«Talento Tech»

React JS

Clase 07

Buenos Aires Ciudad

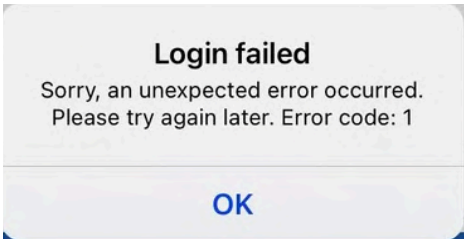
Clase N° 7 | Rutas Dinámicas y Protegidas

Índice:

Objetivos de la Clase:

- Aprender a crear rutas dinámicas que utilicen parámetros para renderizar información específica.
- Configurar rutas protegidas para restringir el acceso a ciertas páginas a usuarios autenticados.
- Implementar redirecciones para gestionar usuarios no autenticados.

Introducción a Rutas Dinámicas y Protegidas



¿Qué son las rutas dinámicas?

Las rutas dinámicas son aquellas que contienen parámetros variables, permitiendo mostrar contenido específico basado en la URL. Por ejemplo, en un eCommerce, la ruta /productos/:id permite mostrar información detallada de un producto con el identificador id.

Creación de Rutas Dinámicas

- **Configuración de rutas dinámicas**
  Para trabajar con rutas dinámicas, se utiliza el componente **Route** de React Router, añadiendo parámetros en el path.

```
import React from 'react';
import { Routes, Route } from 'react-router-dom';
import Home from './pages/Home';
import Productos from './pages/Productos';
import ProductoDetalle from './pages/ProductoDetalle';

function App() {
  return (
    <div>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/productos" element={<Productos />} />
        <Route path="/productos/:id" element={<ProductoDetalle />} />
      </Routes>
    </div> );}
export default App;
```

- **Uso del hook `useParams`**
  El hook useParams permite acceder a los parámetros de la URL dentro del componente.

Ejemplo del componente **ProductoDetalle**:

```
import React from 'react';
import { useParams } from 'react-router-dom';

function ProductoDetalle() {
  const { id } = useParams();

  return (
    <div>
      <h1>Detalle del Producto</h1>
      <p>Este es el detalle del producto con ID: {id}</p>
    </div>
  );
}

export default ProductoDetalle;
```

Rutas Protegidas

¿Qué son las rutas protegidas?

Las rutas protegidas restringen el acceso a ciertas partes de la aplicación según condiciones, como si el usuario está autenticado o tiene permisos específicos.

Implementación de Rutas Protegidas

¿Cómo proteger rutas?

Para proteger una ruta, puedes crear un componente que verifique si el usuario está autenticado antes de renderizar el contenido. Si no está autenticado, se redirige a otra página, como una de inicio de sesión.

Ejemplo de un componente **RutaProtegida**:

```
import React from 'react';

import { Navigate } from 'react-router-dom';



function RutaProtegida({ isAuthenticated, children }) {

  if (!isAuthenticated) {

    return <Navigate to="/login" replace />;

  }

  return children;

}

export default RutaProtegida;
```

Uso de rutas protegidas

En el archivo App.js, usa el componente **RutaProtegida** para proteger las rutas sensibles, como el carrito o la administración de productos.

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

import { Routes, Route } from 'react-router-dom';

import Home from './pages/Home';

import Productos from './pages/Productos';

import ProductoDetalle from './pages/ProductoDetalle';

import Carrito from './pages/Carrito';

import Admin from './pages/Admin';
```

```
import RutaProtegida from './components/RutaProtegida';

function App() {

  const [isAuthenticated, setIsAuthenticated] = useState(false);

  return (

    <div>

      <Routes>

        <Route path="/" element={<Home />} />

        <Route path="/productos" element={<Productos />} />

        <Route path="/productos/:id" element={<ProductoDetalle />} />

        <Route

          path="/carrito"

          element={

            <RutaProtegida isAuthenticated={isAuthenticated}>

              <Carrito />

            </RutaProtegida>

          }

        />

        <Route

          path="/admin"

          element={
```

```
            <RutaProtegida isAuthenticated={isAuthenticated}>
              <Admin />
            </RutaProtegida>
          }
        />
      </Routes>
    </div>
  );
}
export default App;
```

Redirección de Usuarios No Autenticados

El componente **Navigate** de React Router se utiliza para redirigir a los usuarios a otra página si no cumplen ciertas condiciones.

Ejemplo de redirección en una ruta protegida:

```
<Route
  path="/carrito"
  element={
    isAuthenticated ? <Carrito /> : <Navigate to="/login" replace />
  }
/>
```

En este caso, si el usuario no está autenticado (isAuthenticated = false), será redirigido a la página de inicio de sesión (/login).

TalentoLab - Proyecto final

Descripción de tu tarea:

¡Venís haciendo un excelente trabajo! Para tu próxima tarea necesitamos que implemente rutas dinámicas y protegidas para mejorar la seguridad y funcionalidad del eCommerce.

Tareas:

1. **Rutas Dinámicas:**
    - Crea una ruta dinámica para /productos/:id que muestre los detalles del producto seleccionado.
    - Usa el hook useParams para obtener el id del producto desde la URL.
    - Simula una base de datos de productos en un archivo o estado y muestra los detalles según el id.
2. **Rutas Protegidas:**
    - Implementa rutas protegidas para /carrito y /admin.
    - Redirige a los usuarios no autenticados a la página de inicio de sesión (/login).

3. **Interactividad:**
    - Crea un botón "Iniciar Sesión" que permita cambiar el estado de isAuthenticated para simular el inicio y cierre de sesión.
4. **Navbar:**
    - Agrega enlaces para navegar entre inicio, lista de productos, carrito y administración.
    - Usa el componente **Link** para los enlaces.

Reflexión final

En esta clase aprendimos a crear rutas dinámicas y protegidas en React Router para mejorar la funcionalidad y seguridad de nuestras aplicaciones. Exploramos cómo utilizar parámetros en las rutas dinámicas para mostrar contenido específico (como el detalle de un producto) y cómo proteger ciertas rutas para que solo sean accesibles por usuarios autenticados. Además, aprendimos a redirigir usuarios no autenticados para manejar la navegación de forma controlada.

Al dominar estas herramientas, podemos construir aplicaciones React más robustas, personalizadas y seguras, ofreciendo a los usuarios una experiencia fluida y mejorada. Este conocimiento es clave para desarrollar aplicaciones web modernas de una sola página (SPA), donde cada ruta tiene un propósito claro y contribuye a la lógica general de la aplicación.

Materiales y Recursos Adicionales:

★ Documentación oficial de React Router
★ React Router - Guía para Principiantes
★ Tutorial en video: "Rutas dinámicas y protegidas en React Router"

Preguntas para Reflexionar:

- ¿Qué diferencias encuentras entre las rutas estáticas y dinámicas?
- ¿Por qué es importante implementar rutas protegidas en una aplicación?

- ¿Qué ventajas tiene redirigir usuarios no autenticados a una página de inicio de sesión?
- ¿Cómo manejarías una aplicación con muchas rutas? ¿Qué técnicas podrías implementar para mantener el código organizado?

Próximos Pasos:

- Creación de Context API para manejo de estado global.
- Uso de useContext para compartir datos entre componentes.
- Implementación del estado global para el carrito.

Buenos Aires aprende

Agencia de Habilidades para el Futuro

Buenos Aires Ciudad