

Documento: Requerimientos Generales del Proyecto "Generador de Listas Etiquetadas"

(Versión integrada con base en los documentos de requerimientos de front-end, back-end y el borrador inicial **y actualizada según mockups visuales**)

1. Descripción General del Proyecto

Nuestro proyecto, denominado "Generador de Listas Etiquetadas", consiste en el desarrollo de una aplicación web que permita a los usuarios crear y gestionar listas. Cada lista contiene ítems, y cada ítem puede tener múltiples etiquetas y un estado (activo, inactivo o pendiente). Además, el usuario podrá:

- Filtrar los ítems de acuerdo con las etiquetas y estados.
- Agregar campos especiales (o campos personalizados) en cada lista.
- Gestionar vistas personalizadas por filtros u ordenamientos.
- Autenticarse en el sistema (Login/Register) y autorización mediante JWT.
- Disponer de funcionalidades adicionales, como marcar listas como favoritas, archivarlas o compartirlas (~~extensible en iteraciones posteriores~~). **(La funcionalidad de compartir no se visualizó en los mockups actuales, pero se mantiene como posible extensión futura).**

El objetivo es proporcionar una herramienta flexible y personalizable para la organización de información a través de listas, facilitando la categorización de sus elementos mediante etiquetas y estados, y permitiendo una búsqueda/filtrado eficiente. **Las interfaces clave, como el dashboard, editor de ítems, gestor de etiquetas y configuración de vistas personalizadas, han sido visualizadas mediante mockups detallados para guiar el desarrollo.**

2. Entidades en la Base de Datos

Para cubrir la funcionalidad descrita, se ha optado por usar MongoDB. Las principales entidades que se manejarán son las siguientes:

1. Usuarios
2. Listas
3. Ítems
4. Etiquetas **(Confirmada como entidad separada por el Gestor de Etiquetas)**
5. Relaciones Ítem-Etiquetas (~~en caso de optar por un modelo muchos-a-muchos explícito~~)
(Se manejará la relación directamente en el Ítem o mediante la entidad Etiqueta)

3. Atributos de Cada Entidad

A continuación, se describen los atributos principales de cada entidad. Cabe señalar que, en su mayoría, se prevé usar ObjectId como referencia a la base de datos de MongoDB, y se incluyen algunos campos obligatorios y otros opcionales.

3.1 Usuarios

- **username** (String, requerido, único). Ej: "juanp"
- **email** (String, requerido, único). Ej: "juanp@gmail.com"
- **nombre** (String). Ej: "Juan Pérez López"
- **password** (String, en formato de hash).
- **fecha_registro** (String de fecha en formato año-mes-día) o tipo Date. Ej: "2025-02-25"
- **fecha_nacimiento** (String de fecha año-mes-día) o tipo Date (opcional).
- **listasFavoritas** (Array de ObjectId, referencia a Listas). (**Visualizado en Sidebar del Dashboard**)
- **listasArchivadas** (Array de ObjectId, referencia a Listas). (**Visualizado en Sidebar del Dashboard**)
- **createdAt** (Date, default: Date.now).

3.2 Listas

- **userId** (ObjectId, referencia a Usuarios, requerido).
- **title** (String, requerido). Ej: "Tareas Pendientes"
- **description** (String, opcional).
- **camposEspaciales** (Objeto) o campos personalizados que la lista requiera. (**Define la estructura de campos como 'Fecha Límite' o 'Prioridad' visualizados en el mockup del Item Editor**).
- **createdAt** (Date, default: Date.now).
- **updatedAt** (Date, default: Date.now).

3.3 Ítems

- **listId** (ObjectId, referencia a Listas, requerido).
- **title** (String, requerido). Ej: "Comprar leche"
- **description** (String, opcional).
- **estado** (String, enum: ['activo', 'inactivo', 'pendiente'], default: 'pendiente'). (**Visualizado con selectores y badges en mockups**)
- **etiquetas** (Array of ObjectId, referencia a Etiquetas). ~~o (relación muchos-a-muchos con entidad Etiqueta).~~ Ej: ["compras", "urgente"] (**Se opta por referencias a la**

entidad Etiqueta para facilitar la gestión centralizada).

- **imageUrl** (String, opcional). **(Visualizado en Item Editor)**
- **valoresCamposEspaciales** (Objeto, opcional). **(Almacenará los valores específicos de este ítem para los camposEspaciales definidos en su Lista padre. Ej: { fecha_limite: '2024-12-31', prioridad: 'alta' })**
- **createdAt** (Date, default: Date.now).
- **updatedAt** (Date, default: Date.now).

3.4 Etiquetas

- **nombre** (String, único por usuario). **(Visualizado y gestionado en Tag Manager)**
- **userId** (ObjectId, referencia a Usuario).
- **color** (String, opcional). **(Para representación visual, como se ve en el modal de Nueva Etiqueta)**
- **createdAt** (Date, default: Date.now). **(Añadido para trazabilidad)**

3.5 Relaciones Ítem-Etiquetas

(Sólo si se implementa un modelo aparte de muchos a muchos.)

- **itemId** (ObjectId, referencia a Ítem).
- **tagId** (ObjectId, referencia a Etiqueta).

3.6 Vistas Personalizadas

- **userId** (ObjectId, referencia a Usuario).
- **nombre** (String). **(Input visible en mockup)**
- **filtros** (Objeto con criterios de filtrado, ej: { estado: 'activo', etiquetas: ['id_tag1', 'id_tag2'], etiqueta_logica: 'all' }). **(Configurable en mockup)**
- **ordenamientos** (Objeto con criterios de ordenamiento, ej: { campo: 'createdAt', direccion: 'desc' }). **(Configurable en mockup)**
- **createdAt** (Date, default: Date.now). **(Añadido para trazabilidad)**

4. Tipos de Usuarios en la Aplicación

Se contemplan al menos dos tipos de usuarios:

1. Administrador:

- Posee privilegios para gestionar a todos los usuarios, cambiar roles y tener acceso a funcionalidades internas de mantenimiento.
- Puede realizar todas las acciones de un usuario común, además de las de administración. **(No se crearon mockups específicos para vistas de admin).**

2. Usuario Común:

- Rol principal para los usuarios finales.
- Puede crear, editar y eliminar sus listas e ítems, así como gestionar sus etiquetas.
(Flujo principal visualizado en los mockups).

5. Operaciones Disponibles (por Rol)

5.1 Administrador

- **Gestión de Usuarios:**
 - Consultar usuarios registrados.
 - Eliminar usuarios o cambiarles el rol a administrador.
 - Puede realizar las mismas operaciones que un usuario común (login, logout, etc.).
- **Gestión de Listas / Ítems (Interno):**
 - Crear, editar, eliminar cualquier lista o ítem (por motivos de mantenimiento o soporte).
- **Revisión General / Mantenimiento:**
 - Puede acceder a reportes o estadísticas generales (p. ej. número total de listas creadas).

5.2 Usuario Común

- **Usuarios (Perfil):**
 - Registro (signup). **(Visualizado en `login.html`)**
 - Login (inicio de sesión). **(Visualizado en `login.html`)**
 - Logout (cierre de sesión). **(Opción en dropdown de usuario en navbar)**
 - Editar datos básicos de su perfil (por ejemplo, cambiar contraseña). **(Opción en dropdown de usuario, aunque la pantalla específica no se diseñó).**
- **Listas:**
 - Crear listas. **(Botón en sidebar)**
 - Consultar (ver) sus propias listas. **(Dashboard `index.html`)**
 - Editar / actualizar listas. **(Implícito, aunque no se diseñó un modal específico para editar lista)**
 - Eliminar listas. **(Implícito)**
 - Marcar / desmarcar listas como favoritas o archivadas. **(Secciones en sidebar).**
- **Ítems:**
 - Crear ítems dentro de una lista. **(Botón "Nuevo Item" en `index.html` que abriría `item-editor.html`)**
 - Consultar ítems, filtrarlos por estado o etiquetas. **(Dashboard `index.html` con filtros de estado).**
 - Editar / actualizar ítems. **(Modal `item-editor.html`)**
 - Eliminar ítems. **(Implícito, se añadiría botón en item card o editor).**
- **Etiquetas:**

- **Gestionar Etiquetas:** Crear, editar, eliminar y visualizar etiquetas, incluyendo asignación de color y ver estadísticas de uso. (**Visualizado en** `tag-manager.html`).
- **Vistas Personalizadas:**
 - **Configurar Vistas Personalizadas:** Definir nombre, filtros (por estado, etiquetas) y ordenamientos para guardar vistas reutilizables. (**Visualizado en** `custom-view.html`).

6. Requerimientos Adicionales de Back-end

Además de las funcionalidades CRUD principales, se contempla:

- **Autenticación y Autorización de Usuarios:**
 - Uso de JWT (JSON Web Tokens) para proteger las rutas que requieran autenticación.
- **Validación de Datos:**
 - Se validará la información en el back-end (por ejemplo, que "title" no venga vacío al crear listas o ítems, formato de email, etc.).
- **Manejo de Errores Centralizado:**
 - Enviar respuestas consistentes al front-end y facilitar la depuración.
- **Gestión de Imágenes o Archivos (opcional):**
 - Se permite asociar una "imageUrl" a un ítem, **el backend deberá gestionar la subida (si aplica) o simplemente almacenar la URL proporcionada.**
- **Endpoints Específicos para Mockups:**
 - **Endpoint para obtener estadísticas de etiquetas (total, más usada, sin usar)** para `tag-manager.html` .
 - **Endpoints para CRUD de** `VistasPersonalizadas` .
 - **Endpoints para marcar/desmarcar listas como favoritas/archivadas.**

7. Información Precargada y Trato Especial a la Información

En la versión actual (MVP), no se requiere precargar información más allá de un usuario administrador inicial o de prueba. Las etiquetas se crearán y gestionarán a través del `tag-manager.html` . Los campos especiales se definen a nivel de lista.

8. Distribución de Actividades en el Equipo

La distribución se enfoca en funcionalidades clave con interacción directa con la base de datos, basadas en las entidades y mockups:

1. **Federico Humberto Zaragoza Manuel:**

- **Usuarios:** Registro, Login/Logout (implementación JWT), Edición de perfil de usuario, ~~control de roles y accesos~~ **Administración de usuarios (CRUD básico y cambio de rol para administradores)**.
- **Core DB:** Entidad `Usuarios` (CRUD, autenticación).

2. Luis Eduardo González Gloria:

- **Listas:** Creación, edición, eliminación y consulta de Listas. Implementación de lógica para marcar/desmarcar listas como `listasFavoritas` y `listasArchivadas` (actualizando la entidad `Usuarios`).
- **Dashboard:** Lógica de backend para obtener y mostrar las listas del usuario en el dashboard (`index.html`).
- **Core DB:** Entidad `Listas` (CRUD), actualización de arrays de referencia en `Usuarios` .

3. Diego Mercado Coello:

- **Ítems:** Creación, edición, eliminación y consulta de Ítems dentro de una lista, incluyendo la gestión de `estado` , `imagenUrl` y `valoresCamposEspaciales` .
- **Etiquetas:** Creación, edición, eliminación y consulta de Etiquetas (incluyendo `color`). Lógica para asociar/desasociar etiquetas a ítems (actualizando array de referencias en `Ítems`). Endpoint para estadísticas de etiquetas.
- **Vistas Personalizadas:** Creación, edición, eliminación y consulta de Vistas Personalizadas. Lógica para aplicar filtros y ordenamientos guardados al consultar ítems.
- **Core DB:** Entidades `Ítems` , `Etiquetas` , `VistasPersonalizadas` (CRUD completo y relaciones).

9. Flujo de Información (MVP)

El flujo básico de la aplicación es:

Usuario Común:

1. Registro y Login: El usuario se registra (o inicia sesión si ya existe).
2. Visualización de Listas: Tras autenticarse, ve su panel principal (`index.html`) con todas sus listas, **incluyendo acceso a Favoritas y Archivadas en el sidebar**.
3. Creación de Lista: El usuario puede crear nuevas listas (**acción desde el sidebar**).
4. Detalle de Lista: Al hacer clic en una lista (**o al estar en el dashboard**), ve los ítems que contiene.
5. Gestión de Items:
 - Puede crear nuevos ítems (**abre modal** `item-editor.html`), indicando título, descripción, estado, etiquetas, imagen y campos especiales si aplica.
 - Puede editar ítems existentes (**abre modal** `item-editor.html`).
 - Eliminar ítems existentes.
 - Aplicar filtros u ordenamientos (**controles básicos en** `index.html` , **avanzados en** `custom-view.html`).

6. **Gestión de Etiquetas:**
 - Acceder al `tag-manager.html` para crear, editar, eliminar etiquetas y ver estadísticas.
7. **Gestión de Vistas Personalizadas:**
 - Acceder a `custom-view.html` para crear, guardar y aplicar vistas con filtros/ordenamientos complejos.
8. Opciones Adicionales:
 - Marcar listas como favoritas o archivadas (**interactuando con el sidebar**).
9. Logout: Finaliza la sesión (**desde navbar**).

Administrador:

1. Login de Administrador: Ingresa con credenciales de administrador.
2. Gestión de Usuarios: Visualiza la lista de usuarios, puede eliminarlos o cambiarles el rol.
3. Gestión Global (opcional): Puede ver el estado general de las listas o dar soporte a usuarios.
4. Acceso a Funciones de Usuario Común: También puede gestionar sus propias listas o ítems.

10. Tabla de Ventanas/Páginas Principales

Se muestra a continuación una tabla simplificada de las pantallas más relevantes **basada en los mockups HTML creados**:

Ventana/Componente	Mockup HTML	Recurso (BD)	Usuarios con Acceso	Descripción Breve	Responsable
Menú de Navegación (Navbar)	(en todas las págs)	(datos de sesión)	Usuarios Logueados	Barra superior con logo, búsqueda global (a definir alcance) , y menú de usuario (perfil, config, logout).	Feder
Página de Login/Registro	<code>login.html</code>	Usuarios	Todos los usuarios	Formulario de inicio de sesión y botón/link para registrarse.	Feder
Página de Registro	(no separada)	Usuarios	Todos los usuarios	Formulario de registro de nuevo usuario con validaciones: (Integrado en <code>login.html</code> o modal)	Feder
Dashboard (Vista Principal)	<code>index.html</code>	Listas, Ítems, Etiquetas	Usuarios Logueados	Muestra listado de ítems (de una	Luis E

Ventana/Componente	Mockup HTML	Recurso (BD)	Usuarios con Acceso	Descripción Breve	Respo
				lista activa o vista), sidebar (Listas Fav/Arch, Nueva Lista, Etiquetas), filtros rápidos (estado) , botón Nuevo Item.	
Editor de Ítem (Modal)	item-editor.html	Ítems, Etiquetas	Usuarios Logueados	Modal para crear/editar ítems con campos: título, desc, estado, etiquetas, imagen, campos especiales .	Diego Mercader
Detalle de Lista	(integrado)	Ítems	Usuarios Logueados	Muestra y gestiona los ítems de la lista seleccionada (crear, filtrar, editar, eliminar). (Funcionalidad integrada en Dashboard/Item Editor)	Diego Mercader
Gestor de Etiquetas	tag-manager.html	Etiquetas	Usuarios Logueados	Página para crear, editar, eliminar, buscar etiquetas. Muestra lista de etiquetas con contador de uso y color . Incluye modal para nueva/editar etiqueta y estadísticas .	Diego Mercader
Configuración Vista Personalizada	custom-view.html	Vistas Personalizadas	Usuarios Logueados	Página para definir nombre, filtros avanzados (estado, etiquetas con lógica Y/O) y ordenamiento . Incluye vista previa y botón para guardar.	Diego Mercader

Ventana/Componente	Mockup HTML	Recurso (BD)	Usuarios con Acceso	Descripción Breve	Respo
Página de Perfil	(no diseñada)	Usuarios	Usuario Logueado	Permite modificar datos personales (cambiar contraseña, actualizar email, etc.).	Feder
Administración de Usuarios	(no diseñada)	Usuarios	Administrador	Muestra la lista de usuarios, con opciones de eliminar usuario o cambiar su rol.	Feder

11. Diferenciadores del Proyecto

Una posible característica diferenciadora podría ser la incorporación de representaciones gráficas (p. ej. nubes de palabras) para facilitar la selección y visualización de etiquetas. **Los mockups actuales establecen como diferenciadores:**

- **Flexibilidad en la creación de listas con "campos espaciales" o campos personalizados**, permitiendo que cada lista tenga atributos dinámicos definidos por el usuario (**visualizado en `item-editor.html`**).
- **Un Gestor de Etiquetas centralizado (`tag-manager.html`) con asignación de colores y estadísticas de uso.**
- **Un constructor visual para Vistas Personalizadas (`custom-view.html`) que permite definir filtros y ordenamientos complejos con vista previa.**

12. Tecnologías a Utilizar

- **Front-end:** HTML, CSS, JavaScript (consumiendo la API mediante fetch o Axios).
Frameworks/Librerías: **Bootstrap 5**, **Font Awesome 6** (utilizados en los mockups).
- **Back-end:** Node.js con Express.
- **Base de Datos:** MongoDB.
- **Autenticación:** JWT (JSON Web Tokens).
- **Control de Versiones:** Git (GitHub o similar).
- **Despliegue:** Heroku, Render u otro servicio de hosting en la nube.

13. Cambios Realizados (Basados en Mockups)

- **Descripción General:** Se añadió mención a los mockups creados. Se matizó la funcionalidad de compartir.

- **Entidades:** Se confirmó `Etiquetas` y `VistasPersonalizadas` como entidades separadas. Se eliminó la entidad explícita `Relaciones Ítem-Etiquetas`.
- **Atributos:**
 - Se añadió referencia a la visualización de `listasFavoritas` / `Archivadas` en `Usuarios`.
 - Se clarificó el rol de `camposEspaciales` en `Listas` (definición) y se añadió `valoresCamposEspaciales` en `Ítems` (almacenamiento de valor).
 - Se cambió `etiquetas` en `Ítems` a un array of `ObjectId` referenciando `Etiquetas`.
 - Se añadió `color` y `createdAt` a `Etiquetas`.
 - Se añadió `createdAt` a `VistasPersonalizadas` y se detalló la estructura de `filtros` y `ordenamientos`.
- **Operaciones Disponibles:** Se detallaron las operaciones para gestionar Etiquetas y Vistas Personalizadas basadas en los mockups.
- **Requerimientos Back-end:** Se añadieron endpoints específicos necesarios para las funcionalidades visualizadas (stats de etiquetas, CRUD VistasPersonalizadas, fav/archivadas). Se especificó el manejo de `imagenUrl`.
- **Distribución de Actividades:** Se reasignaron responsabilidades a Federico, Luis Eduardo y Diego enfocándose en entidades y funcionalidades de BD específicas derivadas de los mockups.
- **Flujo de Información:** Se incorporaron los pasos de gestión de Etiquetas y Vistas Personalizadas.
- **Tabla de Ventanas:** Se actualizó la tabla para reflejar los mockups HTML (`index.html`, `login.html`, `item-editor.html`, `tag-manager.html`, `custom-view.html`). Se ajustaron descripciones y responsabilidades. Se eliminaron/integraron ventanas redundantes.
- **Diferenciadores:** Se actualizaron para reflejar las funcionalidades destacadas en los mockups (Gestor de Etiquetas con color/stats, Vistas Personalizadas con constructor visual).
- **Tecnologías:** Se añadieron Bootstrap y Font Awesome al stack de front-end.
- **Sección Nueva:** Se añadió esta sección "Cambios Realizados".

Conclusión

El Generador de Listas Etiquetadas responde a la necesidad de organizar información de manera flexible y filtrable, utilizando etiquetas y estados. **Los mockups detallados creados proporcionan una guía visual clara para la implementación.** Se espera que, mediante el MVP descrito y visualizado, el usuario final cuente con una interfaz sencilla e intuitiva para gestionar sus listas, **etiquetas y vistas**, y un back-end robusto con autenticación y validaciones que soporte estas funcionalidades. El proyecto está diseñado para permitir iteraciones futuras, añadiendo funcionalidades como compartir listas, notificaciones en tiempo real, etiquetas colaborativas, etc.

Con esto, se completan los requerimientos generales del proyecto, integrando la información recabada sobre front-end y back-end, **y alineándolos con los mockups visuales desarrollados.**