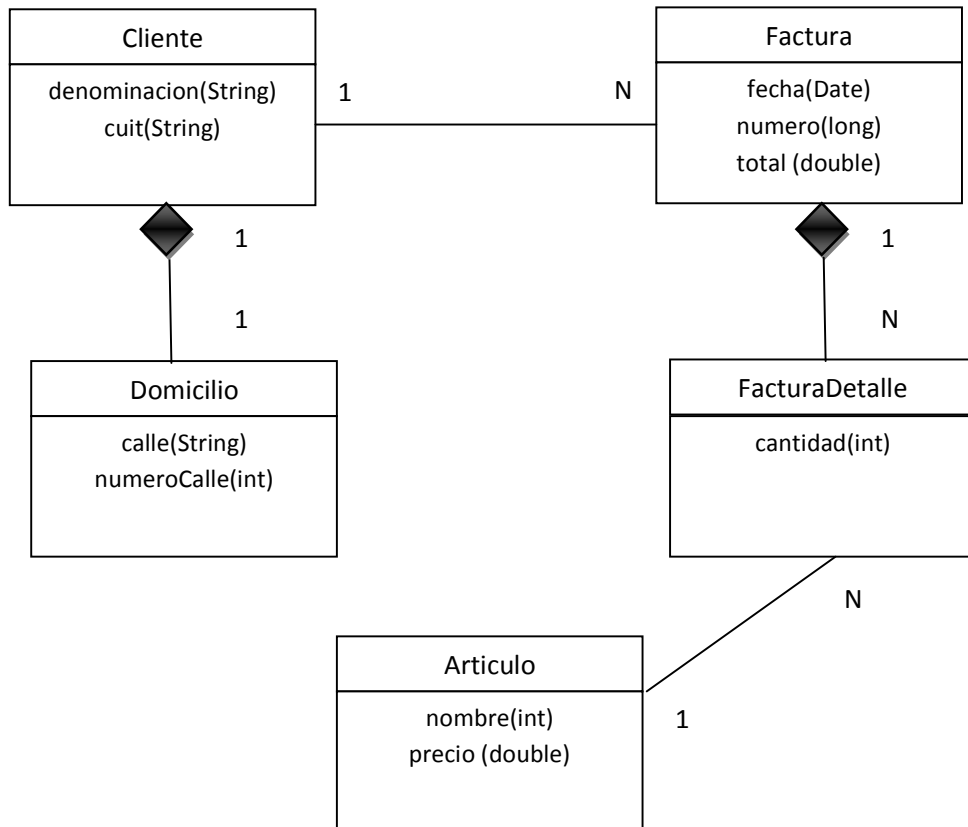


Trabajo Practico N° 2

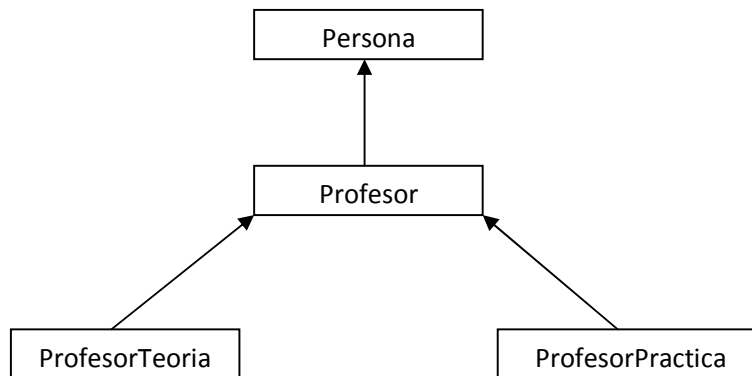
Programación 3 - TSP – FRM –UTN

EJERCICIO 1 –Composición.

Codifique el siguiente diagrama de clases, respetando los tipos de asociación y Cardinalidad.



EJERCICIO 2 – HERENCIA Y POLIMORFISMO - CAST

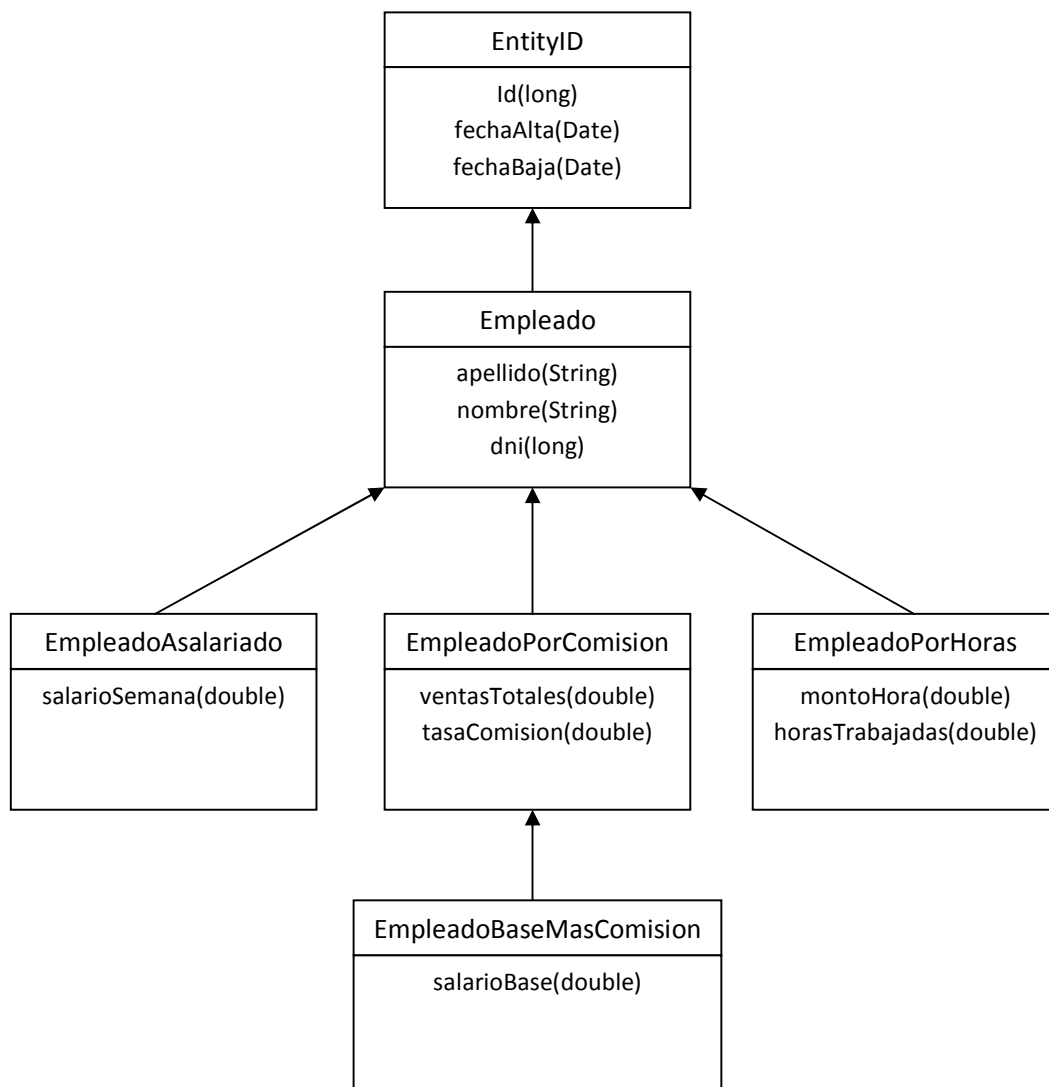


- a- Puede realizarse la conversión de un Profesor a un Profesor de Teoría?
Si/No Justifique y Ejemplifique
- b- Puede realizarse la conversión de un Profesor de Practica a Profesor?
Si/No Justifique y Ejemplifique
- c- Puede realizarse la conversión de un Profesor de Practica a un Profesor de Teoría?
Si/No Justifique

EJERCICIO 3 – HERENCIA Y POLIMORFISMO

Sistema de nómina utilizando Herencia y Polimorfismo.

Ahora aplicaremos métodos abstractos y polimorfismo para realizar cálculos de pago de sueldos a sus empleados. Una compañía paga a sus empleados en forma semanal. La compañía tiene 4 tipos de empleados: empleados asalariados que reciben un sueldo semanal fijo, sin importar el número de horas trabajadas; empleados por hora, que reciben un sueldo por hora y pago por tiempo extra; empleados por comisión, que reciben un porcentaje de sus ventas y el empleado asalariado por comisión, que recibe un salario base más un porcentaje de sus ventas. Para este período de pago, la compañía ha decidido compensar a los empleados por comisión agregando un 10% a sus salarios. La compañía desea implementar una aplicación en Java que realice sus cálculos de nómina en forma polimórfica



Utilizaremos la clase Empleado para representar a un empleado "genérico". Las clases que extienden a Empleado son EmpleadoAsalariado, EmpleadoPorComision y EmpleadoPorHoras. La clase EmpleadoBaseMasComision (que extiende a EmpleadoPorComisión) representa al último tipo de empleado. Un método **ingresosEmpleado** se aplica genéricamente a todos los empleados. Pero el cálculo de los ingresos de cada empleado depende de su clase. Por eso declaramos a **ingresosEmpleado** como un método abstracto en la superclase Empleado (ya que en esta clase no es apropiada ninguna implementación específica) y cada subclase sobrescribe a ingresos con una implementación apropiada. Para calcular los ingresos de un empleado, el programa asigna a una variable de la superclase la referencia al objeto de ese empleado, y después invoca al método ingresos del empleado.

Para cada una de las clases codifique el constructor sobrecargado correspondiente.

CLASE	ingresosEmpleado
EmpleadoAsalariado	Será igual al salario semanal asignado
EmpleadoPorComision	Sera igual a las ventas totales por su tasa de comisión
EmpleadoPorHoras	Sera igual al monto por hora por la cantidad de horas trabajadas.
EmpleadoBaseMasComision	Sera igual al salario base asignado mas las correspondientes comisiones alcanzadas.

Invoca al método **ingresosEmpleado** para cada una de las 4 instancias.