

Desarrollo de sistemas con tecnología Java

Práctica 3

Nombre:

Apellido Paterno

Apellido Materno

Nombre(s)

Fecha:

Calificación:

Objetivo: El o la participante aplicará lo aprendido sobre clases, objetos, atributos, métodos y herencia en Java por medio de la resolución del problema planteado.

Instrucciones: Lea cuidadosamente y genere de manera individual el código para implementar las clases del siguiente modelo.

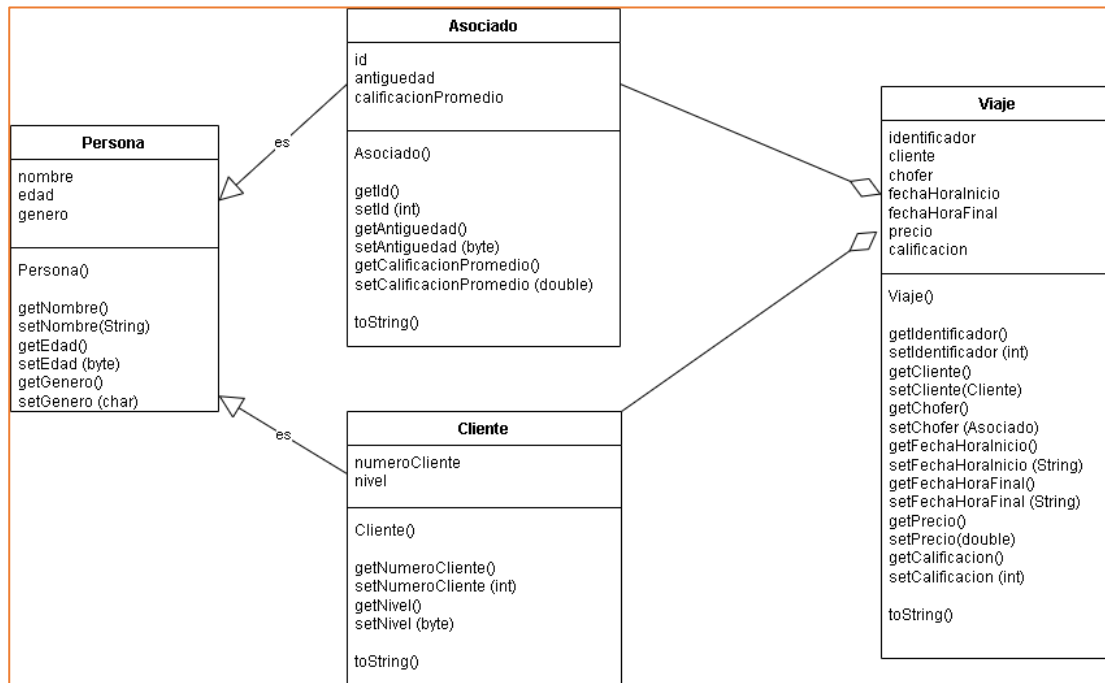


Ilustración 1 Modelo a implementar en Java

El modelo representa las clases necesarias para implementar un viaje en un taxi de aplicación. El modelo está simplificado para fines de esta actividad. Las clases, y las restricciones de los atributos que se deben implementar en el código, se enumeran a continuación:

1. Persona (clase de la cual luego se derivarán Asociado y Cliente)

- El nombre deberá tener una longitud de al menos 7 caracteres.
- La edad será de 7 o más años.
- El género válido será F, M, D o N (Femenino, Masculino, Desconocido, No quiso decirlo).

2. Cliente (subclase de Persona)

- Contará con un identificador positivo.
- El nivel será un valor entero entre 1 y 3 (nivel de suscripción).

3. *Asociado* (subclase de *Persona*)

- Tendrá un identificador que será un valor positivo entero.
- La antigüedad no podrá ser un valor negativo.
- Tendrá una calificación promedio que será un valor entre cero y cinco (la mayor calificación).
- La edad deberá ser mayor a 18 años y menor a 60.

4. *Viaje* (representa un viaje en el servicio de taxi de aplicación)

- Tendrá un identificador que deberá ser un valor positivo.
- Con un atributo de tipo *Cliente* se tendrá el cliente que realiza el viaje.
- El chofer estará representado con un atributo de tipo *Asociado*.
- La fecha-hora de Inicio y de Final del recorrido será, en esta versión, un valor de tipo *String*.
- El precio no podrá ser un valor negativo.
- La calificación que otorga en cliente será un valor entero entre de 0 y 5 (estrellas).

De cada clase será necesario, además, que:

- Los constructores asignen un valor a cada atributo.
- El método *toString* () regrese algunos valores de los atributos (puede tomarse como referencia la imagen del ejemplo).
- Se deberán incluir los métodos *set* y *get* para todos los atributos.

Para poder probar las clases generadas deberá crear un programa (**Practica3.java**) que instancie al menos una vez las clases *Cliente*, *Asociado* y *Viaje*, que modifique valores de algunos atributos (para verificar las restricciones) y que despliegue (haciendo uso del método *toString*) los objetos instanciados.

```
Prueba de clases de la práctica 3
Cliente:Cliente 0: Desconocido, nivel=1
Asociado: Asociado 1: Desconocido (D).0.0 estrellas promedio
Viaje INICIAL: Viaje 0
    Cliente=Cliente 0: Desconocido, nivel=1
    Chofer=Asociado 1: Desconocido (D).0.0 estrellas promedio
    $100.0 3 estrellas
Viaje FINAL: Viaje 0
    Cliente=Cliente 0: Chandler Bing, nivel=1
    Chofer=Asociado 1: Phoebe Buffay (F).0.0 estrellas promedio
    $100.0 3 estrellas
```

Ilustración 2 Ejemplo de ejecución de *Practica3.java*

Opcionales:

- Implementar los atributos de duración del viaje, domicilio y correo electrónico del cliente, así como el correo y el número telefónico del asociado.
- Añadir al modelo los datos del automóvil que proporciona el viaje. Determine si deberá añadirse el automóvil en forma de atributos en algunas de las clases existentes o por medio de una nueva clase.

Entregables

1. Persona.java
2. Cliente.java
3. Asociado.java
4. Viaje.java
5. Practica3.java

Los archivos deberán ser subidos a la plataforma del Diplomado en el apartado correspondiente.

Criterios de evaluación

La calificación que obtenga considerará los siguientes criterios y puntajes de evaluación para cada una de las tres clases asignadas:

| Elemento por evaluar | Porcentaje |
|---|-------------|
| Las clases tienen el nombre indicado | 10% |
| Las clases contienen los atributos y métodos con el nombre y tipo definidos | 20% |
| Las clases contienen al menos un constructor que asigne valores para todos los atributos de la clase | 20% |
| Se integran los métodos <i>set</i> y <i>get</i> para todos los atributos. En aquellos donde se requiere una restricción o validación, se integrará dicha validación en el método <i>set</i> correspondiente | 40% |
| El método <i>toString()</i> regresa todos los atributos de la clase | 10% |
| Total | 100% |

Nota: Para calcular la longitud de un *String* se podrá hacer uso del método *length* () de dicha clase, por ejemplo:

```
String texto="UNAM";  
System.out.println ("Longitud de "+texto+": "+ texto.length());
```

Genera la siguiente respuesta:

Longitud de UNAM: 4

Valor 100 puntos
