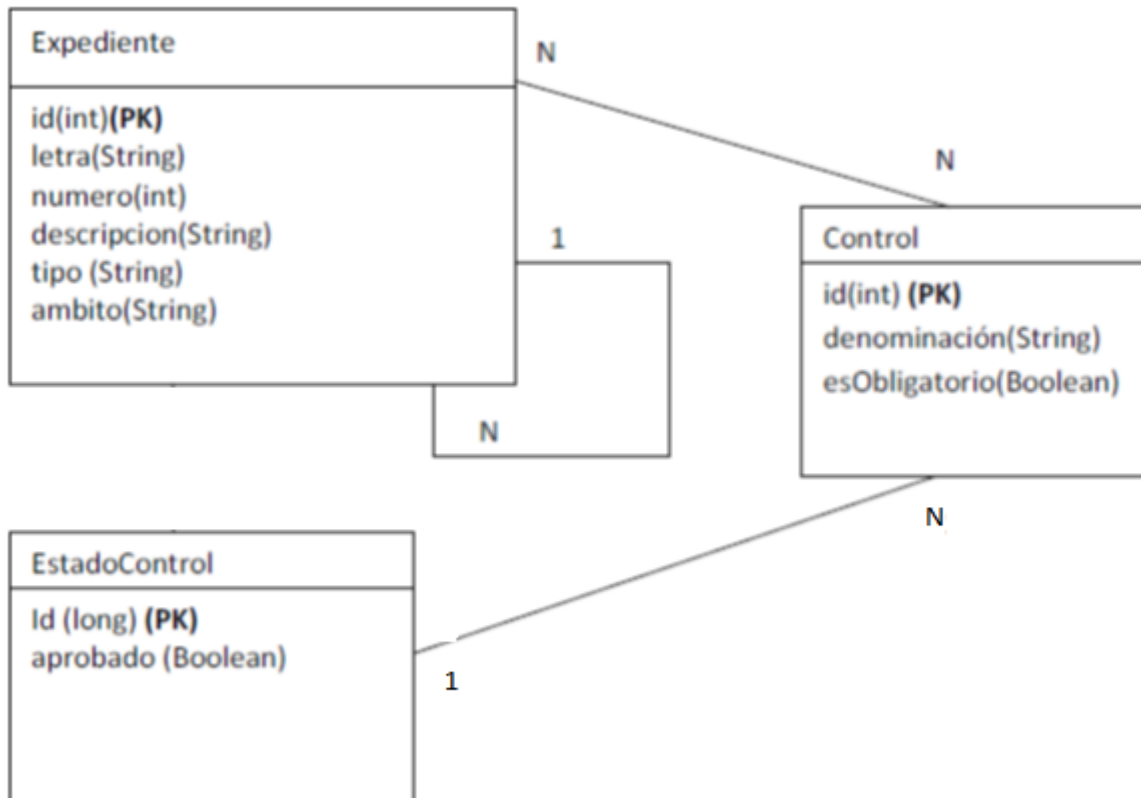


Programación 3 – TSP

Trabajo Práctico – Colaboración entre Objetos.

A)-



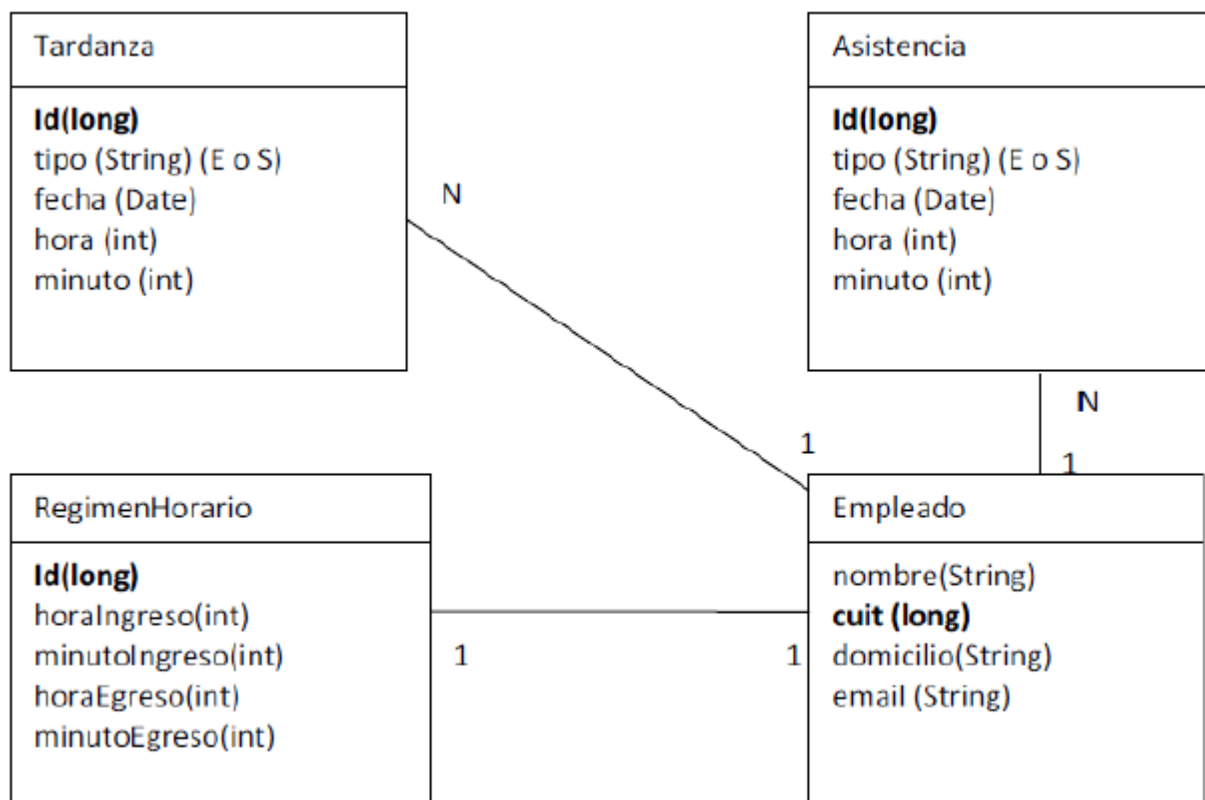
A.1- Codifique en la clase **Expediente** un método denominado “**public String getCaratulaExpediente()**” que retorne la concatenación de los campos número, letra y descripción separados con guion medio -

A.2- Codifique en la clase **Expediente** un método denominado “**public String getControlesObligatorios ()**” que retorne una lista separada con comas de la **denominación** de los controles **obligatorios** asociados al expedientes.

A.3- Codifique en la clase **Expediente** un método denominado “**public boolean getEstadoControles()**” que retorne **true** en caso de que todos los estados de los controles obligatorios se encuentren aprobados, caso contrario retorne **false**.

A.4- Codifique en la clase **Expediente** un método denominado “**public List<Expediente> listaExpedientes()**” que haciendo uso de **RECURSIÓN** retorne una lista de objetos **Expediente** asociados a él. **(Nota: para resolver el ejercicio es recomendable usar un método auxiliar que realice la recursión)**

B)-

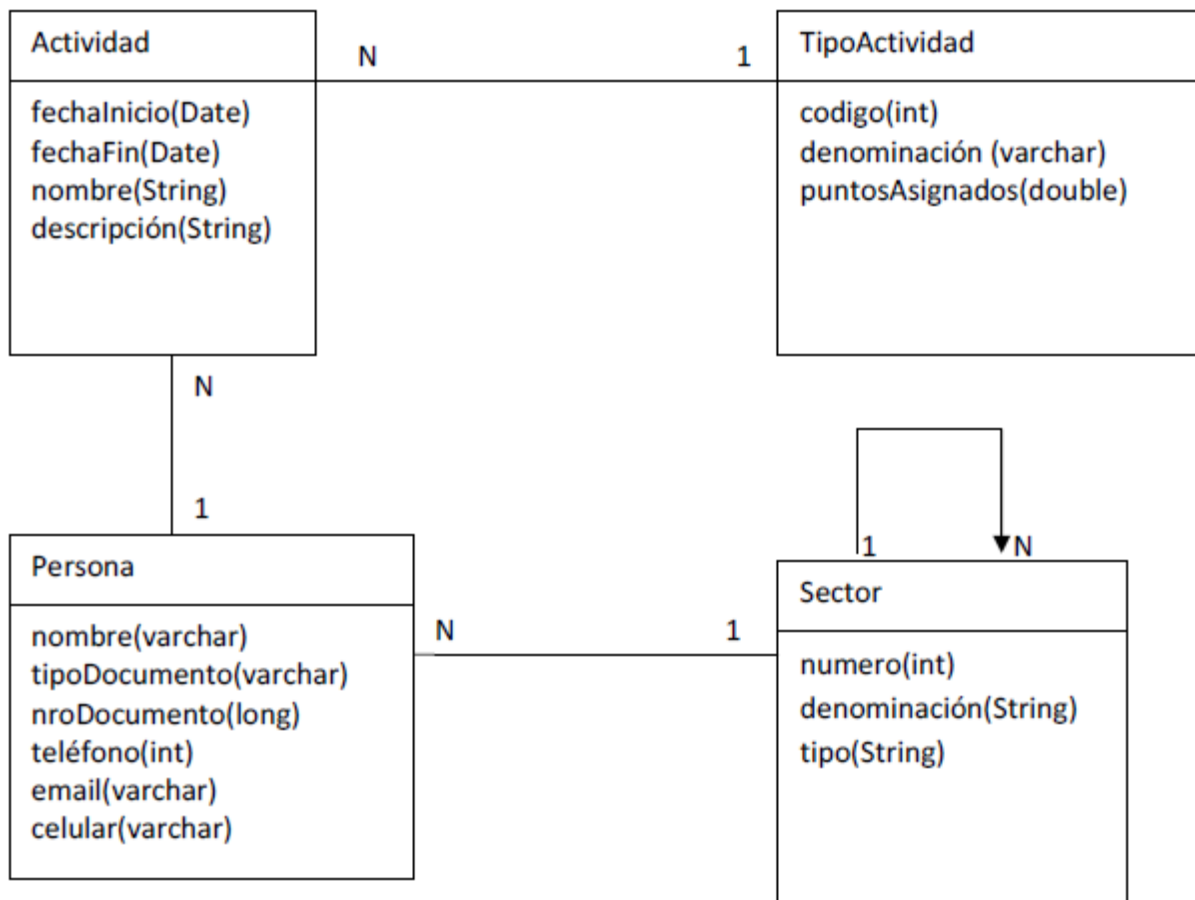


B.1- Represente el modelo de clases de entidad anterior mediante código **JAVA**, todas las asociaciones deben ser bidireccionales.

B.2- Codifique en la clase **public List<Asistencia> getAsistenciaXMesXAnio(int mes, int anio){}** que retorne solo las asistencias del empleado correspondientes al mes y año indicados

B.3- Codifique en la clase **public List<Tardanza> getDiasConTardanza(int mes, int anio){}** retorne los días con tardanza para hacerlo verifique el horario de la asistencia contra el horario asignado en el régimen horario, si la asistencia supera en 15 minutos el horario establecido en el régimen instancia de Tardanza y copie la información de la Asistencia a la Tardanza, almacene la tardanza en un Array del mismo tipo, finalmente retorne el Array de tardanzas. Reutilice el método punto anterior.

C)-



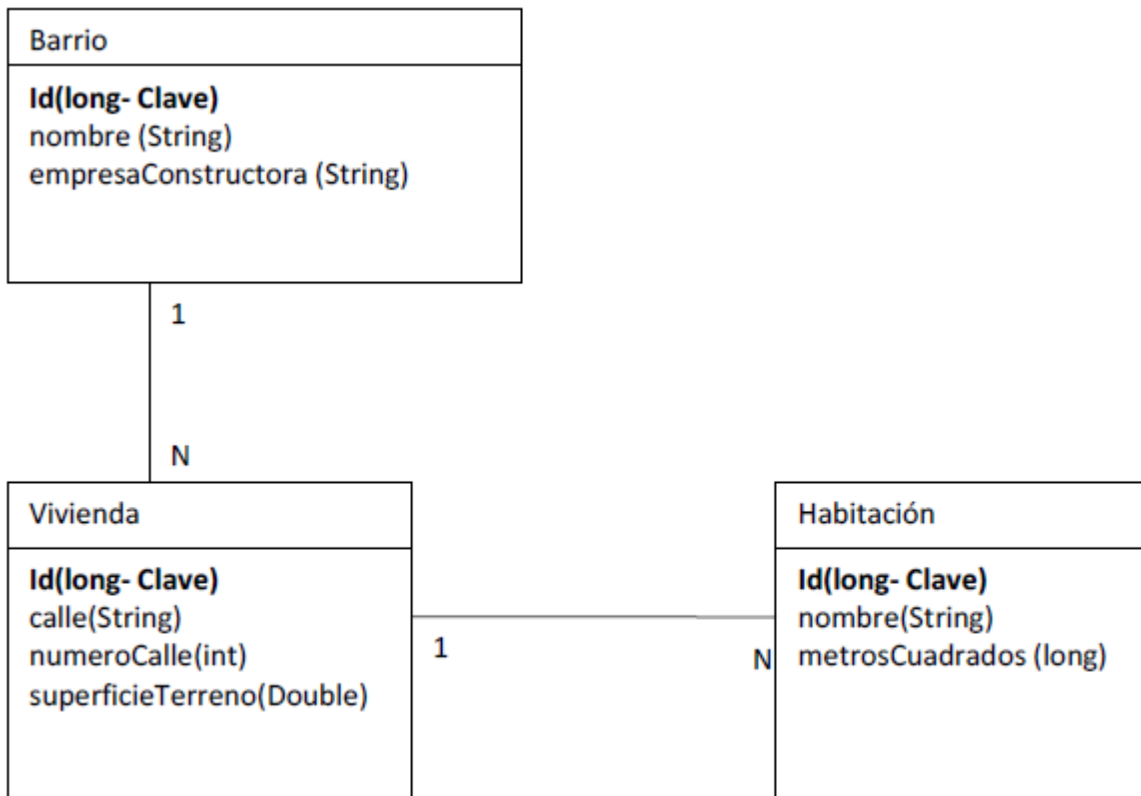
C.1- Codifique en la clase Persona un método denominado “**public double totalPuntosAsignados()**” que retorne el total de puntos asignados de las actividades realizadas por una persona.

C.2- Codifique en la clase Persona un método denominado “**public double totalPuntosAsignados (int código)**” que retorne el total de puntos asignados de un único tipo de actividad realizada por una persona.

C.3- Codifique en la clase Persona un método denominado “**public double totalPuntosAsignados (int código, int anio)**” que retorne el total de puntos asignados de un único tipo de actividad realizada por una persona para un periodo asignado.

C.4- Codifique en la clase Sector un método RECURSIVO denominado “**public List<Sector> obtenerTotalSubsectores()**” que retorne la totalidad de subsectores de un sector.

D)-



D.1- Codifique en la clase **Barrio** un método llamado **public double getSuperficieTotalTerreno(){}** que retorne el total de metros de terreno del barrio teniendo en cuenta la totalidad de viviendas asociadas al mismo.

D.2- Codifique en la clase **Vivienda** un método denominado **public double getMetrosCuadradosCubiertos(){}** que retorne el total de metros cuadrados de la vivienda teniendo en cuenta la cantidad de habitaciones asociadas. Al finalizar el cálculo valide que el valor obtenido no sea mayor que la superficie del terreno, si ocurre esa situación emita una excepción con el mensaje “La superficie cubierta no puede ser mayor a la superficie del terreno”

D.3- Codifique en la clase **Barrio** un método llamado **public double getSuperficieTotalCubierta(){}** que retorne los metros cuadrados cubiertos del barrio sumando la totalidad de metros cuadrados cubiertos de las viviendas que componen al barrio. Reutilice el método **getMetrosCuadradosCubiertos ()** del punto anterior.