

<septiembre 2024>

## Prueba Técnica. Net / Angular

Estimado(a) candidato(a).

Nos enorgullece saber que Double V Partners tiene el potencial de ser parte de tu proyecto de vida. Sabemos que las pruebas técnicas son desgastantes y algo pesadas, pero creemos que un buen proceso de selección es el que nos permite ver el potencial de una persona y también le permite al candidato(a) saber si este va a ser un trabajo retador, divertido, apasionante y acorde a tus expectativas. Date la oportunidad de hacer esto a conciencia no para pasar una prueba, sino para entender si este es el lugar adecuado para tu crecimiento profesional.

### Instrucciones

- ❖ La siguiente es una prueba técnica donde tendrás completa libertad para proponer, idear, planear y plasmar tu filosofía de trabajo.
- ❖ La duración estimada para completar la prueba es de 2 días para terminarla a partir de la hora en que la recibas.
- ❖ Entregamos en un repositorio Git Hub.

A continuación te envío la prueba técnica y la prueba de conceptos para que la puedas realizar. El tiempo de entrega es máximo 2 días.

### Adjunto los ejercicios de la prueba Técnica:

**Objetivo:** Crear un sistema de asignación de tareas con autenticación de usuarios y sistema de roles **utilizando .NET para el back-end y Angular para el front-end.**

**Descripción del Proyecto:** La empresa DVP necesita un sistema para gestionar las tareas de sus colaboradores. El sistema debe permitir la autenticación de usuarios y tener un sistema de roles que determine las funcionalidades accesibles para cada tipo de usuario (Administrador, Supervisor y Empleado).

## **Requerimientos Funcionales:**

### **1. Autenticación de Usuarios:**

- Implementar el registro y el inicio de sesión de usuarios.
- Utilizar JWT (JSON Web Token) para la autenticación.

### **2. Sistema de Roles:**

- Administrador: Puede crear, editar, eliminar usuarios y tareas.
- Supervisor: Puede asignar tareas a los empleados y cambiar el estado de las tareas.
- Empleado: Puede ver sus tareas asignadas y actualizar el estado de las mismas.

### **3. Gestión de Tareas:**

- CRUD (Crear, Leer, Actualizar, Eliminar) de tareas.
- Asignación de tareas a usuarios.
- Actualización del estado de las tareas (Pendiente, En Proceso, Completada).

### **4. Gestión de Usuarios:**

- CRUD de usuarios.
- Asignación de roles a los usuarios.

## **Requerimientos Técnicos:**

### **1. Back-end (.NET):**

- Crear una REST API utilizando ASP.NET Core.
- Utilizar Entity Framework Core para la interacción con la base de datos.
- Implementar la autenticación con JWT.
- Crear las siguientes entidades: Usuario, Rol, Tarea.
- Implementar las validaciones necesarias.

### **2. Front-end (Angular):**

- Crear una aplicación en Angular
- Implementar el inicio de sesión y registro de usuarios.
- Crear vistas para la gestión de usuarios y tareas según el rol del usuario.
- Utilizar Angular Material para la interfaz de usuario(Opcional).
- Consumir la API creada en el back-end para las operaciones CRUD.

## **Entrega:**

- Un repositorio en GitHub con el código del proyecto.
- Instrucciones claras sobre cómo configurar y ejecutar el proyecto

localmente.

- Un archivo README detallando la arquitectura del sistema, decisiones técnicas y cualquier consideración especial.

#### **Criterios de Evaluación:**

1. **Funcionalidad:** El código debe cumplir con los requisitos especificados.
2. **Calidad del Código:** Uso correcto de la sintaxis, nombres de variables claros, y código bien comentado.
3. **Eficiencia:** El código debe ser eficiente y seguir buenas prácticas de programación.
4. **Diseño de la Interfaz:** La aplicación debe ser intuitiva y fácil de usar.
5. **Documentación:** La documentación debe ser clara y completa. **Puntos Extra:**

- Implementación de pruebas unitarias para el back-end y/o front-end.
- Implementación de paginación y búsqueda en la gestión de tareas.
- Uso de Docker para la configuración del entorno de desarrollo.

¡¡¡Mucha suerte y gracias por el tiempo que nos dedicas :) !!!