



Challenge Técnico - Fullstack Developer

Empieza creando el repositorio, donde desarrollarás una aplicación de lista de tareas utilizando Angular con su respectivo backend. La aplicación debe permitir al usuario agregar, editar y eliminar tareas, así como marcarlas como completadas. Deberás utilizar las mejores prácticas de Angular y seguir los principios de arquitectura de componentes.

Requerimientos:

- La aplicación consta de 2 páginas, la primera la cual funcionará como un inicio de sesión y la segunda página que será la principal, en la cual se mostrarán todas las tareas pendientes del usuario ordenadas por fecha de creación.
- La página de inicio de sesión tiene un formulario donde solo se debe pedir el correo. Si el usuario existe, navega a la página principal, en caso contrario se deberá presentar un diálogo que confirme la creación del usuario. Si se crea el usuario debe navegar directamente a la página principal.
- Debe haber un formulario en la página principal que permite agregar nuevas tareas. Además de un botón que permite volver al inicio de sesión.
- Cada tarea debe mostrar su título, descripción, fecha de creación y estado de completado.
- El usuario debe poder marcar una tarea como completada o pendiente mediante una casilla de verificación.
- Debe existir una opción para editar y eliminar una tarea.
- La aplicación debe ser responsive y adaptarse a diferentes dispositivos.
- Desarrollar un API con lo siguiente:
 - Express y Typescript
 - Hosteado en cloudfunctions
 - Firebase Firestore para guardar los datos de las tareas
 - El API deberá tener los siguientes endpoints:



- GET /tasks: Obtener la lista de todas las tareas.
- POST /tasks: Agregar una nueva tarea.
- PUT /tasks/{taskId}: Actualizar los datos de una tarea existente.
- DELETE /tasks/{taskId}: Eliminar una tarea existente.
- GET /users/{email}: Busca el usuario si ha sido creado
- POST /users : Agrega un nuevo usuario

Aspectos a evaluar:

- Estructura y organización del proyecto: Evalúa cómo el desarrollador organiza y estructura los archivos y carpetas en la aplicación Angular, siguiendo las convenciones y mejores prácticas.
- Uso de componentes: Evalúa si el desarrollador utiliza componentes de manera adecuada, dividiendo la lógica y la interfaz de usuario en componentes reutilizables.
- Manipulación de datos: Evalúa cómo el desarrollador maneja los datos del usuario y de las tareas conectándose al API desarrollado.
- Enlace de datos y directivas: Evalúa si el desarrollador utiliza correctamente las directivas de Angular, como ngFor, ngIf y ngModel, para enlazar datos y controlar el flujo de la aplicación.
- Buenas prácticas de codificación y estilo: Evalúa el uso adecuado de Typescript, que el código sea intuitivo, evita errores de sintaxis y que respeta las convenciones del lenguaje y del proyecto.
- Routing: Evalúa si el desarrollador utiliza el enrutamiento de Angular para navegar entre las diferentes vistas de la aplicación.
- Estilo y diseño: Evalúa la calidad del diseño y el estilo de la interfaz de usuario, así como el uso de SCSS y/o librerías de estilos o componentes (Angular Material).

**Puntos iniciales:**

Para este reto, te proporcionamos plantillas con algunas dependencias ya configuradas para no tener que arrancar el proyecto desde 0, sin embargo, puedes hacerlo como más te sientas cómodo.

- FrontEnd
 - [Angular 15](#)
 - [Angular 17](#)
- BackEnd
 - Implementar según tus necesidades.

Entrega de la prueba:

- Carga el código fuente de la aplicación en un repositorio git con acceso público.
- Publica la aplicación en firebase hosting de preferencia. O en su defecto cualquier otro hosting que permita probar online (ej: <https://stackblitz.com/>).
- En el repositorio puedes incluir una breve documentación explicando las decisiones de diseño, las tecnologías utilizadas y/o algún comentario que consideres relevante (README - Comentarios de desarrollo).