

# Informe API REST - Generador de Listas Etiquetadas

## Tabla de Endpoints de la API

A continuación se presenta la tabla actualizada de los endpoints implementados en la API REST.

### Autenticación ( /api/auth )

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Respuesta Exitosa (Ejemplo)	Código
POST	/register	Registrar nuevo usuario	Público	Body: { username , email , password , fullName (nombre), birthDate (fecha_nacimiento), role (opcional) }	{ success:true, user:{...}, token }	401 falla va us ex
POST	/login	Iniciar sesión	Público	Body: { email , password }	{ success:true, user:{...}, token }	401 (cr inv 50
GET	/profile	Obtener perfil de usuario actual	Privado	-	{ success:true, user:{...} }	401
PUT	/profile	Actualizar perfil de usuario actual	Privado	Body: { nombre , fecha_nacimiento , email (opcional) }	{ success:true, user:{...} }	401 (va en us 50
PUT	/updatepassword	Cambiar contraseña (actual)	Privado	Body: { currentPassword , newPassword }	{ success:true, message }	401 (cc ac inc 40 da

### Usuarios ( /api/users )

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Resp Exit (Ej)
GET	/	Listar usuarios (paginado)	Privado/Admin	Query: <code>page</code> , <code>limit</code>	<code>{ success count, paginat {...}, [...] }</code>
GET	/stats	Estadísticas de usuarios	Privado/Admin	-	<code>{ success stats:{</code>
GET	/:id	Obtener usuario por ID	Privado/Admin	Params: <code>id</code>	<code>{ success user:{.</code>
PUT	/:id	Actualizar un usuario (admin)	Privado/Admin	Params: <code>id</code> , Body: { <code>nombre</code> , <code>fecha_nacimiento</code> }	<code>{ success user:{.</code>
PUT	/:id/role	Cambiar rol de usuario	Privado/Admin	Params: <code>id</code> , Body: { <code>role</code> }	<code>{ success user:{.</code>
DELETE	/:id	Eliminar usuario	Privado/Admin	Params: <code>id</code>	<code>{ success message</code>
PUT	/me/lists/favorite/:listId	Toggle lista favorita (actual)	Privado	Params: <code>listId</code>	<code>{ success message isFavor user:{.</code>
PUT	/me/lists/archive/:listId	Toggle lista archivada (actual)	Privado	Params: <code>listId</code>	<code>{ success message isArchi user:{.</code>

## Listas ( `/api/listas` )

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Re: E (Ej)
GET	/	Listar listas del usuario (paginado)	Privado	Query: <code>page</code> , <code>limit</code>	<code>{ succe count pagin</code>

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Respuesta (Ejemplo)
					{...} [...]
POST	/	Crear nueva lista	Privado	Body: { title , description (opc), camposEspaciales (opc) }	{ success: message: listId: items:
GET	/:id	Obtener lista específica con sus ítems	Privado	Params: id	{ success: listId: items:
PUT	/:id	Actualizar lista	Privado	Params: id , Body: { title , description , camposEspaciales }	{ success: message: listId:
DELETE	/:id	Eliminar lista	Privado	Params: id	{ success: message:
POST	/:listId/share	Compartir lista con colaborador	Privado	Params: listId , Body: { emailOrUsername , permiso }	{ success: listId:
DELETE	/:listId/share/:collaboratorUserId	Quitar colaborador de una lista	Privado	Params: listId , collaboratorUserId	{ success: listId:

## Ítems ( /api/items )

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Respuesta Exitosa (Ejemplo)	Error Común
GET	/filtrar	Filtrar ítems por estado, etiquetas	Privado	Query: listId (req), estado (opc), etiquetas (opc, CSV), etiqueta_logica (opc), sortField , sortDir , page , limit	{ success:true, count, pagination: {...}, data: [...]} 400, 403, 500	400, 403, 500
POST	/	Crear nuevo ítem	Privado	Body: { listId , title , description , estado , etiquetas (array IDs), imagenUrl ,	{ success:true, message, item:{...}} 400, 403, 500	400, 403, 500

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Respuesta Exitosa (Ejemplo)	Error Comun
				valoresCamposEspaciales }		
GET	/:id	Obtener Ítem específico	Privado	Params: id	{ success:true, item:{...} }	401, 404, 500
PUT	/:id	Actualizar Ítem	Privado	Params: id , Body: { (campos a actualizar) }	{ success:true, message, item:{...} }	400, 403, 404, 500
DELETE	/:id	Eliminar Ítem	Privado	Params: id	{ success:true, message }	401, 404, 500

## Etiquetas ( /api/etiquetas )

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Respuesta Exitosa (Ejemplo)	Error Comun
GET	/	Listar etiquetas (paginado)	Privado	Query: <code>page</code> , <code>limit</code>	<pre>{   success:true,   count,   pagination:   {...}, data:   [...] }</pre>	401, 500
POST	/	Crear nueva etiqueta	Privado	Body: { <code>nombre</code> , <code>color</code> (opc) }	<pre>{   success:true,   message,   etiqueta:   {...} }</pre>	400 (duplica 401, 500
PUT	/:id	Actualizar etiqueta	Privado	Params: <code>id</code> , Body: { <code>nombre</code> , <code>color</code> }	<pre>{   success:true,   message,   etiqueta:   {...} }</pre>	400, 40 403, 40 500
DELETE	/:id	Eliminar etiqueta	Privado	Params: <code>id</code>	<pre>{   success:true,   message }</pre>	401, 40 404, 500
GET	/estadisticas	Estadísticas de uso de etiquetas (mock)	Privado	-	<pre>{   success:true,   estadisticas:   [...] }</pre>	401, 500

## Vistas Personalizadas ( /api/vistas )

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Respuesta Exitosa (Ejemplo)	Errores Comunes
GET	/	Listar vistas del usuario (paginado)	Privado	Query: <code>page</code> , <code>limit</code>	<pre>{   success:true,   count,   pagination:   {...}, data:   [...] }</pre>	401, 500

Método	Ruta	Descripción	Acceso	Parámetros (Body/Params/Query)	Respuesta Exitosa (Ejemplo)	Errores Comunes
POST	/	Crear nueva vista	Privado	Body: { nombre , filtros , ordenamientos }	{ success:true, message, vista:{...} }	400 (duplicado), 401, 500
GET	/:id	Obtener vista específica	Privado	Params: id	{ success:true, vista:{...} }	401, 403, 404, 500
PUT	/:id	Actualizar vista	Privado	Params: id , Body: { (campos a actualizar) }	{ success:true, message, vista:{...} }	400, 401, 403, 404, 500
DELETE	/:id	Eliminar vista	Privado	Params: id	{ success:true, message }	401, 403, 404, 500

## Enlace al Video de Demostración

A continuación, se presenta el video de demostración del flujo principal de la API REST:

[Ver Video de Demostración de la API](#)

[https://iteso01-my.sharepoint.com/:v/g/personal/diego\\_mercadoc\\_iteso\\_mx/ESUwkdmV3plmO-zpm1IziQBooaSbmrs-tkWWL4V\\_0AE9g?e=BVPUs9](https://iteso01-my.sharepoint.com/:v/g/personal/diego_mercadoc_iteso_mx/ESUwkdmV3plmO-zpm1IziQBooaSbmrs-tkWWL4V_0AE9g?e=BVPUs9)

## Funcionalidades Pendientes

Aunque la API cubre la mayoría de los requerimientos CRUD y de autenticación, algunas funcionalidades avanzadas o mejoras de UI/UX aún están pendientes:

- **Mejoras UI/UX:**
  - Implementación de Drag & Drop (y quizás listas).
  - Pulido visual general de la interfaz.
- **Despliegue:**
  - Configuración de dominio personalizado.