

Prueba Técnica: Sistema de Gestión de Citas (Full Stack .NET + Angular)

Duración estimada: 3 horas.

Objetivo: Desarrollar una solución Full Stack para la gestión de citas de mantenimiento vehicular. Buscamos evaluar tu capacidad para construir una solución escalable, tu limpieza de código y el dominio de las tecnologías solicitadas.

1. Requerimientos de Negocio (El Problema)

Debes construir una aplicación que conste de dos secciones principales:

A. Consulta de Historial

- El usuario ingresa una placa (Ej: PDF-1234).
- El sistema muestra el listado de citas previas y futuras asociadas a esa placa.
- Si no hay citas, debe mostrar un mensaje amigable indicándolo.

B. Agendamiento de Nuevas Citas

- El usuario puede crear una nueva cita seleccionando **Fecha y Hora**.
- **Regla de Negocio 1:** Las citas son estrictamente de **Lunes a Viernes**, entre las **08:00 AM y las 14:00 PM**.
- **Regla de Negocio 2:** Los intervalos de atención son de 30 minutos (ej: 8:00, 8:30, 9:00...).
- **Validación Crítica:** El sistema **NO** debe permitir agendar una cita si ese horario ya está ocupado por otro vehículo. **PUNTO EXTRA**

2. Requerimientos Técnicos (El Stack)

Backend:

- API RESTful escrita en C#.
- Base de datos: La que tengas disponible o In-Memory Database (para facilitar la prueba).
- Se valorará el uso de DTOs y separación de capas (Controller vs Service/Repository).

Frontend:

- Uso de **Angular Material** o Bootstrap para la UI (no pierdas tiempo en CSS personalizado, buscamos funcionalidad y orden).
- Manejo de Validaciones en el cliente (ej: campo placa obligatorio, formato correcto).

3. Criterios de Evaluación

No solo evaluamos que "funcione", evaluamos cómo está construido:

1. **Arquitectura:** ¿Tu código está desacoplado? ¿La lógica de negocio está en el lugar correcto?
2. **Clean Code:** Nombres de variables descriptivos, métodos cortos, código legible.
3. **Manejo de Errores:** ¿Qué pasa si la API falla? ¿Qué pasa si la base de datos está caída?

4. Entregables

1. Sube tu código a un repositorio público en **GitHub**.
2. Incluye un archivo README con:
 - Instrucciones para levantar el proyecto (Backend y Frontend).

- Si usaste base de datos, incluye el script de base de datos o instrucciones de migración.
- 3. Video explicativo de la navegación en la aplicación ó documento explicativo con las capturas de pantalla del funcionamiento.

Recomendaciones Importantes

1. **Prioriza la Funcionalidad:** No buscamos un diseño gráfico perfecto, sino una aplicación que funcione correctamente y valide las reglas de negocio (horarios y disponibilidad).
2. **Calidad de Código:** Evaluaremos el orden, el nombrado de variables y la separación de responsabilidades (Backend vs Frontend).
3. **Tiempo:** Este reto está diseñado para resolverse en **máximo 3 horas**. Si no terminas alguna funcionalidad, indícalo en el README; valoramos más el código limpio incompleto que el código "spaghetti" completo.