

# Challenge Técnico

## FullStack Developer



Bienvenido a nuestro challenge técnico, te invitamos a leer cada uno de los puntos que mencionamos a continuación para que desarrolles el challenge con éxito.

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

1. 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.
2. 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.
3. Debe haber un formulario en la página principal que permite agregar nuevas tareas. Además de un botón que permite volver al inicio de sesión.
4. Cada tarea debe mostrar su título, descripción, fecha de creación y estado de completado.

5. El usuario debe poder marcar una tarea como completada o pendiente mediante una casilla de verificación.
6. Debe existir una opción para editar y eliminar una tarea.
7. La aplicación debe ser responsive y adaptarse a diferentes dispositivos.
8. Desarrollar un API con lo siguiente:

Unset

Express y Typescript

Hosteado en cloudfunctions

Firebase Firestore para guardar los datos de las tareas

El API deberá tener los siguientes endpoints:

Unset

Obtener la lista de todas las tareas.

Agregar una nueva tarea.

Actualizar los datos de una tarea existente.

Eliminar una tarea existente.

Busca el usuario si ha sido creado

Agrega un nuevo usuario

## Arquitectura y Organización

- Organización del proyecto (carpetas, modularidad, separación de capas).
- Uso de arquitectura limpia o hexagonal.

## Patrones de Diseño

- Aplicación de SOLID y principios de diseño.
- En frontend: observables, servicios, componentes bien estructurados.
- En backend: DDD, repositorios, factories, singletons.

## Manejo de Datos

- Uso eficiente de servicios HTTP, validaciones y transformación de datos.
- Seguridad en comunicación con el API (tokens, autenticación).

## Binding y Directivas

- Uso correcto de directivas estructurales y personalizadas.
- Enlace de datos optimizado (one-way, two-way binding, async pipes).
- Manejo eficiente del DOM (trackBy en [ngFor](#)).

## Buenas Prácticas de Código

- Aplicación de DRY, KISS y YAGNI.
- Uso correcto de TypeScript (tipado, interfaces, generics).
- Pruebas unitarias e integración en frontend y backend.
- Código documentado y README útil.

## Seguridad

- Configuración de CORS y validaciones en backend.
- Gestión segura de secretos y tokens.

## Enrutamiento

- Organización del [RouterModule](#), uso de guards y lazy loading.

## Estilo y Diseño

- Implementación de diseño responsivo con Angular Material, Bootstrap, etc.
- Consistencia en estilos (SCSS, variables globales).

- Accesibilidad (etiquetas ARIA, navegación por teclado).

### **Entrega y Despliegue**

- Scripts de build optimizados (tree shaking, minificación).
- Uso de CI/CD para integración y despliegue.
- Documentación clara del proceso de configuración.

### **NOTA**

Se evaluará únicamente lo entregado. Si el participante no está familiarizado con algún aspecto, es preferible **no implementarlo** en lugar de hacerlo de manera incorrecta. La calidad del código y el cumplimiento de buenas prácticas serán priorizados sobre la cantidad de funcionalidades implementadas.

No es necesario implementar todo para demostrar tus habilidades. Nos interesa ver cómo abor das el problema, la calidad del código y tus decisiones técnicas.

Si lo deseas, puedes incluir funcionalidades extra que consideres valiosas, como filtros de búsqueda, categorías de tareas o una experiencia de usuario mejorada

## 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 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).