

# Prueba técnica

El reto consiste en crear una aplicación para gestionar listados de tareas de usuarios (TODOs).

Inicialmente, la aplicación debe ser accesible públicamente sin necesidad de autenticación.

## Modelado de entidades

Modelar estas dos entidades (User y TODO).

Un TODO tendrá los siguientes atributos:

- Id (autoincremental)
- Title (string)
- Completed (boolean)
- User (ManyToOne a usuario)

Un User tendrá los siguientes atributos:

- Id (autoincremental)
- Name (string)
- Username (string)
- Password (string)
- Address
  - Street (string)
  - City (string)
  - Zipcode (string)
  - Country (string)

Para modelar entidades usar annotations JPA (`javax.persistence.*` o `jakarta.persistence.*`), evitar usar directamente annotations de Hibernate siempre que sea posible (`org.hibernate.*`).

Para implementar las operaciones de BBDD usar Spring Data JPA, la BBDD puede ser la que quieras (MySQL, MariaDB, PostgreSQL, H2, ...).

## Pantalla de listado de TODOs

En esta pantalla habrá una tabla que listará los TODOs, con las siguientes columnas: título, username, país y si la tarea está completada o no. Se debe poder filtrar por título (que el título contenga el texto) y por username (que haga match exacto con el nombre de usuario).

El listado mostrará 10 registros y se debe poder paginar. ~~Implementar el~~ sistema de paginado [a implementar](#) es libre (infinite scroll, botones next/prev, botones con los números de página, ...).

[Las columnas deben ser ordenables.](#)

Con formato: Título 1, Izquierda

Con formato: Título 1, Izquierda

## Pantalla de creación de TODOs

En esta pantalla habrá un formulario con campos: `userId`, `title` y `completed`. Con las siguientes validaciones:

- `userId`: selector de usuarios (se muestra el username al usuario, se manda el `userId` al hacer submit)
- `title`: string no vacío y con una longitud < 200
- `Completed`: selector con 2 opciones ("Completed" y "Not completed")

## Pantalla de edición de TODOs

- En la tabla de la pantalla de listado de TODOs, crear una columna extra con un botón que permita editar el TODO. Este botón debe llevarte a la pantalla de creación de TODOs, donde el formulario vendrá informado con los datos de ese TODO y permitirá editarlo.

### Añadir paginación y ordenación

Otra cosa que no nos gusta de la API es que no permite paginar ni ordenar los TODOs. Añadir dos parámetros opcionales (`pageNumber`, `pageLength`) al endpoint `GET /todos` que permita paginar. Si no se mandan los parámetros, se devuelven 10 resultados de la primera página.

Añadir dos parámetros opcionales (`sortBy`, `sortDir`) al endpoint `GET /todos` que permita especificar el campo por el que se quiere ordenar y el sentido (`asc` o `desc`).

### Añadir filtros

Queremos añadir también la opción de poder filtrar tareas por título (que el título contenga el texto) y por username (que haga match exacto con el nombre de usuario).

## Bonus 1 – Añadir seguridad

Tampoco nos gusta que un usuario pueda editar los TODOs de otro usuario. Para evitarlo hay que proteger el endpoint `PUT /todos` de que se llama al editar los TODOs de modo que solo se pueda llamar si estás logado y controlar que el TODO que se está modificando sea el del usuario logado. Los pasos son:

- Crear un endpoint de login: Se le pasa un usuario y password, y devuelve un token JWT si el login es correcto.
- Proteger el endpoint `PUT /todos` de modo que solo pueda ser llamado por usuarios previamente logados. Para ello solo deberá aceptar llamadas que contengan el token JWT en el header `Authorization`.

## Bonus 2 – Añadir tests unitarios

Crear tests unitarios para validar que la aplicación funciona como se espera. Créalos a los niveles que quieras (controller, servicio, repositorio, ...).

Con formato: Título 1, Izquierda

Con formato: Español (España)

Con formato: Normal, Sin viñetas ni numeración

Con formato: Español (España)

Con formato: Título 1, Izquierda

Con formato: Normal, Sin viñetas ni numeración

Con formato: Título 1, Izquierda

# Frontend challenge

El objetivo es hacer una frontend para que los usuarios puedan gestionar tareas. El frontend debe llamar a la API que se haya programado. Los requisitos de la aplicación son los siguientes.

## Pantalla de listado de TODOs

En esta pantalla habrá una tabla que listará los TODOs, con las siguientes columnas: título, username y si la tarea está completada o no. Se debe poder filtrar por username (el filtro debe hacer una nueva llamada a la API, no filtrar en memoria).

## Pantalla de creación de TODOs

En esta pantalla habrá un formulario con campos: `userId`, `title` y `completed`. Con las siguientes validaciones:

- `userId`: campo numérico  $> 0$
- `title`: string no vacío y con una longitud  $< 200$
- `Completed`: un select con 2 opciones ("Completed" y "Not-completed").

El formulario debe ser reactivo (reactive form).

## Bonus 1 — Paginación y ordenación de TODOs

En la tabla de la pantalla de listado de TODOs, permitir paginar y ordenar. Implementar el sistema de paginado es libre (infinite scroll, botones next/prev, botones con los números de página, ...).

## Bonus 2 — Eliminación de TODOs

En la tabla de la pantalla de listado de TODOs, crear una columna extra con un botón que permita borrar el TODO. Este botón debe abrir un popup de confirmación con un mensaje "Are you sure? This operation cannot be undone". Si cancela, no pasa nada. Si confirma se borra. Solo se deben poder borrar tus propios TODOs.

## Bonus 2.3 — Añadir tests unitarios

Crear tests unitarios para validar que la aplicación funciona como se espera. Créalos a los niveles que quieras (controller, servicio, repositorio, ...).

## Bonus 3 — Pantalla de login + edición de TODOs

Hacer una pantalla de login con un formulario con 2 campos: `user`/`password`.

En la tabla de la pantalla de listado de TODOs, crear una columna extra con un botón que permita editar el TODO. Este botón debe llevarte a la pantalla de creación de TODOs, donde el formulario vendrá informado con los datos de ese TODO y permitirá editarlo. El botón y el formulario solo deben ser accesibles si el usuario está autenticado.

Una vez autenticado, la aplicación solo debería permitir editar al usuario sus propios TODOs.

Con formato: Título 1, Izquierda

Con formato: Título 1, Izquierda

## Reglas

1. Los bonus son opcionales, pero cuantos más bonus se hagan mejor.
2. En la entrega incluir un README con los pasos necesarios para configurar/ejecutar la aplicación.
3. Puedes hacernos llegar la entrega subiendo el código fuente a un repositorio GIT público (github, bitbucket, etc) y enviándonos el link.