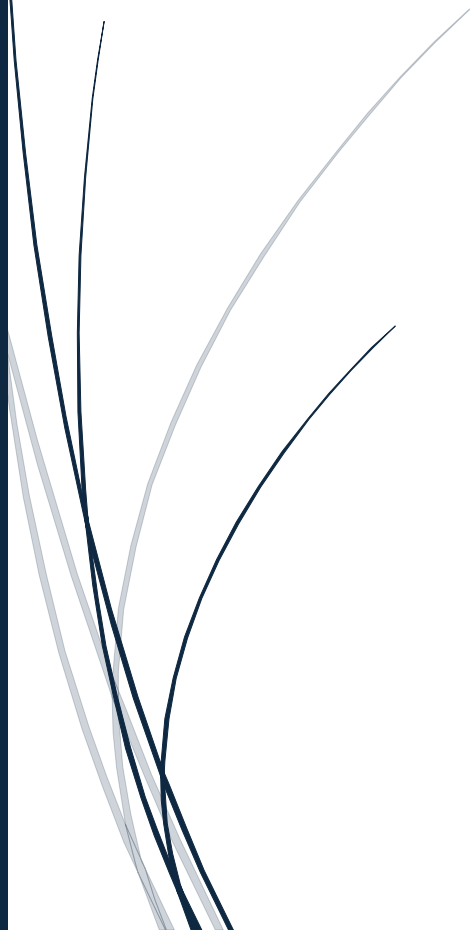




21-4-2025

Documentación prueba técnica Totalum



Rafael Pedrosa Castelo
TOTALUM

Contenido

Introducción 2

Estructura del proyecto 2

Introducción

Este proyecto es una prueba técnica desarrollada para Totalum. Consiste en una aplicación web construida con Angular que permite gestionar entidades genéricas a través de una interfaz reutilizable. La aplicación se conecta a una API REST para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre diferentes entidades.

Estructura del proyecto

prueba_tecnica_totalum/

```
├── src/
|   ├── app/
|   |   ├── components/
|   |   |   └── entity-table/
|   |   |       ├── entity-table.component.ts
|   |   |       ├── entity-table.component.html
|   |   |       └── entity-table.component.css
|   |   ├── services/
|   |   |   └── totalum-api.service.ts
|   |   ├── app.module.ts
|   |   └── app.component.ts
|   └── index.html
├── angular.json
├── package.json
└── README.md
```

Componentes Principales

EntityTableComponent

Componente reutilizable que permite mostrar y gestionar diferentes entidades en forma de tabla. Sus características principales incluyen:

- **Visualización de datos:** Muestra los datos en una tabla con paginación.
- **Búsqueda:** Permite filtrar los datos por un campo específico.
- **Edición en línea:** Posibilidad de editar los datos directamente en la tabla.
- **Modal para creación:** Formulario modal para añadir nuevas entidades.
- **Eliminación:** Permite eliminar entidades.

Inputs

- `tableName`: string - Nombre de la entidad a gestionar.
- `title`: string - Título que se muestra en la tabla.
- `items`: any[] - Lista de elementos a mostrar.
- `columns`: string[] - Columnas que se mostrarán en la tabla.
- `editable`: boolean - Indica si la tabla es editable.
- `searchField`: string - Campo por el cual se realizará la búsqueda.

Métodos

- `loadItems()` - Carga los elementos desde la API.
- `addItem(newItem: any)` - Añade un nuevo elemento.
- `updateItem(item: any)` - Actualiza un elemento existente.
- `deleteItem(id: string)` - Elimina un elemento.
- `startEdit(item: any)` - Inicia la edición de un elemento.
- `saveEdit()` - Guarda los cambios realizados en la edición.
- `cancelEdit()` - Cancela la edición.
- `filteredItems()` - Devuelve los elementos filtrados según la búsqueda.
- `totalPages` - Calcula el total de páginas para la paginación.
- `nextPage()` - Avanza a la siguiente página.
- `previousPage()` - Retrocede a la página anterior.

Base de datos

El servicio TotalumApiService actúa como puente entre el frontend Angular y la base de datos gestionada a través de la SDK de **Totalum API**.

Contiene todos los métodos de las CRUDS, la paginación y el buscador, que vienen dados en la misma documentación de Totalum