



# RecipesAPI

## Entidades

### Recipe (Receta)

- `id` — Identificador único de la receta.
- `title` — Nombre de la receta.
- `description` — Descripción breve de la receta.
- `ingredients` — Lista de ingredientes necesarios para la receta.
- `category` — Categoría de la receta (por ejemplo, "postres", "ensaladas", "bebidas", etc.).
- `imageUrl` — URL de la imagen de la receta.
- `difficulty` — Nivel de dificultad ("fácil", "medio", "difícil").
- `likes` — Lista de IDs de los usuarios que han dado "like" a la receta.
- `creationDate` — Fecha de creación de la receta.

### Ejemplo de `Recipe` :

```
{
  id: String,          // Ejemplo: "60f0c5b4d10f4b4a5026e8c1"
  title: String,        // Ejemplo: "Tarta de manzana"
  description: String, // Ejemplo: "Una deliciosa tarta de manzana casera"
  ingredients: [String], // Ejemplo: ["manzanas", "azúcar", "harina"]
  category: String,    // Ejemplo: "Postres"
  imageUrl: String,     // Ejemplo: "https://example.com/tarta.jpg"
  difficulty: String,   // Ejemplo: "fácil"
  likes: [ObjectId],    // Ejemplo: ["60f0c5b4d10f4b4a5026e8c2", "60f0c5b4d10f4b4a5026e8c3"]
```

```
    creationDate: Date      // Ejemplo: "2021-07-15T14:30:00Z"
}
```

## User (Usuario)

- `id` — Identificador único del usuario.
- `name` — Nombre del usuario.
- `email` — Correo electrónico del usuario.
- `password` — Contraseña del usuario (almacenada encriptada).
- `role` — Rol del usuario ('admin' o 'user').
- `favoriteRecipes` — Array con los IDs de las recetas favoritas del usuario.

### Ejemplo de `User` :

```
{
  id: String,          // Ejemplo: "60f0c5b4d10f4b4a5026e8c2"
  name: String,        // Ejemplo: "Juan Pérez"
  email: String,       // Ejemplo: "juan.perez@example.com"
  password: String,    // Ejemplo: "$2b$10$5R5vVwFv1XfpXIm3gJ.oDuG
  qaCev5X/rRtxik0L1DqXZfiF7uFz9y" (Contraseña encriptada)
  role: String,         // Ejemplo: "user" o "admin"
  favoriteRecipes: [ObjectId] // Ejemplo: ["60f0c5b4d10f4b4a5026e8c1", "6
  0f0c5b4d10f4b4a5026e8c3"]
}
```

## EndPoints

### Sin Autenticación

- `GET /recipes`  
Obtener todas las recetas y en el campo like debe de devolver un numero
- `GET /recipes/:id`

Obtener el detalle de una receta específica por su ID

- `GET /recent_recipes`

Obtener las 5 recetas más recientes.

- `GET /popular_recipes`

Obtener las 5 recetas con mayor cantidad de likes.

- `POST /signup`

Registro de nuevos usuarios.

- `POST /login`

Inicio de sesión para usuarios existentes.

---

## Con Autenticación

- `GET /user/favorites`

Obtener las recetas favoritas del usuario logueado.

- `POST /user/:recipeld/favorite`

Añadir una receta a los favoritos del usuario logueado.

- `DELETE /user/:recipeld/favorite`

Eliminar una receta de los favoritos del usuario logueado.

- `POST /recipes`

Crear una nueva receta (solo para administradores).

- `PATCH /recipes/:id`

Actualizar una receta (solo para administradores).

- `DELETE /recipes/:id`

Eliminar una receta (solo para administradores).

- `POST /recipes/:id/comment`

Añadir un comentario a una receta (solo usuarios logueados).

- `PATCH /user`

Editar los datos del usuario logueado.

- `GET /generateToken`

Obtener el **ACCESS\_TOKEN** y el **REFRESH\_TOKEN**. Este endpoint utiliza el **REFRESH\_TOKEN** para generar nuevos tokens.

---

## Seguridad

- **Tokens** generados con **JWT**:
  - **ACCESS\_TOKEN** válido por 15 minutos.
  - **REFRESH\_TOKEN** válido por 60 minutos.
- **Contraseñas** de usuarios encriptadas usando **bcrypt**.
- **Middlewares de seguridad**:
  - Validación de tokens para garantizar que no han expirado.
  - Validación de rol para asegurar que un usuario tenga permisos de administrador cuando sea necesario.

## Ejercicios extras:

**1 - Correo de bienvenida:** Enviar un correo de bienvenida cuando un usuario se registre en la aplicación.

**2- Notificaciones de likes:** Sistema que notifica por correo electrónico al creador de una receta cuando recibe un like.

### 3- Likes para recetas:

- `POST /recipes/:id/like` — Añadir un like a una receta.
- `DELETE /recipes/:id/like` — Eliminar un like de una receta.

### 4- Filtrar por categoría:

`GET /recipes/category` — Obtener recetas filtradas por categoría (por ejemplo: "Postres", "Bebidas", etc.).

### 5- Sistema de comentarios:

Añadir un nuevo atributo al modelo `Recipe` para manejar un array de comentarios con el siguiente modelo:

- `comment : String` — El texto del comentario.
- `userId : ObjectId` — ID del usuario que realiza el comentario.

- `date` : `Date` — Fecha de creación del comentario.
  - `rating` : `Number` — Puntuación de la receta (del 1 al 5).
- 

## Modelo de Comentarios (Adicional al modelo de Recipe )

```
{  
  comment: String,      // Ejemplo: "¡Me encantó esta receta!"  
  userId: ObjectId,    // Ejemplo: "60f0c5b4d10f4b4a5026e8c2"  
  date: Date,          // Ejemplo: "2021-07-15T14:45:00Z"  
  rating: Number       // Ejemplo: 5  
}
```

Este modelo de comentarios es parte del atributo `comments` dentro de cada receta.