



PROCESO SELECCIÓN PERSONAL .NET

PRUEBA TÉCNICA

Objetivo

Esta prueba busca evaluar las competencias técnicas y de conocimiento al cargo de desarrollador .Net. Se evalúan las siguientes características:

CONOCIMIENTOS FUNDAMENTALES

1. Manejo del IDE Visual Studio, de sus herramientas, del entorno de diseño y del entorno de depuración. (Visual Studio 2019 o superior)
2. Manejo de C# y de .Net
3. Estructuración del código. Buenas prácticas de codificación y documentación
4. Programación Orientada a Objetos, programación orientada a eventos
5. Manejo de principios S.O.L.I.D
6. Manejo de herramientas ORM como Entity Framework, Hibernate o similares
7. Manejo de sistemas de versionamiento de código y trabajo cooperativo

CONOCIMIENTOS DESEABLES

8. Diseño de pruebas unitarias sobre .Net
9. Manejo básico de tecnologías de frontend: jQuery, Angular, REACT, Vue
10. Manejo de componentes de computación en la Nube (Azure, AWS, GCP)
11. Manejo de SQL-TRANSACT, SQL SERVER 2018+

Dinámica de la prueba

Se te entrega una descripción de un reto de desarrollo que involucra los elementos descritos en el objetivo de este documento.

En la prueba se evaluarán los siguientes puntos:

1. Tiempo de ejecución de la prueba
2. Funcionalidad lograda
3. Organización (modularidad y comprensión del código)
4. Arquitectura y calidad de software

Recuerda que estamos buscando una persona con muy buenos conocimientos en el manejo de C# y .Net, por lo que esta prueba será tu principal carta de presentación. Te recomendamos leer detenidamente el problema y desarrollar cada requerimiento con todos los detalles solicitados.

Crea un repositorio GIT al iniciar la ejecución de la prueba y compártelo con el contacto proporcionado (), esto facilitará la revisión incluso sin finalizar la prueba. Te sugerimos realizar múltiples commits con el fin de conocer el avance del desarrollo.

Si consideras que no alcanzará el tiempo para finalizar la prueba agrega en comentarios el cómo lo harías en un caso real.

Tiempo de desarrollo de la prueba



Aunque la prueba está diseñada para realizarse en un lapso de 4 a 5 horas, te otorgamos **2 días (calendario)** en los que podrás realizar la prueba. Con esto esperamos que organices tu disponibilidad horaria como lo consideres necesario y aproveches el tiempo restante para optimizar tu prueba (realizar pruebas, documentar, organizar el código)

Términos y condiciones

1. No tendrás ningún tipo de restricciones en las páginas que desees consultar, ni en el código de ejemplo que consideres utilizar siempre y cuando éste no incumpla ninguna ley de derechos de autor. En el caso de utilizar código de ejemplo deberás incluir las referencias al origen del código (URL, proyecto, autor)
2. Al realizar la prueba aceptas que podrás ser cuestionado acerca del código implementado demostrando el manejo y entendimiento del mismo.
3. Todo el código deberá escrito en INGLÉS, aunque los comentarios podrán ser escritos en el idioma de tu preferencia.

Recursos

1. Te sugerimos utilizar una plantilla de proyecto que promueva patrones de arquitectura como DDD y CLEAN, a continuación te dejamos algunos links de referencia:
 - a.
 - b.
 - c.
 - d.



INSTRUCTIVO

Problema

La empresa TEKUS S.A.S. requiere un aplicativo web (.NET Asp.net o .Net Aps.net Core) que permita administrar (crear, editar y listar) los servicios ofrecidos por sus proveedores. Cada proveedor deberá relacionar los servicios ofrecidos y a su vez, cada servicio deberá relacionar uno o varios países dónde es ofrecido (por ejemplo, El proveedor "Importaciones Tekus S.A." ofrece los servicios "Descarga espacial de contenidos" y "Desaparición forzada de bytes", y el servicio "Descarga espacial de contenidos" es ofrecido en Colombia, Perú y México por el proveedor "Importaciones Tekus S.A.S.")).

- El proveedor deberá tener cómo mínimo los campos ID, NIT, Nombre y correo electrónico.
 - El servicio deberá tener cómo mínimo los campos ID, Nombre y valor por hora (en \$USD).
 - El campo ID en ambas entidades deberá ser un valor autogenerado y único.
 - La lista de países deberá ser consultada desde un servicio externo.
 - Agregue las columnas que considere necesarias para garantizar la funcionalidad solicitada en los siguientes puntos.
1. Aunque es un proyecto de una estructura simple, plantea una solución estructurada, separa los proyectos, considera utilizar patrones de diseño DDD (domain driven design).
 2. Deberás proporcionar un mecanismo de autenticación. No es necesario desarrollar ninguna funcionalidad para crear o administrar usuarios
 3. El usuario podrá agregar campos personalizados sobre cada proveedor, por ejemplo, para el proveedor "Importaciones Tekus S.A" se podría agregar el campo "Número de contacto en marte" o el campo "Cantidad de mascotas en la nómina".
 4. Deberás proveer un endpoint del API que entregue un resumen que permita visualizar 2 indicadores que consideres importante mostrar, por ejemplo: a) Cantidad de clientes por país, b) Cantidad de servicios por país

Requerimientos generales:

1. **Todos los servicios deberán ser expuestos como un API-RESTful**
2. No es necesario crear un front, es completamente opcional, si lo deseas podrás utilizando cualquier lenguaje de frontend (Razor, Native HTML + Javascript, ReactJs, Angular, Vue, etc)
3. Toda lista deberá tener funcionalidades de paginación, ordenamiento y búsqueda.
4. Agrega todas las validaciones que considere necesarias para garantizar el correcto formato de los datos.
5. Documenta las funcionalidades que considere más importantes, aunque consideramos que la mejor documentación siempre será un código bien estructurado
6. Construye las pruebas unitarias y de integración que consideres necesarias. Estas pruebas sólo deberán estar dirigidas al back.

Entregables

1. Deberás entregar un esquema de la Base de datos relacionando las entidades mencionadas (una foto o un diagrama)



2. No recibimos ninguna prueba por correo electrónico o archivo comprimido. Es por esto que deberás publicar el proyecto completo en un repositorio GIT de tu preferencia (Proyecto ejecutable en Visual Studio 2017 o superior). Recuerda enviar el link del repositorio a _____ al iniciar la prueba para facilitar la revisión de los avances y enviar cuantos commits consideres necesarios con el fin de facilitar la revisión de la prueba.
3. Deberás proveer un script de generación de la DB en el mismo repositorio del punto anterior, así como un script de datos iniciales (al menos 10 por cada tabla).
4. Envía un correo a jose.angarita@meeteam.co notificando la finalización de la prueba y cualquier consideración adicional que sea necesaria para facilitar la revisión.