PRUEBA PARA INGRESO A TREDA SOLUTIONS

DESARROLLADOR WEB JUNIOR C#

Bienvenido al proceso de admisiones de Treda Solutions. Gracias por querer hacer parte de nuestro equipo.

Prerrequisitos:

- Tener instalado Visual Studio 2019. https://visualstudio.microsoft.com/vs/community/
- Tener instalado SQL Server 2019 Developer. https://www.microsoft.com/en-us/sql-server/sql-server-downloads
- Tener instalado Git. https://git-scm.com/downloads

Recomendaciones:

- Puede utilizar cualquier base de datos e IDE que conozca.
- Puede utilizar cualquier cliente de git que conozca. (Consola, GitHub, Source Tree Control, Git Kraken).

Contenido de la prueba:

1. Prueba teórica de C#
2. Prueba práctica de C#
3. Prueba de lógica de programación
4. Prueba SQL

Si tiene alguna duda o quiere aclarar algún punto, por favor no dude en contactarme al correo csandoval@tredasolutions.com

PRUEBA PARA INGRESO A TREDA SOLUTIONS

PRUEBA TEÓRICA DE C#

1. Cuál es la salida del siguiente programa, explique. ¿Cambiaría la respuesta si reemplazamos Task.Delay(5) con Thread.Sleep(5)?

```
class xxx {
  private static string res;

  static void Main() {
    method();
    Console.WriteLine(res);
  }

  static async Task<string> method() {
    await Task.Delay(5);
    res = "Hola Mundo!";
    return “Algo”;
  }
}
```

2. ¿Es posible almacenar tipos de datos mixtos como int, string, float, char, todo en una matriz?
3. ¿Qué diferencia hay entre un struct y clases en C#?
4. Dada una cadena de palabras con uno o varios $, ejemplo “Bienvenido a $ treda $ $” ¿Cómo se puede eliminar la segunda y tercera ocurrencia del símbolo, por medio de expresiones regulares?

PRUEBA PARA INGRESO A TREDA SOLUTIONS

PRUEBA PRÁCTICA DE C#

1. Se debe crear un CRUD en C# que satisfaga las necesidades que se describirán.
2. Para desarrollar este punto usted debe usar .NET Framework (Versión más reciente), o .NET Core (Versión más reciente).
3. Todas las consultas a la base de datos deben ser ejecutadas por Entity Framework.
4. El modelo de bases de datos debe ser creado a partir de las clases y luego migrado con las funcionalidades de Entity Framework (Code First Migrations).

Necesidad

Se desea almacenar la información de tiendas y de los productos en cada tienda. La información que se quiere almacenar es:

Tienda	Producto
ID Nombre Fecha de apertura	Nombre SKU Descripción valor Tienda Imagen

- Suponga una relación 1:N entre Tienda y Producto.
- Todos los campos son obligatorios.
- El campo SKU de Producto debe ser único. El ID de la tienda también es único
- El campo imagen debe contener la ruta de la imagen en filesystem.

5. Teniendo como base el primer punto, desarrolle un servicio SOAP o REST que dado el ID de la tienda muestre los productos asociados a esta.
6. Se debe retornar en un JSON la información y la imagen se debe retornar en base 64.
7. El servicio debe tener un log de eventos (El método de su elección). Con el objetivo de conocer cuáles peticiones se han realizado en el servicio.

Nota importante:

- El código debe ser entregado en un repositorio de Github, donde se puedan visualizar cada uno de los commit realizados, comentarios de los mismos, ramas realizadas. No se permite que toda la aplicación tenga un solo commit.
- El repositorio debe contar con instrucciones para su instalación y ejecución.
- Se debe tener por lo menos una petición asíncrona en la aplicación.
- En el backend se debe validar los valores numéricos y el campo fecha tenga el formato dd-mm-YYYY. Esto se debe hacer por medio de expresiones regulares.
- Es deseable que se realice en .Net Core.

PRUEBA PARA INGRESO A TREDA SOLUTIONS

Puntos Extras

Se darán puntos extras a las aplicaciones que tengan lo siguiente.

- Recaptcha en algún formulario de ingreso.
- Utilizar librerías de CSS como bootstrap.
- Realizar la interfaz con Vuejs o React JS.

PRUEBA PARA INGRESO A TREDA SOLUTIONS

PRUEBA DE LÓGICA DE PROGRAMACIÓN.

Para los siguientes puntos se debe construir un método funcional en C# que cumpla las necesidades dadas. Ejemplo:

```
public static SalidaEsperada metodoX(Parametros datos en el problema)
  {
      //Ingrese el código necesario para que cumpla las necesidades
  }
```

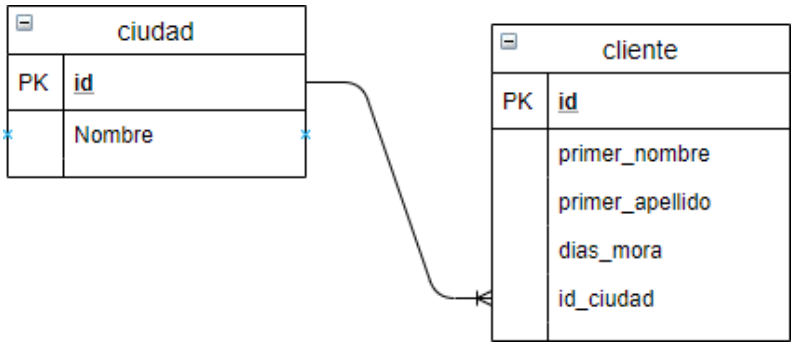
1. Dado un número **n** encontrar todos los múltiplos de 3 y 5 menores al número dado, el método debe retornar la suma de los múltiplos encontrados. Ejemplo: si el número ingresado es 10, los múltiplos de 3 y 5 menores a 10 son: 3, 5, 6, 9, el resultado de la función debe ser 23, debido a que es la suma de 3, 5, 6, 9.
2. Dado un string separado por espacios, guiones y guiones bajos. Se debe implementar una función camel case que transforme la oración.
   Ejemplos
   a. Dado “Bienvenido a_Treda-solutions” retornar “BienvenidoATredaSolutions”
   b. Dado “bienvenido-a_Treda solutions” retornar “bienvenidoATredaSolutions”
3. Dada una frase, devolver la frase donde las palabras con mayor a 5 letras esten al revés. Ejemplos. Dado “Bienvenido a Treda Solutions” retornar “odinevneiB a Treda snoituloS”

PRUEBA PARA INGRESO A TREDA SOLUTIONS

PRUEBA DE SQL

Reporte 1

Dato el siguiente modelo ER



Generar la consulta SQL (script) que retorna un listado con los clientes que se encuentren en mora e indicar en una columna el nivel de riesgo que dicha mora representa según las siguientes reglas:

Mora entre **1** y **20** días:	**Riesgo Bajo**
Mora entre **21** y **35** días:	**Riesgo Medio**
Mora Mayor a **35** días:	**Riesgo Alto**

El reporte deberá contener las siguientes columnas:

cedula	nombre	diasEnMora	riesgo	ciudad

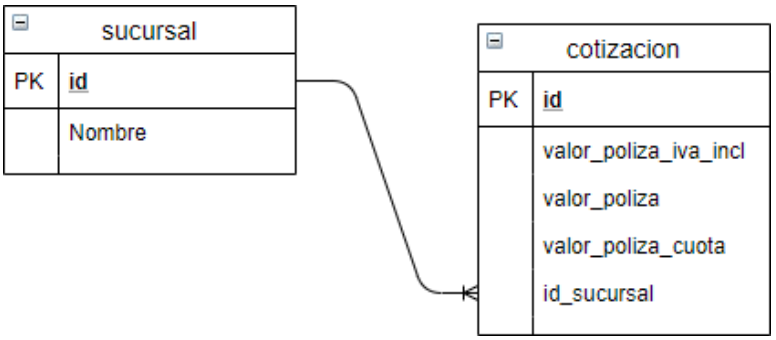
Notas

- El nombre debe ser la concatenación del primer nombre y el primer apellido del cliente.
- El reporte debe estar ordenado de menor a mayor por la cantidad de días en mora.
- En la columna *ciudad* se espera el nombre de la ciudad relacionada.

PRUEBA PARA INGRESO A TREDA SOLUTIONS

Reporte 2

Dado el siguiente ER



Generar la consulta SQL (script) que retorna un listado con las sucursales y el valor **total** de las pólizas (IVA incluido) pagadas por cada una.

El reporte deberá contener las siguientes columnas:

sucursal	valorTotalPagado

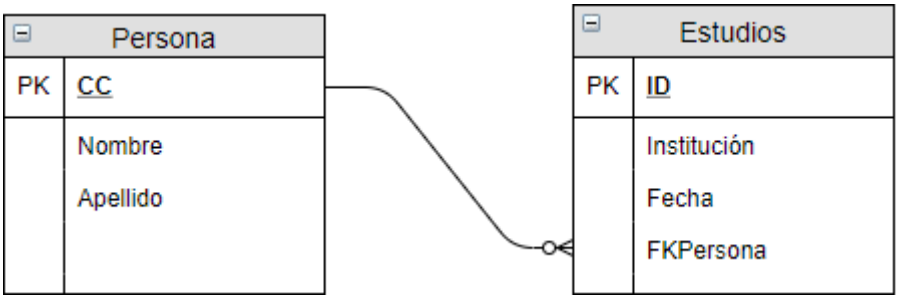
Notas

- El reporte debe estar ordenado de menor a mayor por el valor total pagado.
- En la columna *sucursal* se espera el nombre de la sucursal relacionada.
- El valor total pagado deberá estar debidamente formateado como número con 0 decimales.

PRUEBA PARA INGRESO A TREDA SOLUTIONS

Reporte 3

Se tiene el siguiente modelo E-R



Generar la consulta SQL (script) que retorna un listado de personas con el último estudio realizado según la fecha. El reporte solo debe mostrar un estudio por persona sin repetir.

CC	Nombre	Institución	Fecha

PRUEBA PARA INGRESO A TREDA SOLUTIONS

Reporte 4

Se tiene la siguiente definición

```sql
create table empleados (
    cc int(11) not null primary key,
    nombre varchar(50),
    cargo varchar(50),
    area varchar(50),
    id_jefe int(11),
    constraint jefe_fk foreign key (id_jefe) references empleados (cc)
);
```

Generar la consulta SQL (script) que retorna un listado de personas con su respectivo jefe.

CC	nombre	cargo	área	Nombre Jefe