



## Challenge Técnico: Full-Stack DEV (Angular + NET)

Te invitamos a participar en nuestro desafío técnico para el puesto de Desarrollador Full-Stack (Angular + .NET).

Este ejercicio busca evaluar tus habilidades técnicas, capacidad de resolución de problemas y buenas prácticas de desarrollo en un entorno práctico similar a los proyectos reales que desarrollamos en Tecnom.

### Sobre Tecnom y nuestro producto Boxes:

En **Tecnom** desarrollamos soluciones tecnológicas que optimizan los procesos de negocio de nuestros clientes.

Uno de nuestros principales productos es [Boxes](#), una plataforma enfocada en mejorar la postventa mediante la automatización y gestión eficiente de servicios, ofreciendo así una experiencia superior al cliente final.

El desafío que te proponemos está relacionado con la integración de un caso de uso específico dentro del área de postventa: la reserva de turnos.

A través de este ejercicio, podrás sumergirte en el tipo de proyectos que desarrollamos y demostrar cómo tu talento puede contribuir a mejorar nuestros productos.

---

### Descripción del Desafío:

El desafío consiste en desarrollar una aplicación Full-Stack que permita la reserva y visualización de turnos en talleres, replicando un caso de uso real de Boxes.

## Frontend


El objetivo del frontend es implementar una pequeña aplicación web que permita **crear y visualizar turnos**.

Debe contemplar los siguientes grupos de datos:

- Turno: información del servicio (fecha, hora, taller, tipo de servicio).
- Persona: datos de contacto (nombre, email, teléfono).
- Vehículo: información del vehículo (marca, modelo, año, patente — opcional).

### **Requerimientos funcionales mínimos:**

- Mostrar un **listado de turnos creados**, incluyendo la información principal (servicio, fecha y hora).
- Permitir la **creación de nuevos turnos**, validando los campos obligatorios.
- Obtener el **listado de talleres** disponibles desde la API indicada.
- Los datos del vehículo deben ser opcionales.

 *El diseño, la estructura de pantallas y el flujo de creación quedan a tu criterio. Lo que nos interesa evaluar es tu criterio para construir una **interfaz clara, usable y mantenible**.*

## Backend

Basado en la sección de frontend, implementar una API RESTful en .NET que exponga los siguientes endpoints para la gestión de turnos y consulta de talleres:

### 1. Endpoint POST para la Creación de Turnos (Reserva de Servicio)

- Método: POST
- Ruta: /api/appointments (o /api/leads)
- Objetivo: Recibir y simular el guardado de una reserva de turno.
- Validaciones: Validaciones básicas (campos requeridos: name, email, appointment\_at, service\_type, place\_id).
- Validación externa: validar que el place\_id sea un taller activo.

Estructura:

```
{  
  
  "place_id": 2222,  
  
  "appointment_at": "2023-10-01 T10:00:00Z",
```

```
"service_type": "string",

"contact": {

    "name": "string",

    "email": "string",

    "phone": "string"

},

"vehicle": { // Opcional

    "make": "string",

    "model": "string",

    "year": "number",

    "license_plate": "patente"

}

}
```

## 2. Endpoint GET para el Listado de Turnos

- Método: GET
- Ruta: /api/appointments (o /api/leads)
- Objetivo: Devolver los turnos guardados previamente (simulados en memoria) por el endpoint POST.
- Respuesta: Colección de objetos turno.

## 3. Endpoint GET para el listado de talleres (consumo api externa) (se utilizara en el paso 3 del frontend)

- Método: GET
- Ruta: /api/workshops (o /api/places/workshops)
- Objetivo: Consumir la API de Talleres externa
- Respuesta: Colección de talleres activos.
- URL: <https://dev.tecnomcrm.com/api/v1/places/workshops>
- Autenticación: Basic Authentication (User: max@tecnom.com.ar, Pass: b0x3sApp).
- Opcional: Implementar caching para evitar consultas repetitivas a la API externa.

---

### Requisitos Técnicos Frontend:

- **Framework:** Utilizar Angular en su versión más reciente. Se puede utilizar cualquier biblioteca de componentes y estilos (Angular Material, Bootstrap, Tailwind, etc.).
- **Lenguaje:** Desarrollar la aplicación usando TypeScript.
- **Buenas Prácticas:** Seguir estándares de código y buenas prácticas en HTML y CSS para asegurar la mantenibilidad y escalabilidad del proyecto.

### Requisitos Técnicos Backend:

- **Framework:** Utilizar NET en sus versiones más recientes (8, 9 o la preview 10).
- **Lenguaje:** Desarrollar la aplicación usando c#.
- **Buenas Prácticas:** Seguir estándares de código y buenas prácticas en ASP.NET y desarrollo web en general para asegurar la mantenibilidad y escalabilidad del proyecto.

### Requisitos Generales:

- **Repositorio:** Aloja tu código en un repositorio público o privado (Idealmente GitHub). Si es privado, asegúrate de compartir los permisos necesarios con nosotros a los siguiente Usuarios de github:
    - alainico1
    - matiasguazzaroni
    - mngobbi
  - **Consumo de API:**
    - La aplicación debe leer la configuración del taller (nombre, dirección, email, whatsapp) desde la siguiente API:
      - <https://dev.tecnomcrm.com/api/v1/places/workshops>
    - Para autenticarse usar Basic Authentication con las siguientes credenciales:
      - User [max@tecnom.com.ar](mailto:max@tecnom.com.ar)
      - Pass [b0x3sApp](#)
  - **Mock de Escritura:**
    - No hace falta implementar la funcionalidad de guardar los leads. Se puede simular esto guardando temporalmente en memoria (algo muy simple como variables del service, o algo más robusto como memory provider de EF) en cada ejecución de la app.
-

## Objetivos del Desafío:

- **Implementación de la API:** Que permita crear leads válidos (taller y campos OK), buscando que los errores sean claros cuando el cliente no cumple con lo requerido.
  - **Consumo de Datos Dinámicos:** Demostrar capacidad para consumir datos desde una API y manejar estados dentro de la aplicación.
  - **Calidad de Código:** Escribir código limpio y bien estructurado, facilitando su lectura y mantenimiento.
  - **Documentación:** Incluir comentarios o documentación que expliquen decisiones clave o aspectos destacables de tu implementación.
- 

## Después de la Entrega:

Coordinaremos una reunión con nosotros en la que participarán desarrolladores de nuestro equipo. Durante esta sesión, discutiremos:

- Tu enfoque para resolver el desafío.
- Las decisiones técnicas y de diseño que tomaste.
- Las dificultades encontradas y cómo las superaste.
- Oportunidades de mejora y tu perspectiva sobre el proyecto.

A esa reunión también trae tus dudas/inquietudes (técnicas y/o sobre modos de trabajo) ya que la idea es hacer una descontracturada donde emulamos un espacio de revisión de solución de manera colaborativa.

---

## Contacto:

Estamos a disposición para cualquier duda o consulta que tengas. Podés contactarnos por email. Recordá incluir a todos los contactos que detallamos a continuación; de esta forma, el primero que esté disponible podrá responder tus inquietudes.

- Alain Nicoletta Software Engineer - Monaco Product Team  
(alain.nicoletta@tecnom.com.ar)
  - Matias Guazzaroni Software Engineer - Monaco Product Team  
(mfguazzaroni@tecnom.com.ar)
  - Marco Gobbi CTO (mgobbi@tecnom.com.ar)
- 

¡Éxitos y seguimos en contacto!