**🧱 Requisitos técnicos**

1. **Conexión a MongoDB Atlas usando Mongoose.**
2. **Módulo users con las operaciones CRUD (crear, leer, actualizar, eliminar).**
3. **Módulo auth con registro, login y cierre de sesión.**
4. **Uso de bcrypt para hashear contraseñas antes de guardarlas.**
5. **Validaciones con class-validator y class-transformer.**
6. **Protección de rutas con JWT (solo usuarios autenticados pueden modificar o eliminar su cuenta).**
7. **Pruebas de endpoints en Postman o Insomnia.**

**🧾 Campos a guardar en la colección users**

| **Campo** | **Tipo de dato** | **Validación requerida** |
| --- | --- | --- |
| name | string | Obligatorio. Mínimo 3 caracteres. |
| email | string | Obligatorio. Formato de correo válido. Único en la base. |
| password | string | Obligatorio. Mínimo 6 caracteres. Se guarda hasheada con bcrypt. |
| createdAt | date | Se genera automáticamente al crear el usuario. |
| role (opcional) | string | Valor por defecto: "user". |

**🔐 Flujo de autenticación**

**1️⃣ Registro (POST /auth/register)**

* **Recibe name, email, password.**
* **Valida los datos.**
* **Hashea la contraseña antes de guardar.**
* **Devuelve un mensaje de éxito o el usuario creado (sin contraseña).**

**2️⃣ Inicio de sesión (POST /auth/login)**

* **Verifica que el email exista y la contraseña coincida.**
* **Devuelve un token JWT válido y los datos del usuario.**

**3️⃣ Cierre de sesión (POST /auth/logout)**

* **Invalida el token (podés usar una lista negra temporal o eliminarlo del frontend).**
* **Devuelve confirmación del cierre.**

**👤 Gestión de usuarios**

**4️⃣ Modificar datos (PUT /users/:id)**

* **Solo accesible con un JWT válido.**
* **Permite editar name, email o password.**
* **Si se cambia la contraseña, debe volver a hashearse.**
* **Actualiza el campo updatedAt.**

**5️⃣ Eliminar usuario (DELETE /users/:id)**

* **Solo accesible con un JWT válido.**
* **Elimina al usuario autenticado o, si es admin, a cualquier usuario.**

**⚙️ Validaciones importantes**

* **No permitir registro con email duplicado.**
* **En el login, comparar contraseñas con bcrypt.compare().**
* **Proteger rutas con @UseGuards(AuthGuard('jwt')).**
* **Ocultar el campo password al devolver datos.**

**🚀 Entrega**

* **Subí el proyecto completo a GitHub.**
* **Incluí en el README:**
  + **Instrucciones para ejecutar el proyecto.**
  + **Ejemplos del body para cada endpoint (registro, login, logout, update, delete).**
  + **Capturas de las pruebas en Postman o Insomnia.**