

Prueba técnica Gestor SR – Desarrollador .NET

PARTE 1

Como parte de los proyectos de sostenibilidad de la compañía, se está buscando una solución que permita que los empleados publiquen sus rutas de transporte e indiquen los cupos que tienen disponibles en su vehículo, la hora de salida, el lugar de origen y el destino y que permita a los demás compañeros inscribirse en alguna de las rutas.

La aplicación debe tener las siguientes funcionalidades:

- **Autenticar a los empleados de la compañía**
 - La aplicación sólo puede ser accedida por los usuarios de la compañía.
 - Para facilitar la gestión es necesario que la aplicación use el directorio activo para autenticación y autorización.
- **Registrar mi ruta**
 - Como empleado necesito publicar la ruta que voy a realizar en una fecha particular, para que mis compañeros puedan suscribirse a ella.
 - Para el registro de la ruta se debe poder ingresar fecha, el tipo de vehículo, hora de salida, cupos disponibles, lugar de origen y lugar de destino.
- **Cancelar mi ruta**

Como empleado quiero poder cancelar una de las rutas que he publicado para que no esté disponible para suscripción. o Si la ruta es cancelada, se debe notificar a los usuarios que se habían suscrito.
- **Ver personas suscritas a mi ruta**
 - Como dueño de la ruta quiero poder ver los datos compañeros que se han suscrito a mi ruta para poder coordinar con ellos los detalles de la recogida.
 - De cada suscriptor a la ruta se debe ver nombre, correo electrónico, celular. Estos datos deben consultarse en el directorio activo.
- **Ver rutas disponibles**
 - Como empleado quiero poder ver el listado de rutas que fueron compartidas para la fecha actual para saber a cuál de ellas me puedo suscribir.
 - La lista debe permitir ver los datos generales de la ruta: hora de salida, cupos disponibles, lugar de origen y lugar de destino.
 - La lista me debe permitir ver las rutas para el día actual en orden ascendente de hora de salida.
 - Únicamente se podrán ver las rutas que tengan cupos disponibles y que sean en horas futuras.
- **Suscribirse a una ruta**

Como empleado quiero poder suscribirme a una ruta disponible para asegurar mi cupo. o Se debe notificar al usuario que publicó la ruta vía correo electrónico y a su celular si una persona se inscribe a su ruta.

- **Aspectos para tener en cuenta:**

- Se espera que la solución sea una aplicación móvil y una página web.
- La interfaz de usuario debe estar basada en el manual de marca de la compañía.
- El número aproximado de usuarios esperados es 500 diarios.

Como experto técnico usted debe plantear la solución para la necesidad del negocio y entregar:

- **Descripción detalladamente la solución planteada y sus componentes**

- Diagrama de arquitectura de la solución (Basada en la nube de Azure).
- Modelo entidad relación.
- Tecnologías, lenguajes de programación y frameworks a utilizar y justificación de la selección de cada uno.
- Metodología de desarrollo.
- Descripción de las buenas prácticas metodológicas que pueden agilizar el proceso de desarrollo, para la entrega de software de calidad en los diferentes ambientes.
- Infraestructura y plataformas necesaria para soportar el desarrollo.
- Posibles riesgos que pueden materializarse en la ejecución del proyecto y cómo mitigarlos.
- Otros elementos que considere necesarios para este proyecto.

PARTE 2

Un sistema de información cuenta con tres agentes (A, B y C) cada agente cumple con dos funcionalidades:

Funcionalidad 1: Obtener media

- Definir la función getMedia(Lista de números reales) → valor de retorno: número real
- **Agente A:** Obtener la media aritmética o promedio
- **Agente B:** Obtener media armónica

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

- **Agente C:** Obtener mediana

Si la cantidad de datos es impar, la mediana es el valor que queda en la mitad al ordenar los datos de menor a mayor.

Si la cantidad de datos es par, la mediana es el promedio de los dos valores que quedan al centro al ordenar los datos de menor a mayor.

Funcionalidad 2: Escalera

- Definir función `getStaircase(número entero)` → valor de retorno: cadena de texto
- **Agente A:**
La base y altura son ambas iguales a **n**. Se dibuja usando el símbolo **#** símbolos y espacios. La última línea no va precedida de ningún espacio. Escriba un programa que imprima una escalera de tamaño **n**. Formato de entrada: Un único entero, **n**, que denota el tamaño de la escalera.

Restricciones: $0 < n < 100$. La escalera debe estar alineada a la derecha.

Ejemplo:



(Escalera de tamaño $n = 4$)

- **Agente B:**
La cima y altura son ambas iguales a **n**. Se dibuja usando el símbolo **#** símbolos y espacios. La primera línea no va precedida de ningún espacio. Escriba un programa que imprima una escalera de tamaño **n**. Formato de entrada: Un único entero, **n**, que denota el tamaño de la escalera.

Restricciones: $0 < n < 100$. La escalera debe estar alineada a la derecha.

Ejemplo:

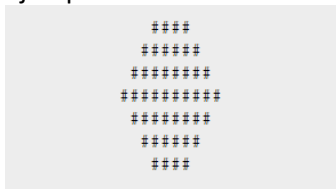


(Escalera de tamaño $n = 4$)

- **Agente C:**
La base y cima son ambas iguales a **n**. Entre los extremos superior e inferior debe haber una distancia igual a **n** con el centro. Se dibuja usando el símbolo **#** símbolos y espacios. La línea central no va precedida de ningún espacio. Escriba un programa que imprima una escalera de tamaño **n**. Formato de entrada: Un único entero, **n**, que denota el tamaño de la primera y última línea de la escalera.

Restricciones: $0 < n < 100$. La escalera debe estar alineada al centro.

Ejemplo



(Escalera de tamaño $n = 4$)

Criterios de evaluación:

- Aplicar patrones de diseño que habilite la selección dinámica de los agentes.
- Implementar Test unitarios
- Aplicar principios de Clean architecture
- Aplicar principios y buenas prácticas de desarrollo
- Implementar en tecnología .NET C#

Entrega del proyecto:

- Este proyecto debe ser accesible en GitHub como repositorio público.
- Tiempo de entrega 24H

Solución parte 1

Tecnologías, lenguajes de programación y frameworks a utilizar

Para el desarrollo de la solución planteada el diseño de una API RESTFULL en la parte de backend para que gestione las solicitudes enviadas desde un browser o una app en un dispositivo móvil, debido a que puede brindarnos más escalabilidad, sostenibilidad en el tiempo y será fácil de entender. Para su desarrollo se usará el lenguaje de programación C#, ofreciendo mayor seguridad y con el podemos construir una solución a gran escala, y estructurada.

Los frameworks a usar serán .Net para el back que brinda una amplia gama de herramientas para la creación de aplicaciones web, permitiendo agilidad, robustez y seguridad al construir las soluciones, al tener compatibilidad con una gran variedad de sistemas operativos podemos desplegar en Linux, MacOS o Windows, también en la nube de nuestra preferencia. La tecnología que usaremos para crear nuestra interfaz de usuario será Angular, un framework respaldado por Google, sinónimo de garantía, que nos permite el diseño de aplicativos web single-response, permitiéndonos crear sitios web robustos y de alta calidad, en tiempo récord de fácil manutención y con una alta compatibilidad con .Net; el lenguaje a utilizar será TypeScript. Se usará como ORM Entity Framework Core, que nos permite hacer conexiones a cualquier motor de base de datos, donde nos permite manejar las búsquedas y acciones a las tables como si fueran objetos dentro de nuestro back permitiéndonos una comunicación más ágil y sencilla con la base de datos. Se usa Linq para realizar alguna de las funcionalidades que se proponen en los requerimientos, es sencillo la manipulación de estructura de datos con este framework, ofreciéndonos una gran gama de herramientas. También se usará AutoMapper para manejar las respuesta de nuestra api sin necesidad de exponer más información que la que necesitara el usuario, mediaTr nos permitirá gestionar con mayor facilidad las solicitudes que lleguen a la aplicación. FluentValidation nos ayudara a gestionar las solicitudes validando los campos que si cumplan los requerimientos evitándonos procesos innecesarios.

Metodología de desarrollo

La metodología a usar será scrum, donde se hará reuniones periódicas de 15 días para establecer los alcances del Sprint, y reuniones diarias para saber las novedades que se puedan presentar y solicitud de auxilio, para el control de las tareas asignadas se usara Monday, donde se registrara la descripción y el tiempo de ejecución de cada etapa.

Descripción de las buenas prácticas metodológicas que pueden agilizar el proceso de desarrollo, para la entrega de software de calidad en los diferentes ambientes

Para el desarrollo del backend se usará una arquitectura onion, el cual organiza el proyecto por capas dando mayor entendimiento a nuestro código una mejor estructura, mayor escalabilidad, y mantenible en el tiempo, también se usará patrones de diseño como repository, mediator, specification, singulary, dependency injection, unitofwork, unittest, e integrationtest.

Para el frontend usaremos arquitectura lift que es la recomendada por angular para el manejo de aplicaciones web robustas, y los estilos scss patrón que nos ayuda a darle mayor calidad a los estilos de nuestras páginas.

Se usará azure devops para la construcción de pipelines que nos permitan instanciación de nuestras soluciones de manera automática y aplicaremos el software sonar para validar la calidad de nuestro código, configurado kpis concordados con el cliente.

Infraestructura y plataformas necesaria para soportar el desarrollo

Para el motor de base de datos se usará SQL server, por la gran cantidad de herramientas que nos provee para la protección, manipulación, visualización, análisis de datos.

Docker nos ayudara a contener nuestras soluciones en ambiente Linux, nos ayudara con el rendimiento y optimización de la solución.

Kubernetes ayuda a mitigar la afluencia de usuarios consumidores de la aplicación, dando elasticidad a la arquitectura y asegurándonos que la solución no colapse por exceso de solicitudes.

Devops Azure automatiza los procesos de instanciación de nuevas versiones de la solución, en los ambientes que se desea instanciar ya sea desarrollo, QA o producción.

GitHub nos ayudará con el trabajo colaborativo entre el equipo de desarrollo, sin necesidad de repetir trabajo y obteniendo actualización del desarrollo en tiempo real.

Posibles riesgos que pueden materializarse en la ejecución del proyecto y cómo mitigarlos

Cantidad de solicitudes que la solución no pueda gestionar, para eso configuramos Kubernetes sobre nuestros contenedores, volviendo la aplicación elástica donde levantara servicios y establecerá recursos para cumplir con la demanda y cuando baje automáticamente matara los servicios que levante sin causar sobrecostos en recursos no usados en periodos de demanda mínima.

Definir certificados de seguridad para evitar hackeos y proveedor a la solución de Tokens de seguridad para asegurarnos que el usuario que está manipulando la aplicación si sea un usuario habilitado y no un agente externo.