

# Documentación de la API REST - Generador de Listas Etiquetadas

## Portada

**Universidad:** ITESO  
**Proyecto:** Generador de Listas Etiquetadas  
**Materia:** Desarrollo de Aplicaciones y Servicios Web  
**Integrantes:**

- Federico Humberto Zaragoza Manuel
- Luis Eduardo González Gloria
- Diego Mercado Coello

**Fecha:** Marzo 2024

## Descripción del Proyecto

El "Generador de Listas Etiquetadas" es una aplicación web que permite a los usuarios crear y gestionar listas personalizadas. Cada lista contiene ítems que pueden ser etiquetados y tener diferentes estados (activo, inactivo o pendiente). La aplicación ofrece funcionalidades avanzadas como filtrado por etiquetas, campos personalizados, vistas personalizadas, y un sistema de autenticación robusto mediante JWT.

## Documentación de la API REST

### Tabla de Endpoints

Ruta	Método	Header	Body
/api/auth/register	POST	Content-Type: application/json	{ "username": "string", "email": "string", "password": "string", "nombre": "string", "fecha_nacimiento": "YYYY-MM-DD" }
/api/auth/login	POST	Content-Type: application/json	{ "email": "string",

Ruta	Método	Header	Body
			<pre>"password": "string" }</pre>
/api/auth/profile	GET	Authorization: Bearer {token}	-
/api/auth/profile	PUT	Authorization: Bearer {token}	<pre>{ "nombre": "string", "fecha_nacimiento": "YYYY-MM-DD" }</pre>
/api/admin/users	GET	Authorization: Bearer {token}	-
/api/admin/users/{userId}	DELETE	Authorization: Bearer {token}	-
/api/lists	GET	Authorization: Bearer {token}	-
/api/lists	POST	Authorization: Bearer {token}	<pre>{ "title": "string", "description": "string",</pre>

Ruta	Método	Header	Body
			"camposEspeciales": { } }
/api/lists/{listId}	GET	Authorization: Bearer {token}	-
/api/lists/{listId}	PUT	Authorization: Bearer {token}	{ "title": "string", "description": "string", "camposEspeciales": { } }
/api/lists/{listId}	DELETE	Authorization: Bearer {token}	-
/api/lists/{listId}/items	GET	Authorization: Bearer {token}	-
/api/lists/{listId}/items	POST	Authorization: Bearer {token}	{ "title": "string", "description": "string", "estado": "string", "etiquetas": ["string"], "imagenUrl": "string" }
/api/lists/{listId}/items/{itemId}	PUT	Authorization: Bearer {token}	{ "title": "string", "description": "string", "estado": "string", "etiquetas": ["string"],

Ruta	Método	Header	Body
			<code>"imageUrl": "string" }</code>
<code>/api/lists/{listId}/items/{itemId}</code>	DELETE	Authorization: Bearer {token}	-
<code>/api/tags</code>	GET	Authorization: Bearer {token}	-
<code>/api/tags</code>	POST	Authorization: Bearer {token}	<code>{ "nombre": "string", "color": "string" }</code>
<code>/api/views</code>	GET	Authorization: Bearer {token}	-
<code>/api/views</code>	POST	Authorization: Bearer {token}	<code>{ "nombre": "string", "filtros": {}, "ordenamientos": {} }</code>

## Conclusión

La API REST diseñada cumple con todos los requerimientos especificados en el documento original:

1. **Cobertura Completa:** La API proporciona endpoints para todas las funcionalidades requeridas:
  - Autenticación y gestión de usuarios
  - CRUD completo de listas
  - Gestión de ítems con etiquetas
  - Sistema de vistas personalizadas
  - Gestión de etiquetas con estadísticas
2. **Seguridad:**

- Implementación de JWT para autenticación
- Protección de rutas mediante tokens
- Validación de roles para acciones administrativas

### 3. **Soporte para el Front-end:**

- Endpoints diseñados para soportar todas las funcionalidades de UI
- Soporte para paginación y filtrado
- Respuestas estructuradas para fácil integración

### 4. **Escalabilidad:**

- Estructura modular que permite agregar nuevos endpoints
- Soporte para futuras características como compartir listas
- Sistema de vistas personalizable

La API proporciona una base sólida para el desarrollo del front-end y cumple con los estándares REST, incluyendo:

- Uso apropiado de métodos HTTP
- Códigos de estado significativos
- Rutas semánticas y consistentes
- Documentación clara de parámetros y respuestas

Se recomienda implementar esta API utilizando Swagger/OpenAPI para documentación interactiva, lo que facilitará las pruebas y la integración con el front-end.