Prueba técnica para Desarrollador Full Stack Junior

Instrucciones:

- Debes completar los ejercicios utilizando Angular y Spring Boot.
- El código se debe subir a un repositorio en GitLab o GitHub
- Mencionar consideraciones hechas por el postulante mediante el correo electrónico de entrega

Notas: Es importante mencionar que cualquier restricción no especificada en la prueba queda a consideración del desarrollador y debe ser abordada de manera adecuada para garantizar el correcto funcionamiento del sistema.

FRONTEND

Crea una aplicación Angular que contenga una vista de listado de usuarios. Los usuarios se deben obtener de una API REST que se proporciona a continuación:

- Método: GET
- Endpoint: /api/users
- Respuesta: un array de objetos JSON con la siguiente estructura:

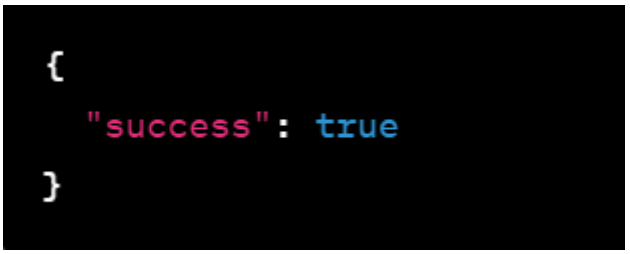
```
[
  {
    "id": 1,
    "name": "John Doe",
    "email": "johndoe@example.com"
  },
  {
    "id": 2,
    "name": "Jane Smith",
    "email": "janesmith@example.com"
  },
  {
    "id": 3,
    "name": "Bob Johnson",
    "email": "bobjohnson@example.com"
  }
]
```

La vista de listado de usuarios debe mostrar los siguientes datos para cada usuario:

- ID
- Nombre
- Email

Debes permitir la eliminación de un usuario desde la vista de listado. Cuando se elimine un usuario, se debe enviar una solicitud DELETE a la API REST con el siguiente endpoint:

- Método: DELETE
- Endpoint: /api/users/{id}
- Respuesta: un objeto JSON con el siguiente formato:



BACKEND
Crea una aplicación Spring Boot que proporcione una API REST para la gestión de usuarios. Debes utilizar una base de datos H2 para almacenar los datos de los usuarios.
La API REST debe proporcionar los siguientes endpoints:

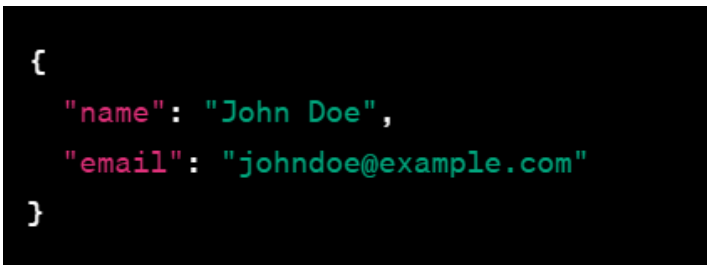
Endpoint 1.

- Método: GET
- Endpoint: /api/users
- Función: Obtener todos los usuarios de la base de datos
- Respuesta: un array de objetos JSON con la siguiente estructura:

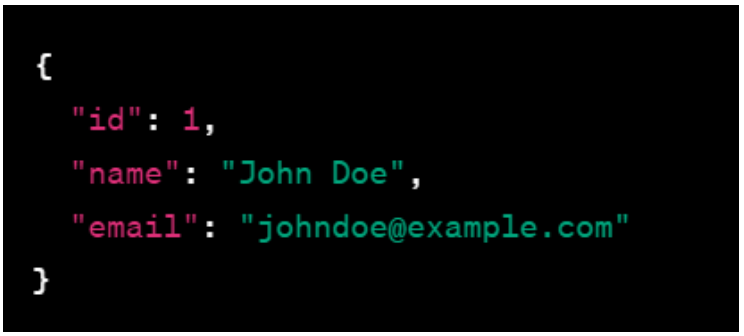
```
[
  {
    "id": 1,
    "name": "John Doe",
    "email": "johndoe@example.com"
  },
  {
    "id": 2,
    "name": "Jane Smith",
    "email": "janesmith@example.com"
  },
  {
    "id": 3,
    "name": "Bob Johnson",
    "email": "bobjohnson@example.com"
  }
]
```

Endpoint 2.

- Método: POST
- Endpoint: /api/users
- Función: Agregar un nuevo usuario a la base de datos
- Cuerpo de la solicitud: un objeto JSON con la siguiente estructura:

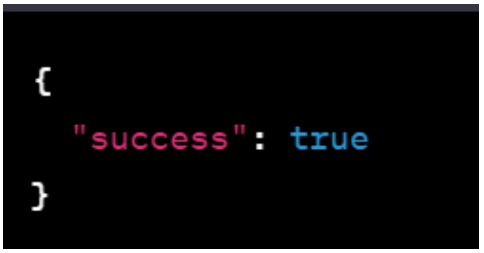


- Respuesta: un objeto JSON con el siguiente formato:



Endpoint 3.

- Método: DELETE
- Endpoint: /api/users/{id}
- Función; Borrar un usuario de la base de datos
- Respuesta: un objeto JSON con el siguiente formato:



BONUS

- Implementa una función de búsqueda en la vista de listado de usuarios. La búsqueda debe permitir buscar usuarios por nombre y email.
- No hay especificaciones para el diseño de las vistas pero se considerará como bonus una interfaz de usuario agradable.