



Reto 3 – Alquiler de Autos

Objetivo:

El objetivo de este reto es que el estudiante reconozca y aplique los elementos básicos del paradigma de la programación orientada a objetos en un escenario abstraído de la cotidianidad.

Contexto:

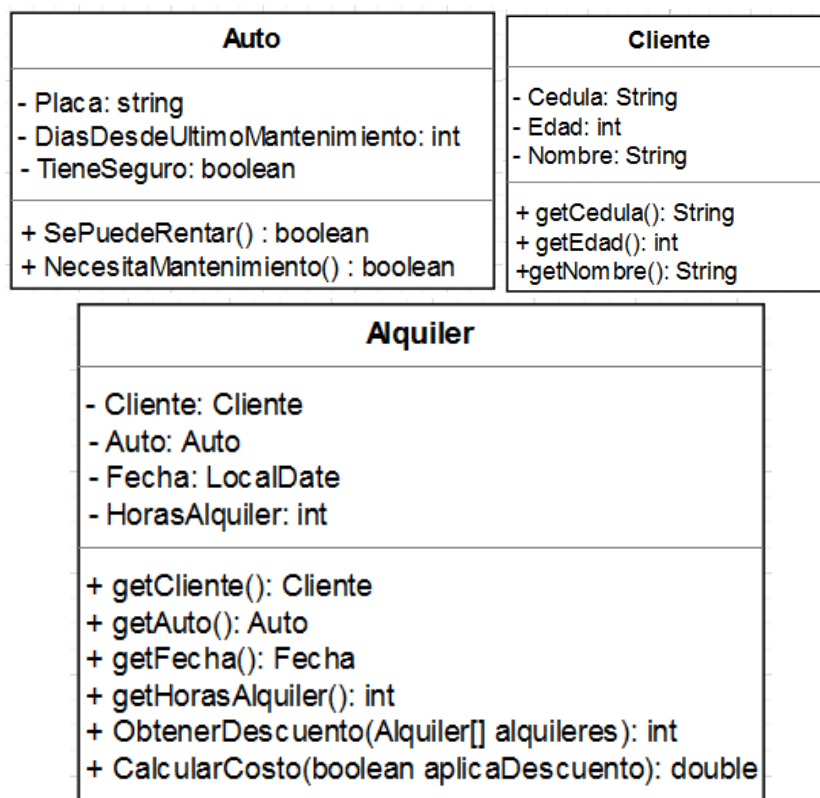
Nuevas sedes del concesionario están teniendo una buena acogida y han atraído una gran cantidad de clientes nuevos. Debido a esto la carga laboral de los empleados de la empresa ha aumentado mucho y la cantidad de errores humanos ha estado aumentando día a día debido a la gran demanda de vehículos y los diferentes factores que se deben tener en cuenta a la hora de calcular el costo de un alquiler.

Por estas razones la empresa le solicita que implemente en el sistema una funcionalidad para calcular el costo de un alquiler. La empresa le informa que su modelo de cobro actual es hacer un cobro por horas de alquiler. Además, han modificado la política de descuentos pues consideran que las sedes están cerca de su máxima capacidad y atraer más clientes en este momento sería contraproducente, por lo que ahora se aplica un descuento fijo del 2% a la hora de calcular el costo de un alquiler. Sin embargo, le solicitan que mantenga la funcionalidad implementada en el reto anterior para hacer uso de esta en un futuro cercano.



Reto:

1. Implemente las clases que se muestran en el siguiente diagrama. Note que las clases **Auto** y **Cliente** son las mismas implementadas en el reto anterior, por lo que se recomienda reutilizarlas.





2. La clase **Alquiler** también es similar a la implementada en el reto anterior, se recomienda modificarla para que se adapte al nuevo diagrama. Note que ahora implementa una nueva función **CalcularCosto** que retorna un valor de tipo **double** que sería el costo del alquiler, también tiene un parámetro de tipo **boolean** que representa que el cliente ha cumplido con cierta cantidad de horas de alquiler en el mes y puede aplicar a un descuento. Para calcular el costo de un alquiler tenga en cuenta las siguientes consideraciones:
 - a. El costo de una hora de alquiler es de 5000 Pesos Colombianos.
 - b. Si el valor del argumento de la función es **verdadero**, significa que el cliente aplica para un descuento del 2%. En caso de ser **falso** el descuento es 0%.
 - c. La función retorna el costo total, entendido como el valor total de las horas del alquiler tras aplicar el descuento.



Casos de Prueba:

Para validar el correcto funcionamiento del programa considere los siguientes escenarios:

Caso de Prueba	Datos de Entrada	Salida Esperada		
1. CalcularCosto(true)	<table><tr><td>Alquiler</td></tr><tr><td>Cliente: Cliente1 Auto: Auto Fecha: 19/08/2021 HorasAlquiler: 8</td></tr></table>	Alquiler	Cliente: Cliente1 Auto: Auto Fecha: 19/08/2021 HorasAlquiler: 8	39.200
Alquiler				
Cliente: Cliente1 Auto: Auto Fecha: 19/08/2021 HorasAlquiler: 8				
2. CalcularCosto(false)	<table><tr><td>Alquiler</td></tr><tr><td>Cliente: Cliente1 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 8</td></tr></table>	Alquiler	Cliente: Cliente1 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 8	40.000
Alquiler				
Cliente: Cliente1 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 8				
3. CalcularCosto(true)	<table><tr><td>Alquiler</td></tr><tr><td>Cliente: Cliente Auto: Auto Fecha:1/08/2021 HorasAlquiler: 124</td></tr></table>	Alquiler	Cliente: Cliente Auto: Auto Fecha:1/08/2021 HorasAlquiler: 124	607.600
Alquiler				
Cliente: Cliente Auto: Auto Fecha:1/08/2021 HorasAlquiler: 124				



En la calificación de este reto se evaluará que las funcionalidades implementadas en el reto anterior sigan funcionando correctamente.

Para un historial de alquileres como el siguiente:

0	1	2	3	4
Cliente: Cliente1 Auto: Auto Fecha: 12/06/2021 HorasAlquiler: 48	Cliente: Cliente2 Auto: Auto Fecha: 12/07/2021 HorasAlquiler: 30	Cliente: Cliente1 Auto: Auto Fecha: 14/07/2021 HorasAlquiler: 25	Cliente: Cliente3 Auto: Auto Fecha: 14/07/2021 HorasAlquiler: 12	Cliente: Cliente1 Auto: Auto Fecha: 16/07/2021 HorasAlquiler: 8

Caso de Prueba	Datos de Entrada	Salida Esperada				
4. ObtenerDescuento(historial)	<table><tr><td>Alquiler</td></tr><tr><td>Cliente: Cliente1 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 26</td></tr><tr><td>Auto</td></tr><tr><td>Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: True</td></tr></table>	Alquiler	Cliente: Cliente1 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 26	Auto	Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: True	5
Alquiler						
Cliente: Cliente1 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 26						
Auto						
Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: True						
5. ObtenerDescuento(historial)	<table><tr><td>Alquiler</td></tr><tr><td>Cliente: Cliente2 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 12</td></tr><tr><td>Auto</td></tr><tr><td>Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: True</td></tr></table>	Alquiler	Cliente: Cliente2 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 12	Auto	Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: True	2
Alquiler						
Cliente: Cliente2 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 12						
Auto						
Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: True						



ENTREGA:

1. Los archivos que suba a la plataforma para su calificación deben llamarse **exactamente** *Auto.java*, *Cliente.java* y *Alquiler.java*, de lo contrario no se calificará.
2. Los nombres de las clases, miembros dato y funciones deben llamarse **exactamente** como se muestran en los diagramas mostrados al comienzo del reto, las firmas de sus clases deben ser cómo se muestra en las siguientes imágenes:

```
public class Cliente {  
    private String Cedula;  
    private int Edad;  
    private String Nombre;  
  
    public Cliente(String cedula, int edad, String nombre) {  
        //Implementación  
    }  
  
    public String getCedula() {  
        //Implementación  
    }  
  
    public int getEdad() {  
        //Implementación  
    }  
  
    public String getNombre() {  
        //Implementación  
    }  
}
```



```
public class Auto {  
    private String Placa;  
    private int DiasDesdeUltimoMantenimiento;  
    private boolean TieneSeguro;  
  
    public boolean NecesitaMantenimiento(){  
        //Implementación  
    }  
  
    public boolean SePuedeRentar(){  
        //Implementación  
    }  
}
```



```
import java.time.LocalDate;
public class Alquiler {
    private Cliente Cliente;
    private Auto Auto;
    private LocalDate Fecha;
    private int HorasAlquiler;

    public Alquiler(Cliente cliente, Auto auto, LocalDate fecha, int horasAlquiler) {
        //Implementación
    }

    public Cliente getCliente() {
        //Implementación
    }

    public Auto getAuto() {
        //Implementación
    }

    public LocalDate getFecha() {
        //Implementación
    }

    public int getHorasAlquiler() {
        //Implementación
    }

    public int ObtenerDescuento(Alquiler[] alquileres){
        //Implementación
    }

    public double CalcularCosto(boolean aplicaDescuento){
        //Implementación
    }
}
```