



Reto 2 – Alquiler de Autos

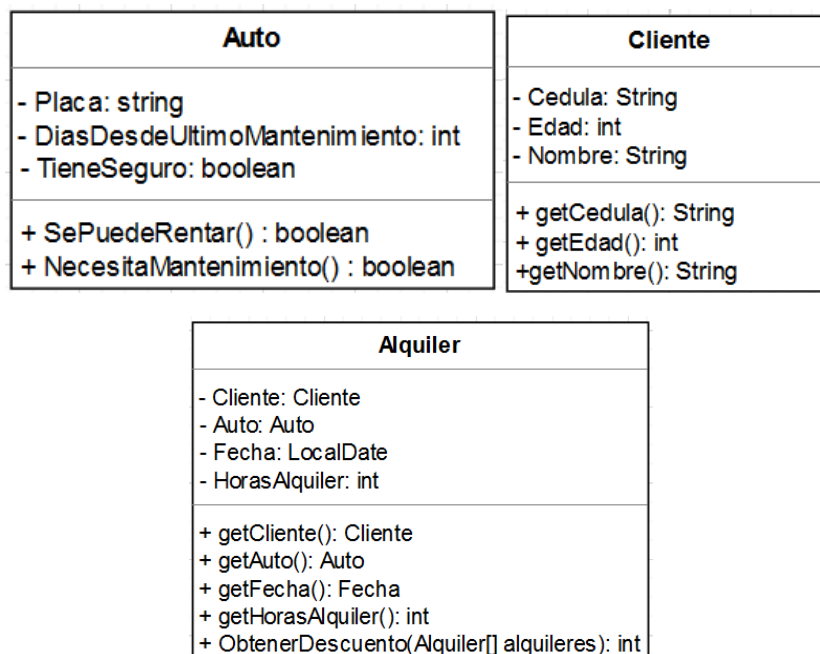
Objetivo:

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:

Gracias al sistema implementado en el reto pasado la empresa de renta de autos ha logrado reducir los costos de mantenimiento y gastos de reparación de sus vehículos. Debido a esto han decidido expandirse y abrir varias sedes nuevas junto con una promoción para promover que las personas conozcan las nuevas ubicaciones.

Para llevar a cabo la expansión la empresa le comenta que necesita implementar un sistema como el del siguiente diagrama.





Reto:

1. Implemente las clases que se muestran en el diagrama, la clase **Cliente** implementa un constructor y los respectivos **getters**. Note que la clase **Auto** es la misma creada en el reto anterior, se recomienda reutilizarla.
2. La clase **Alquiler** implementa un constructor y sus respectivos **getters**.
3. **Alquiler** también implementa el método **ObtenerDescuento** que retorna un valor tipo **int**. Para calcular el descuento de la instancia desde la que se invoca el método la empresa establece la siguiente lógica de negocio:
 - a. Antes de calcular cualquier descuento debe verificar que el auto de la instancia no necesite mantenimiento y se pueda rentar, en caso de que alguna de estas condiciones no cumpla retorna 0.
 - b. La empresa cuenta con una base de datos con los registros de cada vez que un cliente renta un auto, por lo que la función recibe un arreglo de tipo **Alquiler** que representa ese historial de alquileres. Tenga en cuenta que este arreglo es un historial puede contener alquileres de diferentes clientes, recuerde validar que los alquileres son del mismo cliente para el que está calculando el descuento.
 - c. El método debe calcular un descuento en función de la cantidad de horas acumuladas que el cliente tenga en sus alquileres de los últimos 30 días, contando las horas la instancia desde el que invoca el método. (Asuma que la fecha actual es el valor **Fecha** de la misma instancia). El descuento se asigna de acuerdo a la siguiente tabla:

Horas Acumuladas	Valor Descuento
Menos de 20	0
Entre 20 y 50	2
Más de 50	5

Nota: Puede profundizar en el funcionamiento del objeto **LocalDate** en este [link](#).



Casos de Prueba:

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

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

Caso de Prueba	Datos de Entrada	Salida Esperada
1. ObtenerDescuento(historial) Menos de 20 horas acumuladas	<div>Alquiler</div> <div>Cliente: Cliente1 Auto: Auto Fecha: 19/08/2021 HorasAlquiler: 8</div> <div>Auto</div> <div>Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: True</div>	0
2. ObtenerDescuento(historial) Auto no es válido	<div>Alquiler</div> <div>Cliente: Cliente1 Auto: Auto Fecha: 19/07/2021 HorasAlquiler: 8</div> <div>Auto</div> <div>Placa: "DBZ 645" DiasDesdeUltimoMantenimiento: 2 TieneSeguro: False</div>	0



Caso de Prueba	Datos de Entrada	Salida Esperada				
3. 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						
4. 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
    }
}
```