

APTC101 Guía de aprendizaje 2

Semana 10

Sumativa 4: Guía de aprendizaje 2

DESCRIPCIÓN DE LA ENTREGA:

Los alumnos resolverán individualmente el caso que se describe más adelante en este documento, que consiste en aplicar los conceptos de la programación orientada a eventos junto con la POO revisados en la presente unidad. Al término del desarrollo del desafío, antes que se cumpla el plazo de entrega, debe subir a la plataforma habilitada para tal efecto, una carpeta comprimida (con su nombre) con el siguiente producto:

- ***Carpeta con los programas del proyecto Java creado.*** Incluya en la carpeta un readme.txt que explique las consideraciones a tener en cuenta al ejecutar su proyecto.

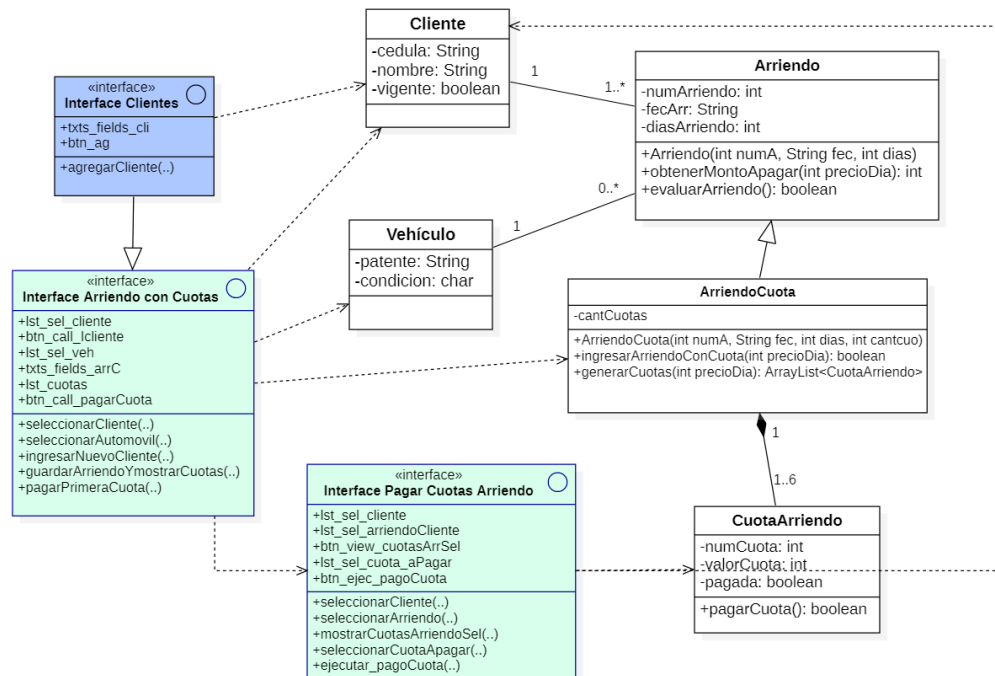
OTRAS CONSIDERACIONES:

- El docente responderá las preguntas de los estudiantes y efectuará el monitoreo del avance en el foro dispuesto para este efecto (en la etapa Ponte a prueba), realizando preguntas sobre los conceptos claves para solucionar los requerimientos de información planteados. Además, incentivará la participación en la búsqueda de buenas soluciones.
- El estudiante puede utilizar cualquier ambiente de desarrollo que le permita programar en Lenguaje Java. En el readme.txt, de la carpeta donde estarán sus programas, indique el **IDE** y las consideraciones que debe tener el evaluador para ejecutar su programa.
- El docente evaluará según los aspectos que se indican en la rúbrica que puede descargar de la plataforma.

ENUNCIADO DEL CASO

La empresa de arriendo de vehículos **"Car-REnt"** ha decidido implementar un sistema informático que le permitirá mejorar la gestión de su empresa. Hasta el momento, ha diseñado la solución y ha avanzado en la construcción usando la POO; ahora, debe implementar la interfaz usando la Programación Orientada a Eventos (POE). Por lo tanto, el gerente lo ha contratado a usted, nuevamente, para programar el arriendo de vehículos con cuotas con los respectivos pagos que realizan los clientes. La especificación de lo que el gerente requiere se detalla a continuación:

- El diagrama de clases para esta parte específica del proyecto se muestra a continuación, donde la capa de presentación (vista del usuario) se destaca con color azul. A través de estas interfaces el usuario podrá generar los eventos que deberá programar usando la POE. Los eventos necesarios para cubrir los requerimientos se deben controlar usando las operaciones definidas en esta misma interfaz. Además, usted debe programar las clases del diagrama con sus componentes necesarios usando la POO:



En este diagrama solo se han mostrado los elementos para la comprensión de la especificación del requerimiento. Asuma que los constructores, setters, getters, toString están siendo considerados de acuerdo con los principios de la POO. Los métodos customer que debe considerar en su implementación (y que están representados en el mismo diagrama) son:

- a) Obtener monto a pagar: Este método calcula y entrega el monto del arriendo instanciado multiplicando el número de días por el precio diario. Este precio lo indica el usuario una vez que se ingresa un arriendo al sistema.
 - b) Evaluar arriendo: Operación que se ejecuta antes de guardar el arriendo al sistema y valida que el cliente del arriendo esté vigente y que el vehículo de este mismo arriendo tenga condición D (disponible). El método retorna un true (si está ok) o false (si no es posible arrendar).
 - c) Generar cuotas del arriendo: operación que se ejecuta al guardar un arriendo exitoso al sistema, que recibe el precio por día del arriendo y crea las cuotas según la siguiente especificación:
 - ☐ El número de cada cuota es un correlativo que comienza en 1 y se aumenta en uno para las cuotas siguientes.
 - ☐ El valor de cada cuota es la división entre monto a pagar y cantidad de cuotas.
 - ☐ Todas las cuotas deben quedar inicialmente con pagada igual a False.
 - ☐ Este método debe retornar la lista (ArrayList) de cuotas generada para que sea asignada (o referenciada) al arriendo respectivo.
 - d) Ingresar arriendo con cuotas: Este método evalúa los datos del arriendo instanciado (invocando la función evaluarArriendo) retornando true si la operación fue exitosa y false si no lo fue. En el caso de que la operación sea exitosa, este método automáticamente dejará el vehículo arrendado con condición A y para aplicar correctamente la relación compuesta entre los objetos arriendo y cuotas, asignará las cuotas respectivas del arriendo invocando la función definida en c).
 - e) Pagar cuota: operación que recibe la cuota a pagar y busca la cuota en la lista respectiva. Si la encuentra, el método actualiza el atributo pagada con True y retorna un true, en caso contrario, retorna un false.
- Además, se requiere que su implementación considere lo siguiente:
 - a) Las validaciones de los atributos que se implementan en los mutadores, se deben programar usando métodos públicos en la clase respectiva. Estas validaciones se ejecutan en el momento de actualizar (setear) el objeto, por lo tanto, debe considerar, además, que los datos ingresados por el usuario (en el caso que corresponda) y que son ocupados para instanciar los objetos, podrán ser validados usando estos mismos métodos.
 - b) Para que los objetos del sistema entreguen al usuario los diversos mensajes generados por el sistema, programar en cada clase un método que reciba el string con el mensaje, el tipo de mensaje y genere la salida.

- La interfaz con la que interactúa el usuario (representada en el diagrama) para ingresar un arriendo con cuotas se muestra a continuación.

ARRIENDOS CON CUOTAS

Fecha Arriendo:

Días:

Precio por día:

MONTO A PAGAR:

Cantidad de cuotas:

CUOTAS A PAGAR

Número	Valor	¿pagada?

Las interacciones del usuario con la interfaz "Arriendo con Cuotas" se deben controlar considerando las asociaciones que se muestran en el diagrama. Al guardar el arriendo debe agregarlo al arraylist y al archivo correspondiente.

- La interfaz (representada en el diagrama) con la que interactúa el usuario para agregar un cliente se muestra a continuación:

CLIENTES

Cédula:

Nombre:

¿ Vigente ?:

Agregar

Después que el usuario agrega un cliente debe agregarlo al arraylist y archivo correspondiente y retornar a la interfaz de Arriendo con Cuotas.

- Para realizar el pago de una cuota, el usuario debe interactuar con la interfaz siguiente (representada en el diagrama) que le permita seleccionar un cliente, luego seleccionar uno de sus arriendos, enseguida pincha el número de cuota a pagar y, finalmente, ejecuta realizar pago.

PAGAR CUOTAS ARRIENDOS

Seleccione cliente

PAGOS

	Número	Valor	¿pagada?
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Seleccione arriendo:

Mostrar pagos arriendo seleccionado >>>

Realizar Pago

- La interfaz principal no se representa en el diagrama, por lo tanto, usted debe diseñarla y programarla considerando al menos lo siguiente:
 - ☐ Barra de menús con las opciones que el usuario puede operar (Arrendar con cuotas y pagar cuota). Implementar dichas opciones con controladores de los eventos generados por el usuario.
 - ☐ Título e imagen *ad hoc* para el sistema de arriendo de vehículos.
 - ☐ Que cargue los datos de los archivos clientes, vehículos y arriendos con cuota en los arrayList respectivos. Use POO.

- El método main debe invocar interfaz principal del sistema (la diseñada por usted) para empezar a operar el sistema con las condiciones especificadas en este documento.