UD3-Programación

El reto consiste en que practiques y aprendas a realizar un diagrama de clases **UML**en el que intervienen **más de dos entidade**s de forma que hay algún tipo de relación entre ellas.

Utiliza **algún software de modelado**, por ejemplo, los citados en la unidad, o cualquier otro que sepas o quieras aprender a manejar, para crear un **sistema sencillo**de registro y gestión de empleados utilizando los principios de la programación orientada a objetos en Java y notación **UML**para el diseño de clases.

El sistema permitirá a los usuarios registrar nuevos empleados, actualizar información existente, consultar datos de empleados y generar informes principalmente.

 Se deben definir las **clases, atributos y métodos que consideres** relevantes, así como las **relaciones**de asociación, agregación, herencia y visibilidad de cada elemento del sistema.

**NOTA**: **NO SE PIDE CÓDIGO JAVA,** simplemente ahora mismo estamos **diseñando**, más abajo tienes **ejemplos de lo que se pide** en este reto.

Las **funcionalidades**del sistema son las siguientes: (Proporciona el método apropiado en la clase correspondiente (**solo se pide la cabecera de cada método**, fíjate de los **ejemplos**que hay más abajo)

* Registro de **empleados**con sus datos personales y laborales.
* Actualización de información de **empleados**(datos personales y laborales).
* Consulta de datos de **empleado**(todos).
* Los empleados tendrán una **categoría**profesional.
* Consulta de **categorías**profesionales existentes.
* Registros de **departamentos**a los que puede pertenecer el empleado.

**El diagrama consta de tres clases:**

- Empleado.

- Categoría Profesional.

- Departamento.

**Relaciones**

Asociación (Entre Empleado y Categoría Profesional):

- La clase “Empleado” tiene un atributo “categoría Profesional” que representa la asociación con la categoría profesional.

Agregación (Entre Departamento y Empleado):

- La clase “Departamento” tiene una lista de empleados, lo que indica una relación de agregación.

**Visibilidad**

- Los atributos de las clases son privados para encapsular la implementación interna.

- Los métodos son públicos para que puedan ser utilizados desde otras clases.