

## **TEST TÉCNICO**

### **ANGULAR - NODE - MongoDB**

#### **Objetivo:**

Evaluar las habilidades del candidato en el desarrollo de aplicaciones web completas utilizando las tecnologías Angular, Node.js y MongoDB. La prueba se centrará en la creación de una aplicación funcional que gestione una lista de empleados, demostrando las siguientes capacidades del candidato:

#### **Habilidades Técnicas:**

- Implementación de funcionalidades CRUD (Crear, Leer, Actualizar y Eliminar) para la gestión de empleados.
- Consumo de datos de una API externa utilizando Node.js.
- Conexión de la aplicación Angular con una base de datos MongoDB.
- Diseño de una interfaz de usuario intuitiva y amigable.
- Escritura de código limpio, organizado, reutilizable y legible.
- Implementación de mecanismos de autenticación y autorización con tokens JWT.
- Seguimiento de las mejores prácticas de desarrollo de Angular, Node.js y MongoDB.

#### **Descripción de la Prueba:**

El candidato deberá desarrollar una aplicación web completa con Angular que permita gestionar una lista de empleados. La aplicación debe cumplir con los siguientes requisitos:

#### **Funcionalidades:**

- **Listado de empleados:**
  - Visualizar todos los empleados almacenados en la base de datos.
  - Buscar empleados por nombre.
- **Contratación de empleados:**
  - Registrar nuevos empleados con nombre, apellidos, puesto de trabajo y fecha de nacimiento.
  - Permitir la selección del puesto de trabajo desde un listado predefinido obtenido de la API: <https://ibillboard.com/api/positions>.
- **Modificación de empleados:**
  - Permitir al usuario editar el nombre de un empleado existente.
  - Permitir al empleado modificar su propio puesto de trabajo.
- **Baja de empleados:**
  - Permitir al empleado eliminar su registro de la aplicación.

#### **Aspectos Técnicos:**

- **Interfaz de usuario:**
  - Diseño amigable y fácil de usar.
  - Experiencia de usuario intuitiva.
  - Implementación de componentes reutilizables y modulares.

- **Código:**
  - Código limpio, organizado, reutilizable y legible.
  - Seguimiento de las mejores prácticas de desarrollo de Angular, Node.js y MongoDB.
  - Documentación del código.
- **Integración:**
  - Consumir datos de la API de puestos de trabajo: <https://ibillboard.com/api/positions>, utilizando Node.js.
  - Conectar la aplicación Angular con una base de datos MongoDB para almacenar y recuperar datos de empleados.
- **Autenticación y Autorización:**
  - Implementar un sistema de autenticación y autorización con tokens JWT para controlar el acceso a las funcionalidades de la aplicación.

### Implementación del Token JWT:

El candidato debe implementar un sistema de autenticación y autorización con tokens JWT para controlar el acceso a las funcionalidades de la aplicación. Esto implica lo siguiente:

- **Registro de usuarios:**
  - Crear un endpoint en el servidor Node.js para recibir los datos de registro del usuario (correo electrónico y contraseña).
  - Validar los datos de registro del usuario.
  - Hashear la contraseña del usuario.
  - Crear un nuevo usuario en la base de datos MongoDB con la información proporcionada.
  - Generar un token JWT con el ID del usuario y otra información relevante.
  - Enviar el token JWT al cliente Angular.
- **Inicio de sesión:**
  - Crear un endpoint en el servidor Node.js para recibir las credenciales del usuario (correo electrónico y contraseña).
  - Validar las credenciales del usuario contra la base de datos MongoDB.
  - Si las credenciales son válidas, generar un token JWT con el ID del usuario y otra información relevante.
  - Enviar el token JWT al cliente Angular.
- **Protección de rutas en Angular:**
  - Utilizar el guard de autenticación de Angular para proteger las rutas que requieren que el usuario esté autenticado.
  - Verificar el token JWT en el guard de autenticación para determinar si el usuario está autenticado.
  - Redirigir al usuario a la página de inicio de sesión si no está autenticado.
- **Almacenamiento y uso del token JWT:**
  - En el cliente Angular, almacenar el token JWT en el almacenamiento local o en una cookie segura.
  - Incluir el token JWT en las cabeceras de las solicitudes HTTP a las API protegidas.
  - El servidor Node.js debe verificar el token JWT

- **Servidor Node.js:**
  - Verificar el token JWT en cada solicitud para validar la autenticación del usuario.
  - Extraer la información del usuario del token JWT.
  - Utilizar la información del usuario para controlar el acceso a los recursos y funcionalidades de la aplicación.
- **Cierre de sesión:**
  - Crear un endpoint en el servidor Node.js para cerrar la sesión del usuario.
  - Invalidar el token JWT del usuario en el servidor.
  - Eliminar el token JWT del almacenamiento local o de la cookie en el cliente Angular.
  - Redirigir al usuario a la página de inicio de sesión.

#### **Consideraciones:**

- Implementar un mecanismo para la recuperación de contraseña en caso de que el usuario la olvide.
- Establecer una política de expiración para los tokens JWT para mejorar la seguridad.
- Implementar estrategias de seguridad adicionales, como el uso de HTTPS y la protección contra CSRF.

#### **Integración con la Prueba Técnica Existente:**

- Los endpoints de autenticación (registro, inicio de sesión y cierre de sesión) deben integrarse con el resto de la aplicación.
- El token JWT debe usarse para identificar al usuario en las solicitudes a las API que requieren autenticación.
- Las funcionalidades de la aplicación deben estar disponibles sólo para usuarios autenticados.

#### **Entrega:**

El candidato debe entregar la aplicación web completa en un repositorio de GitHub público. El repositorio debe incluir:

- Código fuente de la aplicación Angular.
- Código del servidor Node.js para la integración con la API, la base de datos y la autenticación.
- Script para la ejecución de la aplicación.
- README con instrucciones de instalación y ejecución.
- Colección de endpoints de la API en Postman o similar.

#### **Presentación:**

El candidato debe presentar el enlace al repositorio de GitHub durante la entrevista o en el plazo establecido.

¡Esperamos una excelente presentación de tu trabajo!