



Reto 3 – Renta de Motos Acuáticas

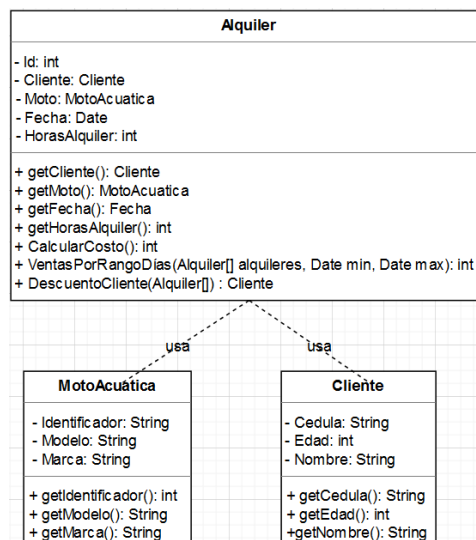
Objetivo:

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:

El alquiler de motos acuáticas a turistas en temporada alta ha sido un gran éxito, sin embargo, para el resto del año las ganancias de las empresas se han visto bastante reducidas, al punto en que no es rentable para la empresa seguir operando. Los administradores de estas empresas creen que el problema puede ser que las personas que viven cerca no tienen tanto interés en alquilar una moto acuática durante el día, por lo que han diseñado una estrategia de marketing para atraer a este público en particular.

La estrategia sería que cada día se selecciona a la persona que haya hecho el alquiler corto del día para darle un pequeño descuento en su próximo alquiler. Le proponen modificar el sistema que ya tiene implementado de la siguiente manera:





Reto:

Debe implementar las clases cómo se indica en el diagrama teniendo en cuenta las siguientes consideraciones:

1. Note que las clases **MotoAcuatica** y **Cliente** son las mismas que se implementaron en los retos anteriores, se recomienda reutilizarlas.
2. La clase **Alquiler** en su mayor parte también es igual a la implementada en el reto anterior. Se recomienda modificarla para adaptarse al nuevo diagrama. Note que ahora implementa un **getter getHorasAlquiler** que retorna el miembro dato **HorasAlquiler**.
3. **Alquiler** también implementa una nueva función estática **DescuentoCliente** que toma por argumento un arreglo de tipo **Alquiler** y debe retornar un objeto de tipo **Cliente**. Para seleccionar el cliente que retorna la función valide las siguientes instrucciones:
 - a. La función debe buscar en el arreglo el **Alquiler** que tenga la menor cantidad de horas **Y** que el cliente asociado a este alquiler sea mayor de edad.
 - b. La función debe retornar entonces el **Cliente** que está asociado con ese alquiler.
 - c. Si ninguno de los clientes de estos alquileres es mayor de edad (Edad mayor o igual a 18) entonces debe retornar **Null**.
 - d. Puede asumir que el arreglo de alquileres contiene solo uno o ningún **Alquiler** válido para aplicar el descuento.



Casos de Prueba:

Puede verificar el correcto funcionamiento de su programa con los siguientes escenarios:

Caso de Prueba	Datos de Entrada			Salida Esperada
1.	Alquiler	Alquiler	Alquiler	Cliente Cedula: 35214 Edad: 43 Nombre: Juan
	Id: 23 Cliente: Cliente Moto: MotoAcuatica Fecha: 31/07/2021 HorasAlquiler: 2	Id: 24 Cliente: Cliente Moto: MotoAcuatica Fecha: 31/07/2021 HorasAlquiler: 2	Id: 24 Cliente: Cliente Moto: MotoAcuatica Fecha: 31/07/2021 HorasAlquiler: 4	
	Cliente	Cliente	Cliente	
	Cedula: 15536614 Edad: 14 Nombre: Juanito	Cedula: 35214 Edad: 43 Nombre: Juan	Cedula: 121344 Edad: 23 Nombre: Carlos	
2.	Alquiler	Alquiler	Alquiler	Null
	Id: 23 Cliente: Cliente Moto: MotoAcuatica Fecha: 1/08/2021 HorasAlquiler: 4	Id: 24 Cliente: Cliente Moto: MotoAcuatica Fecha: 1/08/2021 HorasAlquiler: 4	Id: 24 Cliente: Cliente Moto: MotoAcuatica Fecha: 31/08/2021 HorasAlquiler: 4	
	Cliente	Cliente	Cliente	
	Cedula: 15536614 Edad: 14 Nombre: Juanito	Cedula: 35214 Edad: 16 Nombre: Juan	Cedula: 121344 Edad: 17 Nombre: Carlos	
3.	Alquiler	Alquiler	Alquiler	Cliente Cedula: 121344 Edad: 32 Nombre: Felipe
	Id: 23 Cliente: Cliente Moto: MotoAcuatica Fecha: 2/08/2021 HorasAlquiler: 5	Id: 24 Cliente: Cliente Moto: MotoAcuatica Fecha: 2/08/2021 HorasAlquiler: 4	Id: 24 Cliente: Cliente Moto: MotoAcuatica Fecha: 2/08/2021 HorasAlquiler: 1	
	Cliente	Cliente	Cliente	
	Cedula: 15536614 Edad: 18 Nombre: Alexander	Cedula: 35214 Edad: 26 Nombre: Aaron	Cedula: 121344 Edad: 32 Nombre: Felipe	



ENTREGA:

1. Los archivos que suba a la plataforma para su calificación deben llamarse **exactamente** *Cliente.java*, *MotoAcuatica.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:

```
import java.util.Date;

public class Alquiler {
    private int Id;
    private Cliente Cliente;
    private MotoAcuatica Moto;
    private Date Fecha;
    private int HorasAlquiler;

    public Alquiler(int id, Cliente cliente, MotoAcuatica moto, int horasAlquiler) {
        //Implementación
    }

    public Alquiler(int id, Cliente cliente, MotoAcuatica moto, Date fecha, int horasAlquiler) {
        //Implementación
    }

    public Cliente getCliente() {
        //Implementación
    }

    public MotoAcuatica getMoto() {
        //Implementación
    }

    public Date getFecha() {
        //Implementación
    }

    public int getHorasAlquiler() {
        //Implementación
    }

    public int CalcularCosto(){
        //Implementación
    }

    public static int VentasPorDias(Alquiler[] alquileres, Date min, Date max){
        //Implementación
    }

    public static Cliente DescuentoCliente(Alquiler[] alquileres){
        //Implementación
    }
}
```



```
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 MotoAcuatica {  
    private String Identificador;  
    private String Modelo;  
    private String Marca;  
  
    public MotoAcuatica(String identificador, String modelo, String marca) {  
        //Implementación  
    }  
  
    public String getIdentificador() {  
        //Implementación  
    }  
  
    public String getModelo() {  
        //Implementación  
    }  
  
    public String getMarca() {  
        //Implementación  
    }  
}
```