

## Prueba técnica Gestor SR – Desarrollador .NET

### Descripción detallada de la solución:

La solución propuesta consiste en desarrollar una aplicación móvil y una página web para gestionar las rutas de transporte de los empleados de la compañía, cumpliendo con los requisitos mencionados. La aplicación se autenticará a través del directorio activo de la compañía para garantizar que solo los empleados tengan acceso.

### Diagrama de la arquitectura:

La arquitectura estará basada en la nube de Azure, utilizando servicios gestionados para garantizar escalabilidad, disponibilidad y seguridad.

**Frontend:** Tanto la aplicación móvil como la página web se desarrollarán utilizando un marco de trabajo de desarrollo multiplataforma como React Native o MAUI para la versión móvil y ASP.NET (con Razor Pages o MVC), React, Angular o Vue para la web, garantizando una interfaz de usuario consistente y basada en el manual de marca de la compañía.

Aunque mencioné varias opciones en el párrafo anterior recomiendo tener muy en cuenta:

ASP.NET Y MAUI (Por hacer parte del ecosistema .NET para la parte del Frontend).

Angular (Por su robustez, aunque haga parte del ecosistema .NET)

**Backend:** Se utilizará Azure Functions para implementar microservicios sin servidor que manejen las diversas funcionalidades del sistema, como el registro de rutas, cancelación, visualización de rutas y suscripción. Estos servicios estarán escritos en Node.js o C# según las preferencias y habilidades del equipo.

**Base de Datos:** Se utilizará Azure SQL Database para almacenar la información de las rutas y los usuarios. El modelo entidad-relación incluirá tablas para usuarios, rutas, suscripciones y notificaciones.

**Modelo Entidad Relación:** Se usará el siguiente modelo entidad relación.

Tabla	Atributos	Restricciones
Usuario	UserID (PK), Nombre, CorreoElectronico, Celular	UserID como clave primaria
Ruta	RutaID (PK), Fecha, TipoVehiculo, HoraSalida, CuposDisponibles, Origen, Destino, Estado (activo o cancelado), UserID (FK)	RutaID como clave primaria, UserID como clave foránea referenciando a Usuario

SuscripcionRuta	SuscripcionID (PK), RutaID (FK), UserID (FK)	SuscripcionID como clave primaria, RutaID como clave foránea referenciando a Ruta, UserID como clave foránea referenciando a Usuario
-----------------	---	--

**Autenticación y Autorización:** Azure Active Directory (AAD) se utilizará para autenticar y autorizar a los usuarios. Se implementará OAuth 2.0 para garantizar un flujo seguro de autenticación.

**Notificaciones:** Azure Notification Hubs se utilizará para enviar notificaciones push a los usuarios móviles y correos electrónicos para las notificaciones por correo.

**Almacenamiento:** Azure Blob Storage se usará para almacenar imágenes y otros archivos relacionados con las rutas.

#### **Metodología de desarrollo:**

Se utilizará un enfoque ágil, manejando Scrum como marco de trabajo. Los sprints se planificarán para iteraciones cortas y entregas incrementales, lo cual nos permite adaptarnos fácilmente a los cambios y mejorar continuamente a lo largo del tiempo.

#### **Buenas prácticas metodológicas:**

**Continuous Integration/Continuous Deployment (CI/CD):** Se implementará para garantizar la entrega continua y la detección temprana de problemas.

**Pruebas automatizadas:** Su uso sería para garantizar la calidad del software.

#### **Infraestructura y plataformas necesarias:**

Para simplificar la gestión y optimizar el despliegue, se propone un único entorno en Azure para desarrollo, pruebas y producción. Utilizando Azure DevOps o herramientas equivalentes, este entorno unificado empleará servicios administrados y contenedores para facilitar el desarrollo, las pruebas y la implementación directa en producción. El enfoque CI/CD permitirá una entrega continua, validando la aplicación en un entorno realista antes de su despliegue final. Se

recomienda ejecución de pruebas exhaustivas en el entorno de desarrollo y pruebas, junto con medidas de respaldo para la base de datos de producción.

**Posibles riesgos y mitigación:**

**La siguiente tabla muestra los posibles riesgos y la estrategia de mitigación propuesta.**

Riesgo	Mitigación
Problemas de integración debido a la complejidad de la arquitectura.	Mitigación: Realizar pruebas de integración continuas y utilizar entornos de desarrollo y prueba adecuados.
Cambios en los requisitos durante el desarrollo.	Mantener una comunicación constante con los stakeholders y adoptar un enfoque ágil para adaptarse a cambios.
Escalabilidad insuficiente para 500 usuarios diarios.	Utilizar servicios escalables de Azure y realizar pruebas de carga para identificar cuellos de botella.

**Otros elementos de prevención:**

Se recomienda:

Implementar un sistema de registro y monitoreo de errores utilizando Azure Application Insights.

Establecer políticas de seguridad y cumplimiento para proteger la información confidencial de los empleados.