

## Hito 8 - Pizzería Mamma Mía

En este Hito validaremos nuestros conocimientos de React Router II. Para lograrlo, necesitarás aplicar los conocimientos adquiridos hasta el momento.

Lee todo el documento antes de comenzar el desarrollo **individual**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

// Tiempo asociado: 2 horas cronológicas

### Descripción

En este hito final vamos a implementar la autenticación real con el backend utilizando JWT.

Para lo cual consumiremos las siguientes rutas del backend:

#### Auth

```
JavaScript
POST /api/auth/login
POST /api/auth/register
```

body:

```
JavaScript
{
  "email": "test@example.com",
  "password": "123123"
}
```

#### Checkout & Profile

Esta ruta requiere un token JWT en el header, el token se obtiene en el endpoint de Auth explicado en el ítem siguiente (JWT).

Además puedes enviar un carrito con los productos a comprar, esto es solo una simulación, no se guarda en la base de datos.

```
JavaScript  
POST /api/checkouts
```

body:

```
JavaScript  
{  
  "cart": [...]  
}
```

Endpoint para obtener el perfil del usuario autenticado. Necesitas enviar el token JWT en el header.

```
JavaScript  
GET /api/auth/me
```

Para obtener el token JWT, se debe hacer una petición a `/api/auth/login` o a `/api/auth/register` con el body correspondiente.

El token JWT se debe enviar en el header Authorization de la siguiente manera:

```
JavaScript  
Authorization Bearer token_jwt
```

Ejemplo con fetch:

```
JavaScript  
await fetch("http://localhost:5000/api/checkout", {  
  method: "POST",  
  headers: {  
    "Content-Type": "application/json",  
    Authorization: `Bearer token_jwt`,  
  },  
  body: JSON.stringify({  
    cart: carrito,  
  }),  
});
```

## Requerimientos

1. En el UserContext, implementa los métodos para hacer login y register, consumiendo las rutas /api/auth/login y /api/auth/register respectivamente. Estas rutas te devolverán un token JWT y un email que debes almacenar sus respectivos estados. **(2 puntos)**
2. En el UserContext, implementa un método para hacer logout, este método debe eliminar el token y el email del estado. **(1 punto)**
3. En el UserContext, implementa un método para obtener el perfil del usuario autenticado, consumiendo la ruta /api/auth/me. **(1 punto)**
4. Tanto la página de Login como la de Register, deben implementar los métodos creados en UserContext para acceder al sistema. **(2 puntos)**
5. En la página profile, muestra el email del usuario autenticado y un botón para cerrar sesión. **(1 punto)**
6. El botón logout del navbar debe cerrar la sesión del usuario. **(1 punto)**
7. En la página de Cart.jsx, implementa el método para enviar el carrito de compras al backend, consumiendo la ruta /api/checkouts. **(1 punto)**
8. En la página de Cart.jsx, muestra un mensaje de éxito cuando se haya realizado la compra. **(1 punto)**



¡Mucho éxito!