

EVALUACION	Examen Cursos dictados desde Marzo 2019 en adelante	FECHA	18/02/2020
MATERIA	PROGRAMACIÓN 2		
CARRERA	Analista Programador / Analista en Tecnologías de la Información		
CONDICIONES	<ul style="list-style-type: none"> - Puntos: 100 - Duración: 3 Hrs - Sin material - No escriba la hoja de la letra - Consultas solamente sobre interpretación de la letra y sintáxis específica del lenguaje. 		

Ejercicio 1

TELEBARRERA es un novedoso sistema de gestión para peajes que permite el cobro automático a los vehículos afiliados. De cada vehículo se conoce su matrícula, marca, año, tipo y su propietario. De cada tipo de vehículo se conoce un código, descripción y precio base de peaje. Los propietarios pueden ser Empresas o Particulares. En el caso de las empresas se conoce su RUT, razón social, dirección, teléfono de contacto, porcentaje base de descuento (que es el mismo para todos los propietarios empresa) y tipo de convenio (1-Comun, 2-ORO, 3-Platinum). De los propietarios particulares se conoce su cédula, nombre, dirección, teléfono de contacto y si aplica descuento por cercanía. Todos los propietarios tienen un código.

Este sistema se está probando solo en uno de los peajes. Cuando un vehículo afiliado se acerca al peaje se levanta la barrera y automáticamente se registra una pasada en el sistema en el cual se almacena la fecha, hora y el vehículo que acaba de pasar.

El costo de cada pasaje depende del tipo de vehículo y de quien es su propietario. Se calcula de la siguiente forma:

Si el propietario del vehículo es un particular se cobra el precio base del vehículo según el tipo y en caso de que aplique descuento por cercanía, sobre este precio se aplica un 10 % de descuento.

Si el propietario del vehículo es una empresa, al precio base del vehículo del vehículo se le aplica el porcentaje base de descuento definido mas un descuento adicional según el tipo de convenio:

- Común: No aplica descuento adicional.
- Oro: Aplica un 10 % de descuento si el año del vehículo posterior al 2015 y 15 % si el vehículo es del año 2018 o más nuevo.
- Platinum: Aplica un 20 % de descuento y un 5 % adicional para los vehículos cuyo precio base de peaje es mayor a 200 pesos.

Sin importar si el propietario es particular o empresa, sobre el costo a pagar se le aplica un descuento adicional de un 2 % en caso de que el vehículo tenga 50 pasadas o mas registradas.

Se pide:

- a) Diagrama de clases completo que permita resolver los siguientes requerimientos: **20 puntos**

R01-Obtener el vehículo que tenga la mayor cantidad de pasadas registradas por el peaje. Si hay más de uno con la cantidad máxima devolver todos los mayores.

R02-Dado un vehículo y una fecha inicial y final, obtener el costo total a cobrar por las pasadas realizadas por ese vehículo en el peaje en las fechas indicadas.

R03- Dado un propietario listar las pasadas de sus vehículos por el peaje realizados entre dos fechas dadas, ordenados por fecha en forma descendente.

- b) Implementar solo a nivel de dominio los requerimientos R01 **(15 puntos)**, R02 **(20 puntos)** y R03 **(15 puntos)**

Importante: En el requerimiento R02 es obligatorio usar polimorfismo para su implementación, en caso de no usarlo no se obtendrán puntos por la resolución de ese requerimiento.

Ejercicio 2 (20 puntos)

Tomando en cuenta el diagrama de clases definido en el ejercicio 1 parte a), implementar lo que considere necesario a nivel de la vista y el controlador para listar todos los tipos de vehículo registrados, mostrando de cada uno todos sus datos.

Ejercicio 3 (10 puntos)

Indique cuál o cuáles de las siguientes afirmaciones son verdaderas justificando la respuesta:

- i. Si trabajo con el modelo a nivel de MVC, y tengo la siguiente estructura en la clase que utilizo como modelo y en la vista:

```
// Modelo
public class Persona

    private int numeroPersona;
    private string nombre;
    private string documento;

    public Persona(int numeroPersona, string nombre, string apellido){
        this.numeroPersona=numeroPersona;
        this.nombre=nombre;
        this.apellido=apellido;
    }
}

//Vista
@model Proyecto.Persona

<form method="post" action="~/Persona/Alta">
<label>Ingrese número de persona</label>
<input type="number" value=@Model.NumeroPersona name="numeroPersona">
<label>Ingrese nombre de la persona</label>
<input type="text" value=@Model.NombrePersona name="nombrePersona">
<label>Ingrese apellido de persona</label>
<input type="text" value=@Model.ApellidoPersona name="apellidoPersona">
<input type="submit" value="Guardar">
</form>
```

- a) La clase que utilizo como modelo, posee la estructura correcta
 - b) La implementación no es correcta, para que funcione es necesario agregar en la clase Persona, las properties de los atributos numeroPersona, nombre y apellido y un constructor sin parámetros
 - c) La implementación no es correcta, solamente es necesario agregar en la clase Persona un constructor sin parámetros
 - d) La implementación no es correcta, solamente es necesario borrar de la clase Persona el constructor con parámetros
 - e) La implementación realizada a nivel de la vista y la clase Persona es correcta.
- ii. El ViewData es utilizado para almacenar datos.
- a) Se utiliza para pasar datos del controlador a la vista
 - b) Dura durante todos los request
 - c) Puede almacenar solamente datos numéricos o string.
 - d) Puede ser usado en el dominio.
- iii. Se tiene la siguiente URL: localhost.8888/Productos/Listado?categoria=Comestibles
- a) Se tiene que haber implementado un controlador Producto con el método Index
 - b) Se guarda la información en un ViewBag con el nombre Listado
 - c) Se tiene un controlador Productos y una acción llamada listado por el metodo get sin parámetros
 - d) Se tiene un controlador Productos y una acción llamada listado con el parámetro categoría