

ARRIENDO DE VEHÍCULOS – COLABORACIÓN o ASOCIACIÓN

ENUNCIADO:

Una empresa de arriendo de vehículos ha decidido implementar un sistema que permita administrar información de sus arriendos. Para el Arriendo, se requiere almacenar *el vehículo arrendado, el cliente que lo arrendó, fecha arriendo, medio de pago ('T' o 'E'*, donde 'T' significa Tarjeta y 'E' significa Efectivo), *cantidad de días arrendado* (un vehículo no puede ser arrendado por más de 10 días). Para el Vehículo se debe almacenar la *patente (largo 8)*, marca, modelo, *año* fabricación (entre 2000 y 2020) y la situación del vehículo que indique si está arrendado o no (**true**: si está arrendado, **false**: no arrendado). Del Cliente se requiere almacenar el rut (con largo 10 y penúltimo dígito debe ser un guion '- '), nombre y si el cliente está vigente o no (**true**: si está vigente, **false**: no vigente).

Cuando un vehículo es creado (ingresado al sistema), la situación del vehículo inicialmente debe ser asignada **false**. Cuando un cliente es creado (ingresado al sistema), su vigencia inicialmente es asignada **true**. Por otro lado, los valores de la patente, el modelo y marca siempre deben ser guardados con mayúscula.

El sistema debe cubrir las siguientes necesidades del usuario implementados a través de métodos customer en la clase Arriendo:

- i) Registrar vehículo arrendado: Que se asegure que el cliente esté vigente y que el vehículo no esté arrendado. Si cumple ambas condiciones que registre la situación del vehículo como arrendado y retorne un true, en caso contrario que no haga nada y que retorne un false.
- ii) Obtener valor del arriendo: El valor arriendo = cantidad de días * precio por día. Considere precio por día recibido por parámetro.
- iii) Generar Ficha de Arriendo: que reciba por parámetro el precio por día y que permita mostrar la información en el formato que indica el ejemplo siguiente:

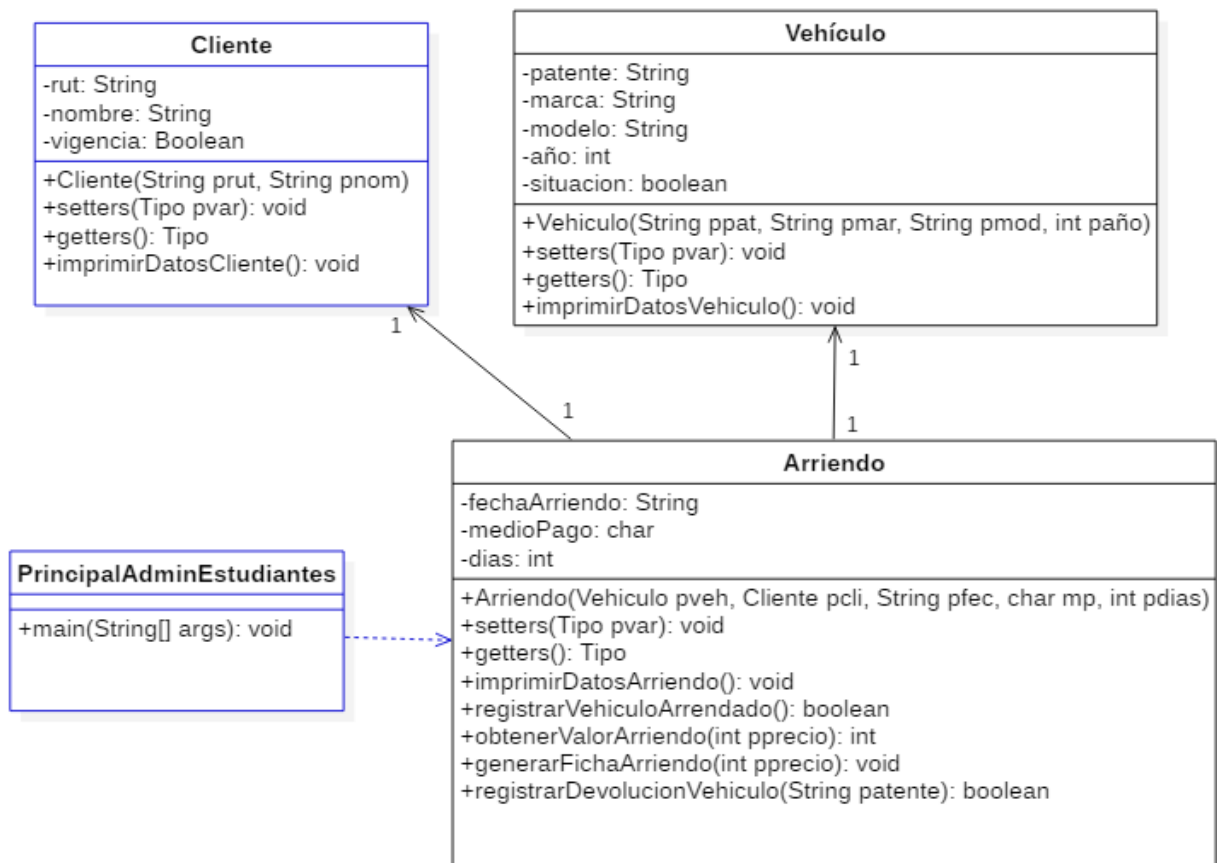
FICHA ARRIENDO DE VEHÍCULO

VEHÍCULO : CZSL60-6 Toyota Corolla
PRECIO POR DIA : \$25000

FECHA ARRIENDO	RUT CLIENTE	NOMBRE CLIENTE	DIAS ARIENDO	VALOR ARRIENDO	MEDIO DE PAGO
10/09/2019	20023066-3	Juan Perez	3 días	\$75000	Tarjeta

- iv) Registrar Devolución del Vehículo: que reciba por parámetro una patente y que esta patente sea igual a la almacenada en el arriendo registrado (o creado) y que esté registrada con situación arrendado. Si es correcta que actualice el vehículo con situación disponible (o no arrendado) y que retorne un true, en caso contrario que no haga nada y que retorne un falso.

El Diagrama de Clases (DC) que representa lo descrito en el enunciado anterior es:



HAGA LO SIGUIENTE PASO A PASO:

- 1) Programe el DC definido anteriormente (use sólo el constructor definido en el DC).
- 2) Para probar el funcionamiento del sistema, programe el método main con las siguientes operaciones:
 - a) Crear un arriendo. Los datos requeridos asígnelos directamente para probar su sistema y luego muestre los datos del arriendo.
 - b) Registre el vehículo arrendado. Si es satisfactorio (true) enviar mensaje de que vehículo ha sido arrendado, en caso contrario (false) enviar mensaje correspondiente. En el caso de que el vehículo se haya registrado satisfactoriamente generar la ficha de arriendo. Revise que el formato de la ficha de arriendo sea similar al formato mostrado en el customer iii)
 - c) Probar la ejecución de la devolución del vehículo.
- 3) Después que haya realizado lo anterior haga lo siguiente:
 - a) Incorpore las validaciones en los setters correspondientes, pero use funciones (métodos) para cada validación. Por lo tanto, haga lo siguiente:
 - Cada set que requiera una validación debe llamar a una función (método) que valide el atributo. El resultado del método será un true o un false. Cuando es true que el set haga la actualización del valor del atributo, en caso contrario que no lo actualice.
 - Cree una clase llamada **Validation**. Esta clase tendrá varios métodos (public static), cada uno recibirá un parámetro con el valor a validar, realizará la validación correspondiente y enviará un true o false según corresponda.
 - b) Modifique el main incorporando un menú con 4 opciones:
 1. Arrendar vehículo → 2) a) y b)
 2. Devolver vehículo → 2) c)
 3. Invalidar un Cliente → poner al cliente vigente
 3. Salir.

Su programa no debe terminar hasta ingresar la opción salir. Incorpore entrada de datos por teclado. Las entradas deben ser validadas usando la clase **Validation**.