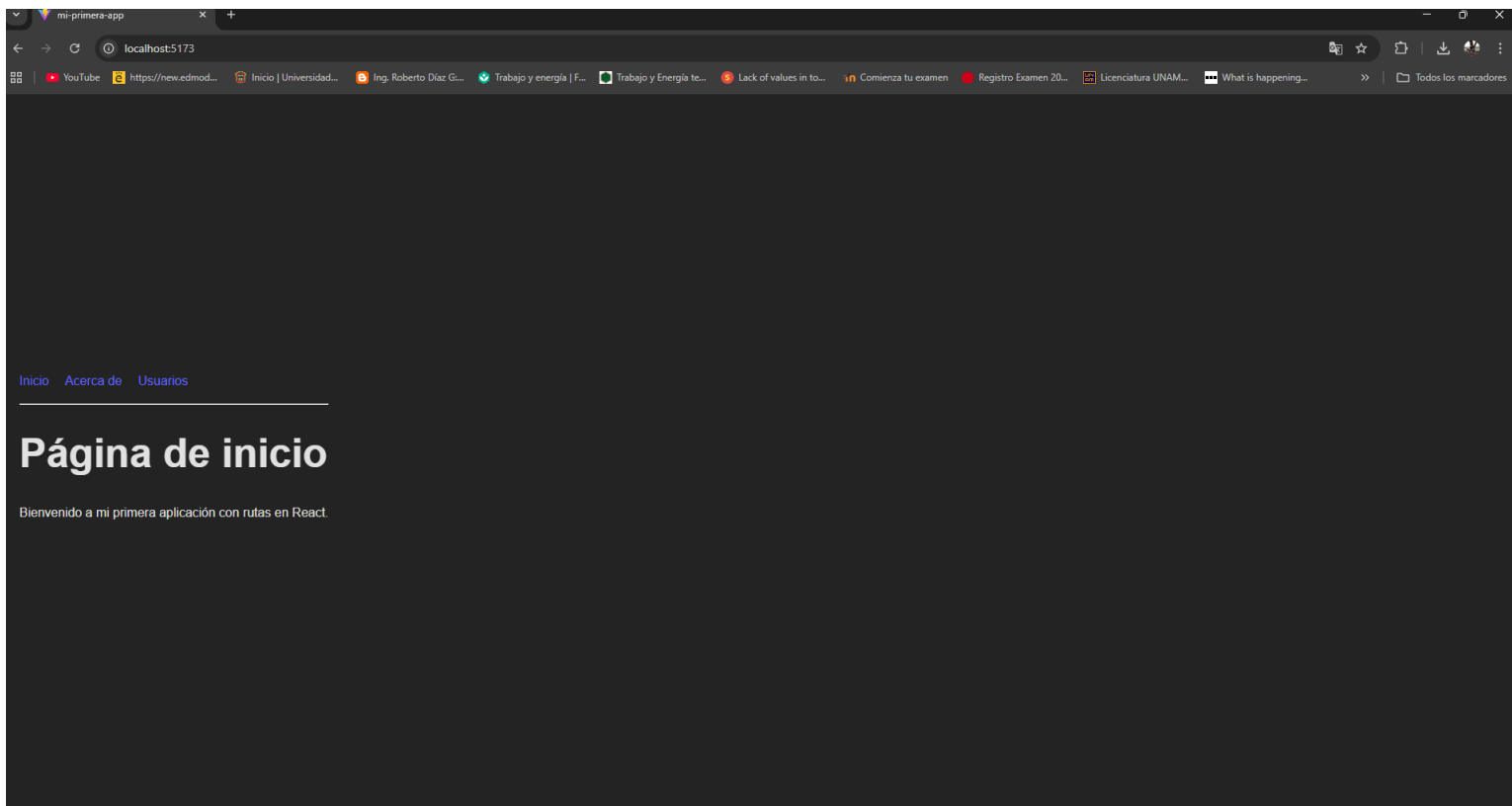
- **Manejo de Estado:** Se declararon tres estados fundamentales con `useState` : `usuarios` (para guardar los datos), `cargando` (para mostrar un *loader*) y `error` (para capturar fallos de red).
- **Lógica de Fetching:** La función `cargarUsuarios` se ejecutó dentro de `useEffect` , asegurando que la llamada a la API solo se realice una vez al cargar la página.
- **Flujo de Datos:** Se utilizó `fetch()` para obtener los datos , se manejó la lógica de errores y se usó `setUsuarios` para actualizar la lista recibida.
- **Renderizado Condicional:** Se implementó un renderizado condicional para mostrar:
 - Un mensaje de "Cargando usuarios..." mientras se espera la respuesta de la API.

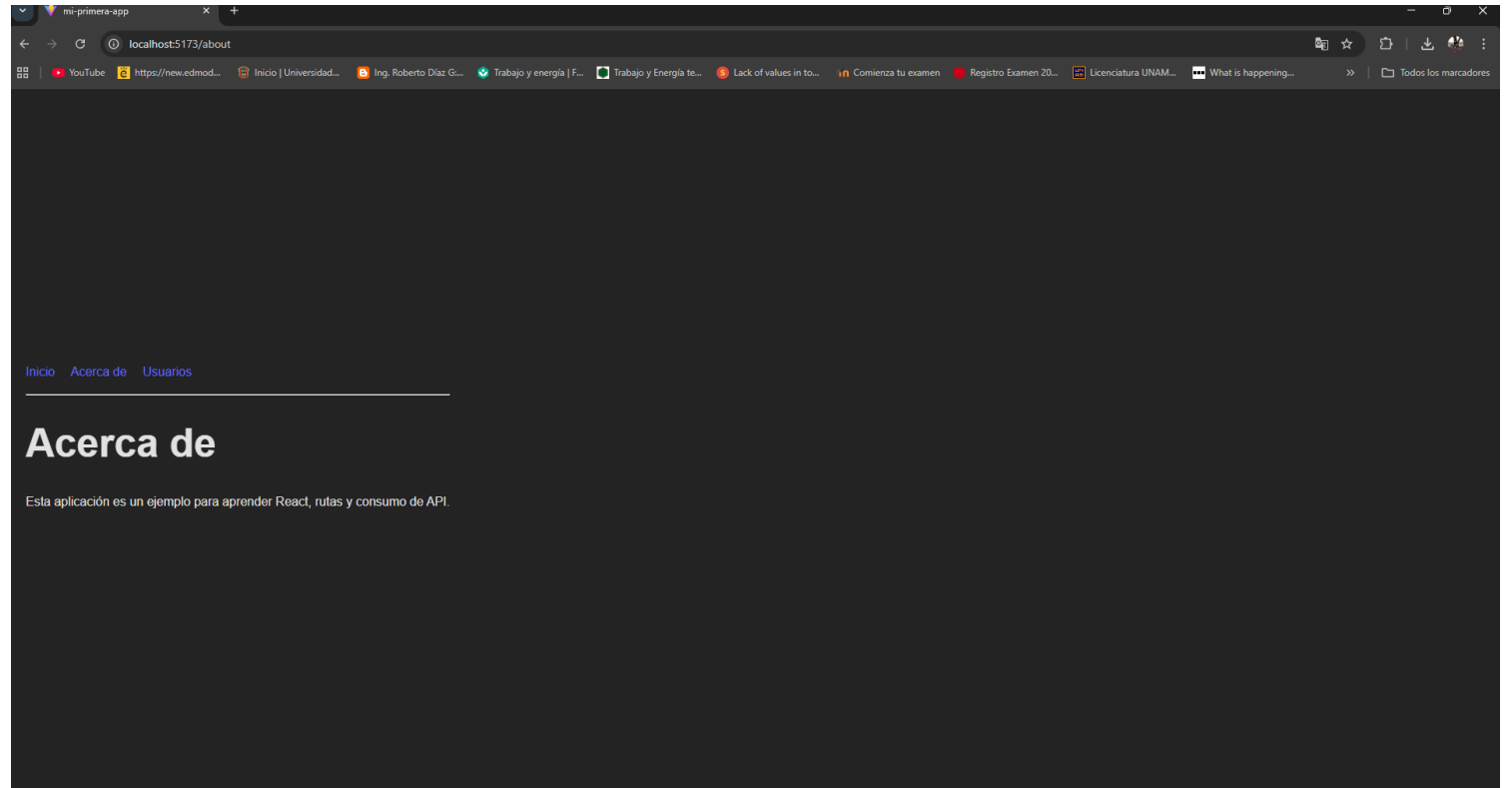
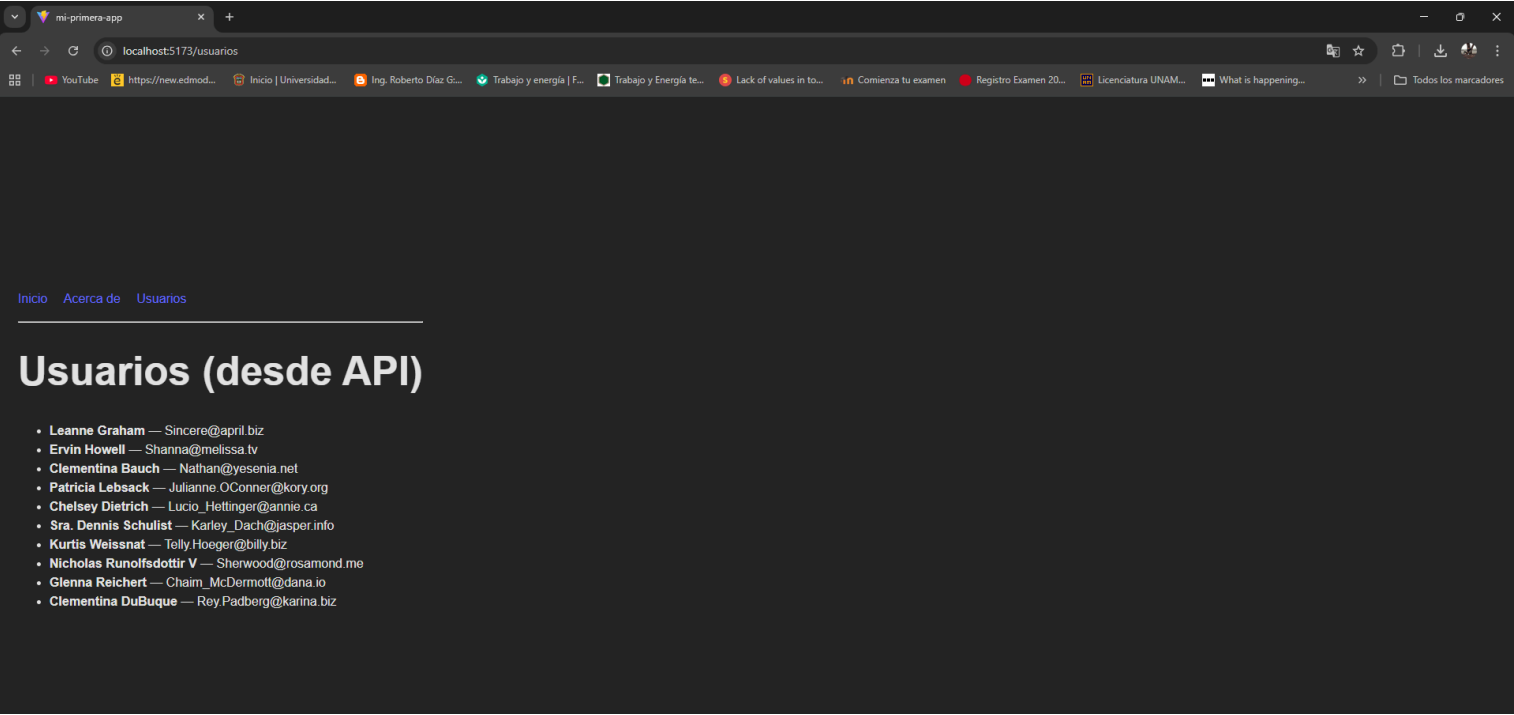
- Un mensaje de error si la conexión falla.
- Finalmente, la lista completa de usuarios mapeados en formato `` si la llamada es exitosa .

Conclusión de la Práctica

El proyecto **mi-app2-con-apis** ha sido completado al nivel más avanzado requerido, demostrando el uso de herramientas de producción como Vite, la gestión de componentes y estado (`useState`, `useEffect`), y la integración esencial de la aplicación con servicios externos (APIs) a través de `react-router-dom`. La verificación final confirma que la navegación funciona y la lista de usuarios de la API es visible y correcta.

Evidencia:





```

34 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> mkdir src\pages

Directorio: C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app\src

Mode                LastWriteTime         Length Name
----                -
d-----          05/12/2025   03:59 a. m.         pages

PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> # 1. Componente Home.jsx
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> @'
>> function Home() {
>>   return (
>>     <div>
>>       <h1>Página de inicio</h1>
>>       <p>Bienvenido a mi primera app con rutas en React.</p>
>>     </div>
>>   )
>> }
>> export default Home
>> '@ | Out-File -FilePath "src/pages/Home.jsx" -Encoding UTF8 -Force
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app>
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> # 2. Componente About.jsx
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> @'
>> function About() {
>>   return (
>>     <div>
>>       <h1>Acerca de</h1>
>>       <p>Esta app es un ejemplo para aprender React, rutas y consumo de APIs.</p>
>>     </div>
>>   )
>> }
>> export default About

```

```

>> export default About
>> '@ | Out-File -FilePath "src/pages/About.jsx" -Encoding UTF8 -Force
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app>
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> # 3. Componente Usuarios.jsx (Inicial)
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> @'
>> function Usuarios() {
>>   return (
>>     <div>
>>       <h1>Usuarios</h1>
>>       <p>Aquí mostraremos usuarios obtenidos desde una API.</p>
>>     </div>
>>   )
>> }
>> export default Usuarios
>> '@ | Out-File -FilePath "src/pages/Usuarios.jsx" -Encoding UTF8 -Force
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app>
>> import React from 'react'
>> import ReactDOM from 'react-dom/client'
>> import { BrowserRouter } from 'react-router-dom'
>> import App from './App'
>> import './index.css'
>>
>> ReactDOM.createRoot(document.getElementById('root')).render(
>>   <React.StrictMode>
>>     <BrowserRouter>
>>       <App />
>>     </BrowserRouter>
>>   </React.StrictMode>,
>> )
>> '@ | Out-File -FilePath "src/main.jsx" -Encoding UTF8 -Force
PS C:\Users\juannm\Aplicaciones-Distribuidas\mi-primer-app> @'
>> import { Link, Routes, Route } from 'react-router-dom'
>> import Home from './pages/Home'
>> import About from './pages/About'
>> import Usuarios from './pages/Usuarios'
>>
>> function App() {

```

```

>> function App() {
>>   return (
>>     <div style={{ padding: '1.5rem', fontFamily: 'sans-serif' }}>
>>       {/* Menú de navegación */}
>>       <nav style={{ marginBottom: '1rem' }}>
>>         <Link to="/" style={{ marginRight: '1rem' }}>Inicio</Link>
>>         <Link to="/about" style={{ marginRight: '1rem' }}>Acerca de</Link>
>>         <Link to="/usuarios">Usuarios</Link>
>>       </nav>
>>
>>       <hr />
>>
>>       {/* Definición de rutas */}
>>       <Routes>
>>         <Route path="/" element={Home} /> />
>>         <Route path="/about" element={About} /> />
>>         <Route path="/usuarios" element={Usuarios} /> />
>>         {/* ruta "catch-all" para 404 */}
>>         <Route path="*" element={404} /> />
>>       </Routes>
>>     </div>
>>   )
>> }
>> export default App
>> '@ | Out-File -FilePath "src/App.jsx" -Encoding UTF8 -Force
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> npm run dev

> mi-primer-app@0.0.0 dev
> vite

4:00:24 a.m. [vite] (client) Re-optimizing dependencies because lockfile has changed

VITE v7.2.6 ready in 518 ms

```

```

found 0 vulnerabilities
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> '@ | Out-File -FilePath "src/main.jsx" -Encoding UTF8 -Force
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> # Crear carpeta pages
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> mkdir src\pages
mkdir : Ya existe un elemento con el nombre especificado: C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app\src\pages.
En línea: 1 Carácter: 1
+ mkdir src\pages
+ ~~~~~
+ CategoryInfo          : ResourceExists: (C:\Users\juanm\...a-app\src\pages:String) [New-Item], IOException
+ FullyQualifiedErrorId : DirectoryExist,Microsoft.PowerShell.Commands.NewItemCommand

PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app>
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> # 1. Crear Home.jsx
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> '@ | Out-File -FilePath "src/Home.jsx" -Encoding UTF8 -Force
>> function Home() {
>>   return (
>>     <div>
>>       <h1>Página de inicio</h1>
>>       <p>Bienvenido a mi primera app con rutas en React.</p>
>>     </div>
>>   )
>> }
>> export default Home
>> '@ | Out-File -FilePath "src/Home.jsx" -Encoding UTF8 -Force
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app>

```

```
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app>
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> # 1. Crear Home.jsx
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> @'
>> function Home() {
>>   return (
>>     <div>
>>       <h1>Página de inicio</h1>
>>       <p>Bienvenido a mi primera app con rutas en React.</p>
>>     </div>
>>   )
>> }
>> export default Home
>> '@ | Out-File -FilePath "src/pages/Home.jsx" -Encoding UTF8 -Force
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app>
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> # 2. Crear About.jsx
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> @'
>> function About() {
>>   return (
>>     <div>
>>       <h1>Acerca de</h1>
>>       <p>Esta app es un ejemplo para aprender React, rutas y consumo de APIs.</p>
>>     </div>
>>   )
>> }
>> export default About
>> '@ | Out-File -FilePath "src/pages/About.jsx" -Encoding UTF8 -Force
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app>
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> # 3. Crear Usuarios.jsx (Versión Inicial)
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> @'
>> function Usuarios() {
>>   return (
>>     <div>
>>       <h1>Usuarios</h1>
>>       <p>Aquí mostraremos usuarios obtenidos desde una API.</p>
>>     </div>
>>   )
>> }
>> }
```

```

>> export default Usuarios
>> '@ | Out-File -FilePath "src/pages/Usuarios.jsx" -Encoding UTF8 -Force
PS C:\Users\juann\Aplicaciones-Distribuidas\mi-primer-app> @'
>> import { Link, Routes, Route } from 'react-router-dom'
>> import Home from './pages/Home'
>> import About from './pages/About'
>> import Usuarios from './pages/Usuarios'
>>
>> function App() {
>>   return (
>>     <div style={{ padding: '1.5rem', fontFamily: 'sans-serif' }}>
>>       {/* Menú de navegación */}
>>       <nav style={{ marginBottom: '1rem' }}>
>>         <Link to="/" style={{ marginRight: '1rem' }}>Inicio</Link>
>>         <Link to="/about" style={{ marginRight: '1rem' }}>Acerca de</Link>
>>         <Link to="/usuarios">Usuarios</Link>
>>       </nav>
>>
>>       <hr />
>>
>>       {/* Definición de rutas */}
>>       <Routes>
>>         <Route path="/" element={<Home />} />
>>         <Route path="/about" element={<About />} />
>>         <Route path="/usuarios" element={<Usuarios />} />
>>         {/* ruta "catch-all" para 404 */}
>>         <Route path="*" element={<h2>Página no encontrada (404)</h2>} />
>>       </Routes>
>>     </div>
>>   )
>> }
>> export default App
>> '@ | Out-File -FilePath "src/App.jsx" -Encoding UTF8 -Force
PS C:\Users\juann\Aplicaciones-Distribuidas\mi-primer-app> @'
>> import { useEffect, useState } from 'react'

```

```

>> '@ | Out-File -FilePath "src/App.jsx" -Encoding UTF8 -Force
PS C:\Users\juann\Aplicaciones-Distribuidas\mi-primer-app> @'
>> import { useEffect, useState } from 'react'
>>
>> function Usuarios() {
>>   const [usuarios, setUsuarios] = useState([])
>>   const [cargando, setCargando] = useState(true)
>>   const [error, setError] = useState(null)
>>
>>   useEffect(() => {
>>     // Llamada a la API al montar el componente
>>     async function cargarUsuarios() {
>>       try {
>>         setCargando(true)
>>         setError(null)
>>
>>         const respuesta = await fetch('https://jsonplaceholder.typicode.com/users')
>>
>>         if (!respuesta.ok) {
>>           throw new Error('Error al obtener los usuarios')
>>         }
>>
>>         const datos = await respuesta.json()
>>         setUsuarios(datos)
>>       } catch (err) {
>>         setError(err.message)
>>       } finally {
>>         setCargando(false)
>>       }
>>     }
>>
>>     cargarUsuarios()
>>   }, []) // [] => solo se ejecuta una vez al cargar la página
>>
>>   if (cargando) {
>>     return <p>Cargando usuarios...</p>
>>   }
>>
>>   if (error) {
>>     return <p>Ocurrió un error: {error}</p>

```

```

>>   if (cargando) {
>>       return <p>Cargando usuarios...</p>
>>   }
>>
>>   if (error) {
>>       return <p>Ocurrió un error: {error}</p>
>>   }
>>
>>   return (
>>       <div>
>>           <h1>Usuarios (desde API)</h1>
>>           <ul>
>>               {usuarios.map((usuario) => (
>>                   <li key={usuario.id}>
>>                       <strong>{usuario.name}</strong> - {usuario.email}
>>                   </li>
>>               ))}
>>           </ul>
>>       </div>
>>   )
>> }
>>
>> export default Usuarios
>> '@ | Out-File -FilePath "src/pages/Usuarios.jsx" -Encoding UTF8 -Force
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> npm run dev

> mi-primer-app@0.0.0 dev
> vite

VITE v7.2.6  ready in 601 ms

  → Local:   http://localhost:5173/
  → Network: use --host to expose
  → press h + enter to show help
PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> cd ..
PS C:\Users\juanm\Aplicaciones-Distribuidas> Rename-Item -Path ".\mi-primer-app" -NewName "mi-app2-con-apis"
PS C:\Users\juanm\Aplicaciones-Distribuidas> git add .
warning: in the working copy of 'mi-app2-con-apis/.gitignore', LF will be replaced by CRLF the next time Git touches it

```

```

PS C:\Users\juanm\Aplicaciones-Distribuidas\mi-primer-app> cd ..
PS C:\Users\juanm\Aplicaciones-Distribuidas> Rename-Item -Path ".\mi-primer-app" -NewName "mi-app2-con-apis"
PS C:\Users\juanm\Aplicaciones-Distribuidas> git add .
warning: in the working copy of 'mi-app2-con-apis/.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/README.md', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/eslint.config.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/package-lock.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/package.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/App.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/App.jsx', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/Saludo.jsx', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/index.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/main.jsx', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/pages/About.jsx', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/pages/Home.jsx', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/src/pages/Usuarios.jsx', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'mi-app2-con-apis/vite.config.js', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\juanm\Aplicaciones-Distribuidas> git commit -m "Practica final de React: Implementación de Router y consumo de APIs en mi-app2-con-apis"
[main 483989c] Practica final de React: Implementación de Router y consumo de APIs en mi-app2-con-apis
29 files changed, 136 insertions(+), 29 deletions(-)
rename {mi-primer-app => mi-app2-con-apis}/.gitignore (100%)
rename {mi-primer-app => mi-app2-con-apis}/BUJM_React.001.pdf (100%)
rename {mi-primer-app => mi-app2-con-apis}/README.md (100%)
rename {mi-primer-app => mi-app2-con-apis}/eslint.config.js (100%)
rename {mi-primer-app => mi-app2-con-apis}/index.html (100%)
rename {mi-primer-app => mi-app2-con-apis}/package-lock.json (98%)
rename {mi-primer-app => mi-app2-con-apis}/package.json (90%)
rename {mi-primer-app => mi-app2-con-apis}/public/vite.svg (100%)
rename {mi-primer-app => mi-app2-con-apis}/src/App.css (100%)
create mode 100644 mi-app2-con-apis/src/App.jsx
create mode 100644 mi-app2-con-apis/src/Saludo.jsx
rename {mi-primer-app => mi-app2-con-apis}/src/assets/react.svg (100%)
rename {mi-primer-app => mi-app2-con-apis}/src/index.css (100%)
create mode 100644 mi-app2-con-apis/src/main.jsx
create mode 100644 mi-app2-con-apis/src/pages/About.jsx
create mode 100644 mi-app2-con-apis/src/pages/Home.jsx

```