



Buenos  
Aires  
Ciudad

Agencia de Habilidades  
para el Futuro

<Talento  
Tech/>

# ReactJS

---

**Clase 07 | Rutas Dinámicas y Protegidas**

# ¡Les damos la bienvenida!



Vamos a comenzar a grabar la clase.

## Índice

---

### Router - Rutas Estáticas y Dinámicas

- Instalación y configuración de React Router.
- Creación de rutas estáticas (inicio, lista de productos).
- Uso de Link para navegar entre componentes.

### Rutas Dinámicas y Protegidas

- **Creación de rutas dinámicas (detalle de productos).**
- **Implementación de rutas protegidas (carrito, administración de productos).**
- **Redirección de usuarios no autenticados.**

### Introducción a Context API

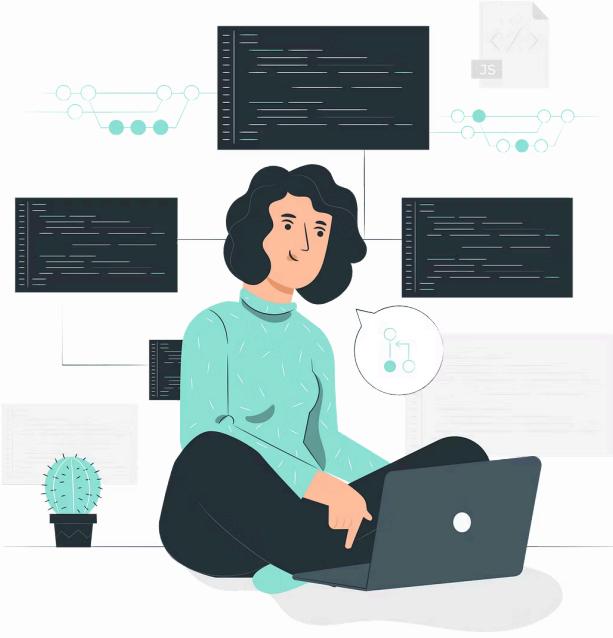
- Creación de Context API para el manejo de estado global.
- Uso de useContext para compartir datos entre componentes.
- Implementación del estado global para el carrito de compras.

# Objetivos de la Clase

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

# Rutas dinámicas





# ¿Qué son las Rutas Dinámicas?

---

Las rutas dinámicas contienen parámetros variables. Permiten mostrar contenido específico basado en la URL.

## Ejemplo

En un eCommerce, la ruta /productos/:id muestra información detallada de un producto específico.

# Beneficios de las Rutas Dinámicas

---

## 1 Flexibilidad

Permiten crear páginas con contenido variable basado en parámetros de URL.

## 2 Reutilización

Un solo componente puede manejar múltiples variaciones de contenido.

## 3 SEO

Mejora la indexación de páginas individuales en motores de búsqueda.

# Configuración de Rutas Dinámicas

```
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

---

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 específicas.

## Uso Común

Se utilizan para áreas que requieren autenticación o permisos especiales del usuario.

# Ventajas de las Rutas Protegidas



## Seguridad

Restringen el acceso a áreas sensibles de la aplicación.



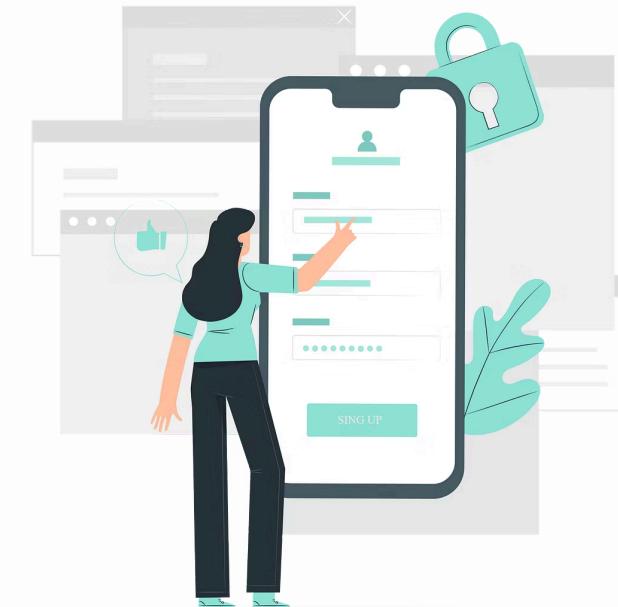
## Experiencia de Usuario

Dirigen a los usuarios a las áreas apropiadas según su estado de autenticación.



## Control de Acceso

Permiten implementar diferentes niveles de acceso para distintos tipos de usuarios.



# **Implementación de Rutas Protegidas**

---

# Implementación de Rutas Protegidas

---

## 1 Crear Componente

Desarrollar un componente que verifique la autenticación del usuario.

## 2 Verificar Autenticación

Comprobar si el usuario está autenticado antes de renderizar el contenido.

## 3 Redireccionar

Si no está autenticado, redirigir a la página 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;
```

# Breve Explicación

---

## 1 Componente Navigate

Se utiliza para redirigir usuarios a otra página si no cumplen ciertas condiciones.

## 2 Return children

Sirve para renderizar los componentes hijos que se envuelven dentro de este componente



# Uso de Rutas Protegidas

## Implementación

Usar el componente RutaProtegida para proteger rutas sensibles como el carrito o la administración.

## Ejemplo

Envolver componentes como Carrito o Admin con RutaProtegida en App.js.

```
<Route
  path="/carrito"
  element={ <RutaProtegida isAuthenticated={isAuthenticated}> <Carrito /> </RutaProtegida> }
/>

<Route
  path="/admin"
  element={ <RutaProtegida isAuthenticated={isAuthenticated}> <Admin /> </RutaProtegida> } />
```

# **Redirección de Usuarios No Autenticados**

---

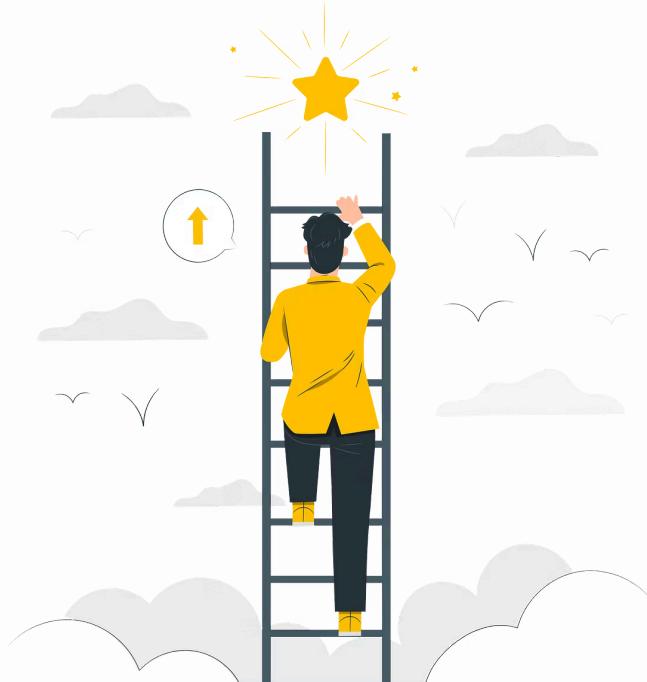
# Redirección de Usuarios No Autenticados

**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).



# Reflexión Final

---

En esta clase aprendimos a crear rutas dinámicas y protegidas en React Router. Exploramos cómo utilizar parámetros en las rutas dinámicas para mostrar contenido específico 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. .

# Preguntas para Reflexionar

---

## 1 Rutas Estáticas vs Dinámicas

¿Qué diferencias encuentras entre las rutas estáticas y dinámicas?

## 2 Importancia de Rutas Protegidas

¿Por qué es importante implementar rutas protegidas en una aplicación?



# Próximos Pasos



- 1 **Context API**  
Creación de Context API para manejo de estado global.
- 2 **useContext**  
Uso de useContext para compartir datos entre componentes.
- 3 **Estado Global**  
Implementación del estado global para el carrito.

# Materiales y Recursos Adicionales

---



## Documentación

[Documentación oficial de React Router](#)



## Guía

[React Router Guía para Principiantes](#)



## Tutorial

[Video: "Rutas dinámicas y protegidas en React Router"](#)



# Ejercicios Prácticos

---



**silvia**

Product Owner

# 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.



# Ejercicio Práctico

Obligatorio

## Requisitos:

### 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.

### Rutas Protegidas:

- Implementa rutas protegidas para /carrito y /admin.
- Redirige a los usuarios no autenticados a la página de inicio de sesión (/login).



# Ejercicio Práctico

Obligatorio

---

## Requisitos:

### Interactividad:

- Crea un botón "Iniciar Sesión" que permita cambiar el estado de isAuthenticated para simular el inicio y cierre de sesión.

### Navbar:

- Agrega enlaces para navegar entre inicio, lista de productos, carrito y administración.
- Usa el componente **Link** para los enlaces.