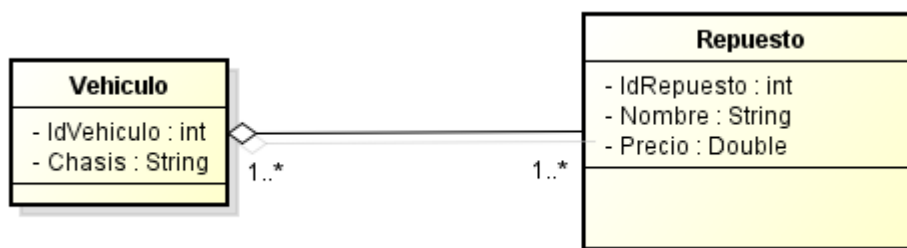


<b>EVALUACION</b>	PARCIAL	<b>GRUPO</b>	61242-P3 - N3C	<b>FECHA</b>	12/05/22
<b>MATERIA</b>	PROGRAMACIÓN 3				
<b>CARRERA</b>	AP/ATI				
<b>CONDICIONES</b>	- Puntos: 35 - Duración: 2 Horas - Otros: Sin material  Consultas solamente sobre interpretación de letra y sintaxis específica del lenguaje.				

1. Escriba los esquemas de las tablas de una base de datos que permita mapear las clases del diagrama (5pts)



2. Escriba un método en el repositorio adecuado que permita obtener todos los repuestos mediante ADO.NET (8 pts)
3. Si el servidor de base de datos es un SQL Server con ip 127.0.0.1, el nombre de la base de datos es TallerDB, el nombre de la instancia SQLSERVER\_R19, se accede a él mediante seguridad integrada de Windows, complete el connection string: (4 pts)

SqlConnection con = new SqlConnection(@"data source= \_\_\_\_\_ \ \_\_\_\_\_;  
database=\_\_\_\_\_; integrated security=\_\_\_\_\_;  
TrustServerCertificate=True");

4. En un controlador MVC de Repuesto, implemente la acción HTTPGet que permita listar todos los repuestos. Asuma que tiene implementado el repositorio de Repuesto y el método GetAll() que devuelve todos los repuestos. (5 pts)
5. Dadas las siguientes clases:

1. Complete los campos en la vista de MVC que realiza el alta del cliente. (5 pts)

```

public class Cliente
{
    public int Id { get; set; }
    public string Nombre { get; set; }
    public string Apellido { get; set; }
    public TipoCliente TipoCliente { get; set; }
}

public class TipoCliente
{
    public int Id { get; set; }
    public string NombreTipoCliente { get; set; }
}
  
```

@model ejemplo.mvc.Cliente

```
<h1>Nuevo Cliente</h1>

<div class="row">
  <div class="col-md-4">
    <form asp-action="Create">
      <div class="form-group">
        <label asp-for=" " class="control-label"></label>
        <input asp-for=" " class="form-control" />
      </div>
      <div class="form-group">
        <label asp-for=" " class="control-label"></label>
        <input asp-for=" " class="form-control" />
      </div>

      <div class="form-group">
        <label asp-for=" " class="control-label"></label>
        @Html.DropDownListFor(model => model. .Id,
          new SelectList(@ViewBag.TiposClientes, "Id", "NombreTipoCliente"),
            "Selecione...",
            new { @class = "form-control" })
      </div>
      <div class="form-group">
        <input type="submit" value="Guardar" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>
```

2. Escriba el método HttpGet en el controlador de cliente que permita cargar la vista que realiza el alta. Asuma que tiene implementado el repositorio de Tipo de Cliente y el método GetAll() que devuelve todos los Tipos de Clientes. **(3 pts)**

6. Dado el siguiente código c# y las siguientes entradas, indique cual es la salida por pantalla. **(5 pts)**

```
public class Operaciones
{
    public int Division(int x, int y)
    {
        try
        {
            return x / y;
        }
        catch (DivideByZeroException)
        {
            throw;
        }
        finally
        {
            Console.WriteLine("Finally: Nunca se ejecuta!!!");
        }
    }
}
```

```
static void Main(string[] args)
{
    string entrada;
    Console.WriteLine("Ingrese un valor:");
    entrada = Console.ReadLine();
    int denominador = 0;
    Operaciones operaciones = new Operaciones();
    try
    {
        denominador = int.Parse(entrada);
        int resultado = operaciones.Division(200, denominador);
        Console.WriteLine($"El resultado de dividir 200 entre {denominador} es {resultado}");
    }
    catch (DivideByZeroException)
    {
        Console.WriteLine("DivideByZeroException: No es posible dividir por 0");
    }
    catch (OverflowException)
    {
        Console.WriteLine("OverflowException: El número ingresado no se puede convertir a Int32");
    }
    catch (Exception)
    {
        Console.WriteLine("Exception: Otras excepciones");
    }
    finally
    {
        Console.WriteLine("Finally: Siempre se ejecuta!!!");
    }
}
```

Desde la consola se digitan los siguientes valores:

- 0
- 2
- int.MinValue