



CICLO: [DAM]
**MÓDULO DE [ENTORNOS DE
DESARROLLO]**

[Tarea N° 03]

Alumno:
[Juan Carlos Filter Martín]
[REDACTED]

Contenido

1. Documentos que se adjuntan a este informe.....	3
2. Caso práctico.....	3
3. (RA5_c) Se ha interpretado el significado de diagramas de clases.....	4
4. (RA5_d) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.....	6
5. (RA5_e) Se ha generado código a partir de un diagrama de clases.....	9

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Captura diagrama UML
3. Zip con el código java generado y el archivo .mdj

2. Caso práctico.

Se desea desarrollar una aplicación que permita llevar a cabo la gestión de la actividad de **distintas empresas, almacenando información acerca de sus clientes, empleados y directivos.**

La aplicación deberá cubrir los aspectos funcionales que se describen a continuación:

Clases

Atributos

Relaciones

- La aplicación deberá almacenar información acerca de **empresas**, **empleados** de dichas empresas y **clientes**.
- Tanto de los **empleados** como de los **clientes**, se desea almacenar su **DNI, nombre, dirección, teléfono y fecha de nacimiento**.
- De los **empleados**, interesa conocer su **sueldo mensual y la fecha de contratación**. De entre todos los **empleados**, hay un tipo especial que es el empleado "**Directivo**" que es el jefe de un conjunto de **empleados**, y del cual interesa conocer su **categoría profesional**.
- De los **clientes** es de interés conocer un **teléfono de contacto** y su **ocupación laboral**.
- De cada **empresa** se desea conocer su **nombre identificativo, año de fundación y la dirección de su sede**. La aplicación necesita controlar que **empleados** y que **clientes**, están ligados a cada empresa concreta
- De un **proveedor** interesa conocer su **código identificativo, su nombre, su dirección, y el teléfono de contacto**.
- De un **producto** se desea conocer el **código del producto, su nombre y tipo**.
- **Un determinado proveedor puede suministrar muchos productos a un conjunto empresa y un determinado producto, puede ser suministrado por varios proveedores.**

3. (RA5_c) Se ha interpretado el significado de diagramas de clases.

Tanto de los empleados como de los clientes, se desea almacenar su DNI, nombre, dirección, teléfono y fecha de nacimiento.

→ Las clases serían : Cliente (hijo) y empleado (hijo) junto a otra clase más llamada persona (padre) para que sea heredado a las clases cliente y empleado con sus atributos en común

→ dni, nombre, dirección teléfono y fecha de nacimiento irán en la clase persona (padre)

De los empleados, interesa conocer su sueldo mensual y la fecha de contratación. De entre todos los empleados, hay un tipo especial que es el empleado "Directivo" **que es el jefe de un conjunto de empleados**, y del cual interesa conocer su categoría profesional.

→ Se crea otra clase Directivo heredado de la clase empleado sueldo mensual y fecha de contratación y esta a su vez hereda de la clase persona con los atributos en común de cliente y empleado.

→ Sueldo mensual y fecha de contratación se le añade a empleado y a directivo le ponemos el atributo categoría profesional.

→ Para indicar que directivo es jefe de 1 o muchos empleados lo indicamos mediante otra relación teniendo 2 relaciones (1 herencia y otra de asociación con la multiplicidad 1 directivo es jefe de 1..* Empleados)



De los clientes es de interés conocer un teléfono de contacto y su ocupación laboral.

→ Le añadimos el atributo ocupación laboral y el teléfono de contacto lo hereda de la clase persona (padre)

De cada empresa se desea conocer su nombre identificativo, año de fundación y la dirección de su sede.

→ Se crea una clase llamada Empresa (padre). Desde esta clase va a ir asociada las clases empleado y cliente.

→ nombre identificativo, año de fundación y dirección sede serán los atributos de la Empresa(padre)

La aplicación necesita controlar que empleados y que clientes, están ligados a cada empresa concreta

→ Cliente: tipo de multiplicidad será: 1 proveedor → * empresas

* Empresas puede tener * clientes

* Clientes puede ir a * Empresas



→ Empleado: tipo de multiplicidad será:

1 Empresa puede tener 1..* Empleados

1..* Empleados puede estar en 1 Empresa



De un proveedor interesa conocer su código identificativo, su nombre, su dirección, y el teléfono de contacto.

→ Se crea una clase llamada Proveedor siendo un tipo de relación Asociación a la clase Empresa

→ Código identificativo, dirección, teléfono serán los atributos de esta clase

De un producto se desea conocer el código del producto, su nombre y tipo.

→ Se crea una clase llamada Producto siendo un tipo de relación Agregación a la clase Proveedor

→ Código producto, nombre, tipo serán los atributos de esta clase

Un determinado proveedor puede suministrar muchos productos a un conjunto empresa y un determinado producto, puede ser suministrado por varios proveedores.

→ Proveedor: tipo de multiplicidad será: 1 proveedor → * empresas



→ Producto: tipo de multiplicidad será: 1 producto → * Proveedor



Empresa: #Nombre_identificativo, #año_fundación, #dirección sede

Persona: #dni, #nombre, #dirección, #teléfono, #fecha nacimiento

Empleado: #Sueldo mensual, #fecha contratación

Cliente: -Ocupación Laboral

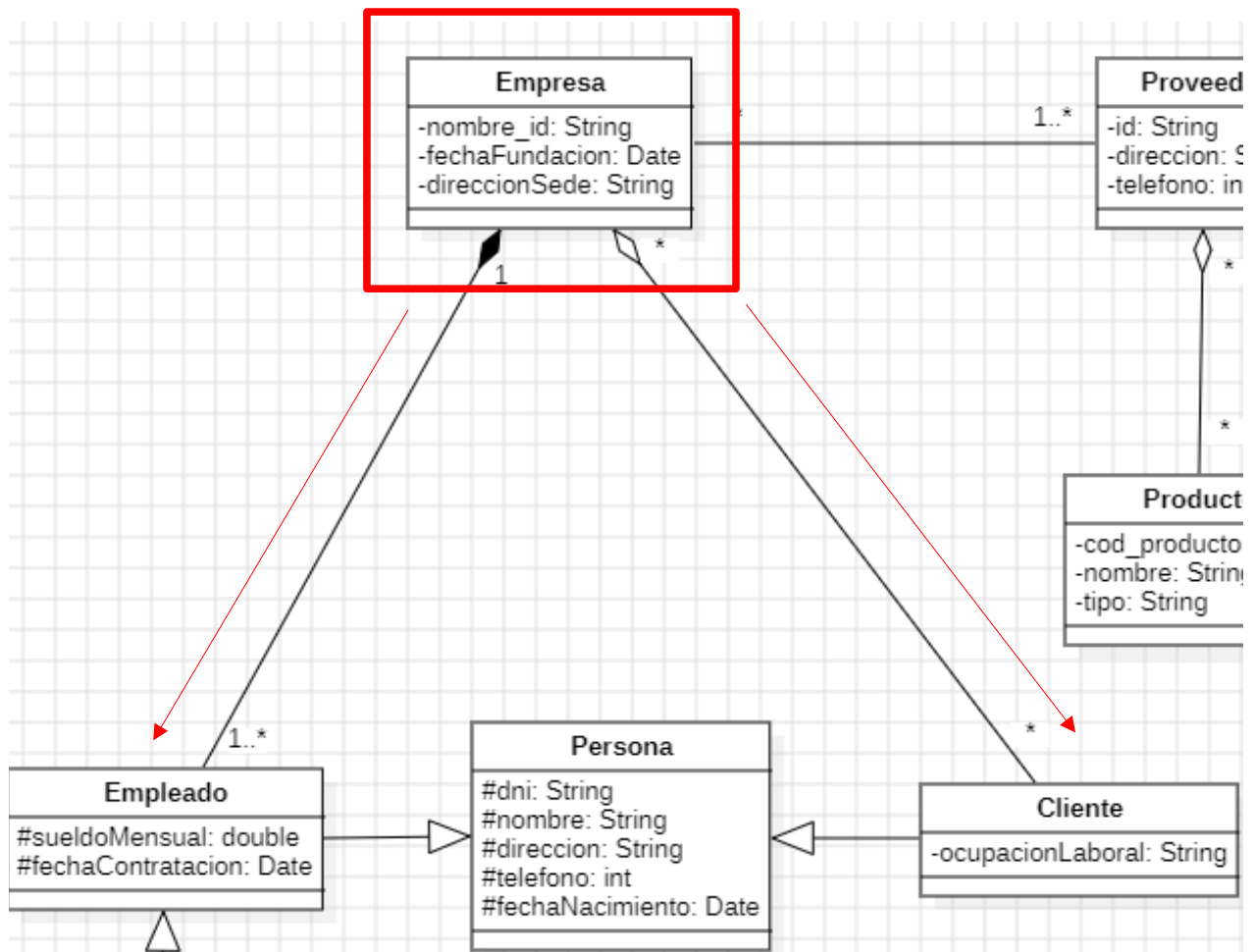
Directivo: -Categoría profesional

Proveedor: -Código identificativo, -dirección, - teléfono

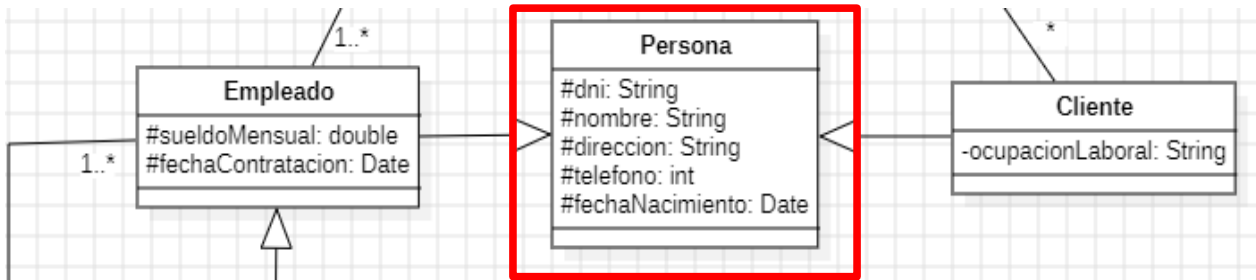
Producto: - Código producto, - nombre, -tipo

4. (RA5_d) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.

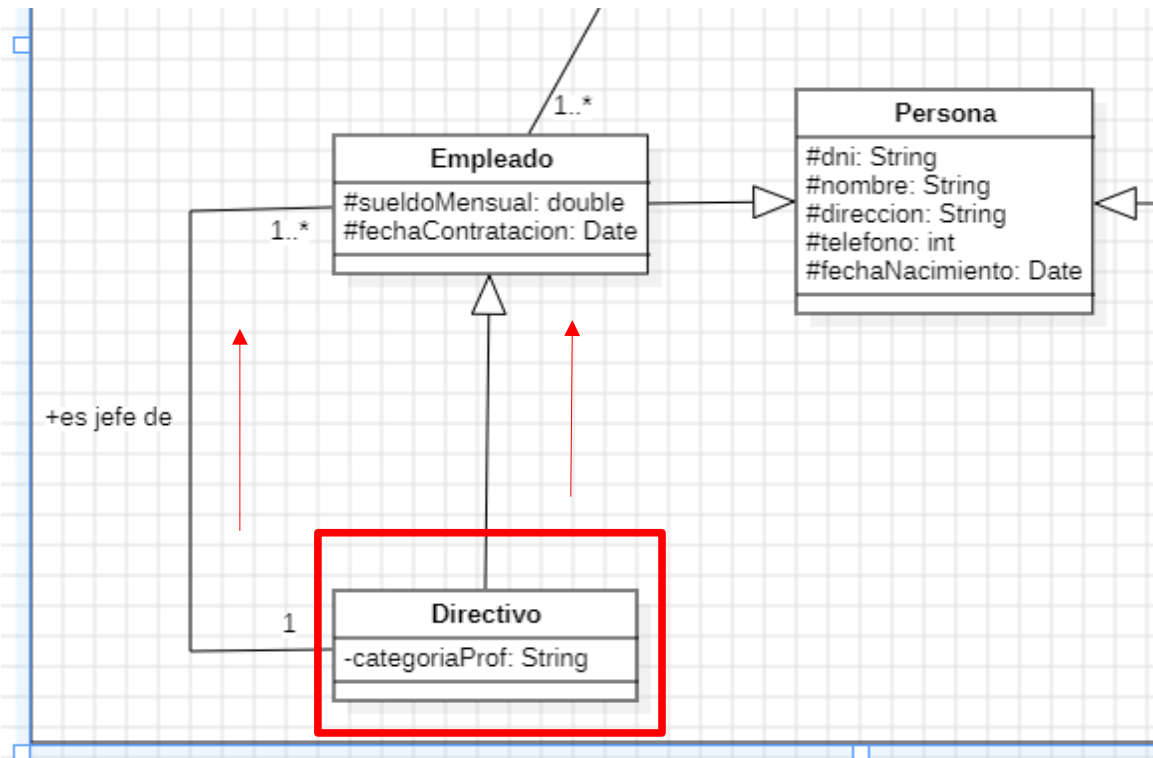
- ✖ De la clase Empresa mediante una relación asociación va a ir la clase empleado y cliente
- ✖ En la clase Empresa los atributos serán privados ya que no se quiere dar acceso a esos datos en otra clase, siendo empleado protected y en cliente privado porque cliente no hereda ninguna otra clase.
- ✖ Siendo de agregación para cliente porque si una empresa deja de existir el cliente no y composición para la relación de empleado a empresa ya que si una empresa no existe el empleado tampoco.



- × También se crea una clase global que se va a llamar persona en ella va a contener los atributos de empleado y cliente que tienen en común mediante herencias



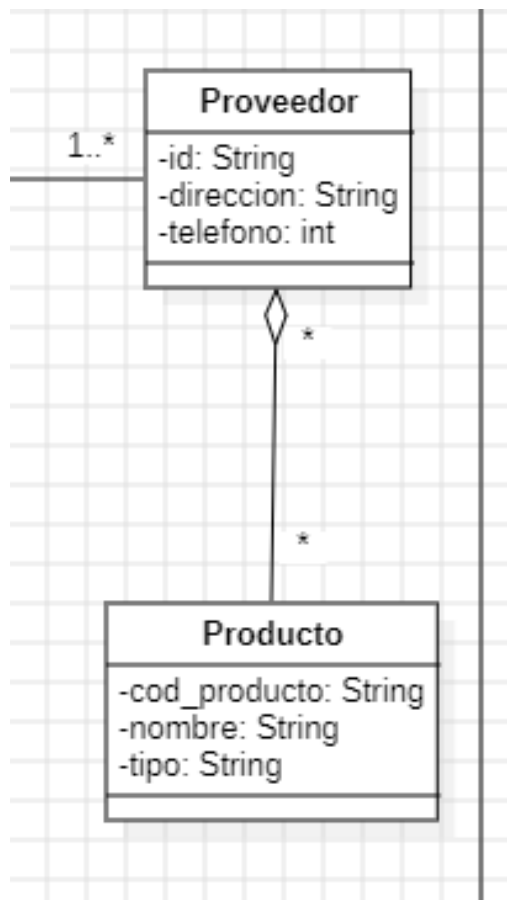
- × Se va a crear una clase llamada directivo que va a contener un atributo llamado categoría profesional y a su vez esta clase va a heredar de empleado.
- × Para indicar que un directivo es jefe de 1 o muchos empleados se indica mediante otra relación con su multiplicidad.



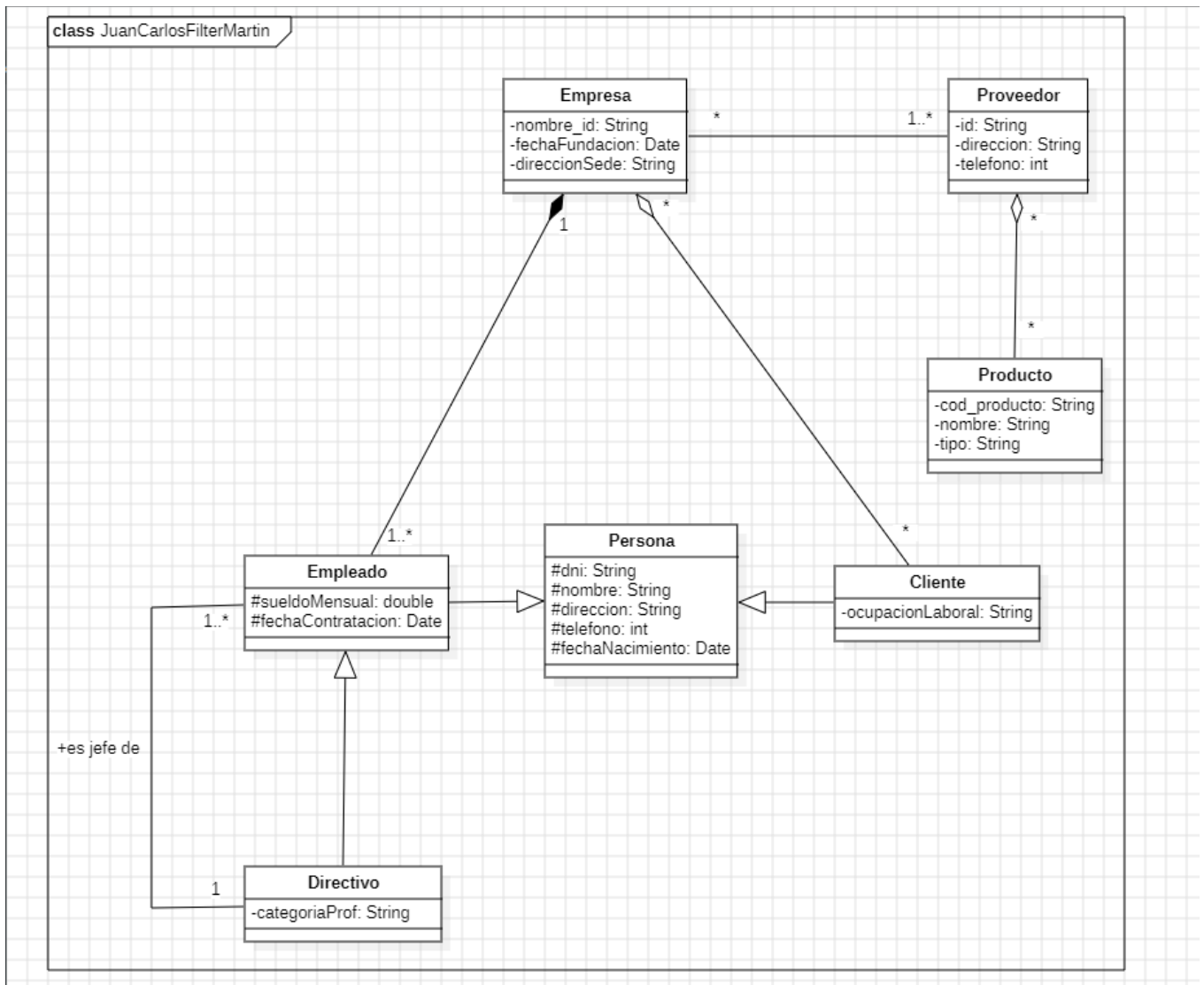
- × Se crea una clase llamada proveedor con una de relación a empresa y le indicamos la siguiente multiplicidad:



- * Por ultimo creamos una clase llamada producto con una relación de agregación a proveedor y le indicamos la siguiente multiplicidad:



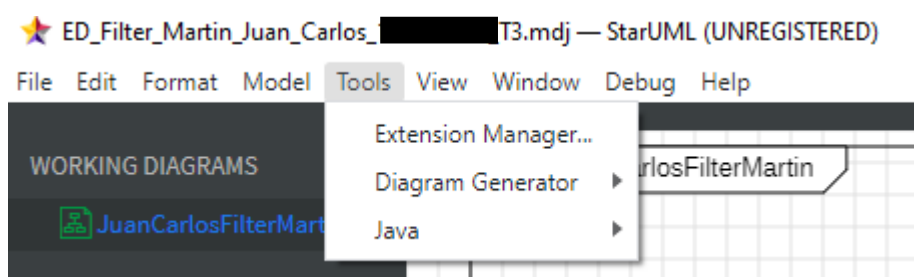
- ✖ El resultado sería el siguiente:



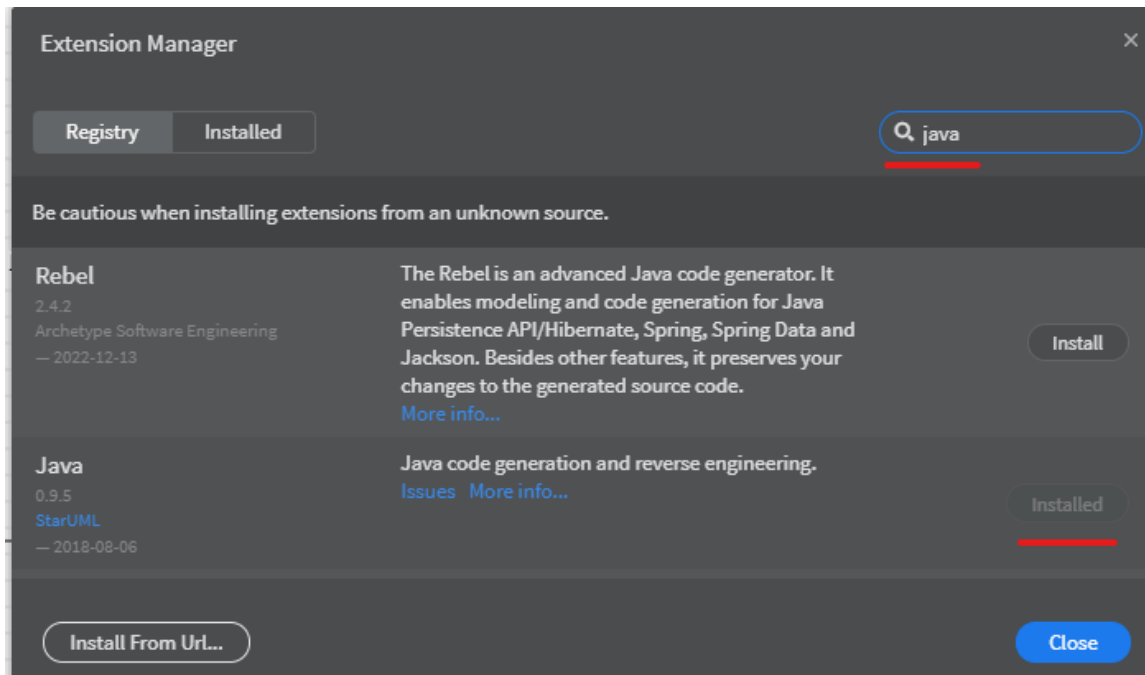
5. (RA5_e) Se ha generado código a partir de un diagrama de clases.

Para generar el código de java desde StarUML lo primero sería instalar la extensión java para ello haremos los siguientes pasos:

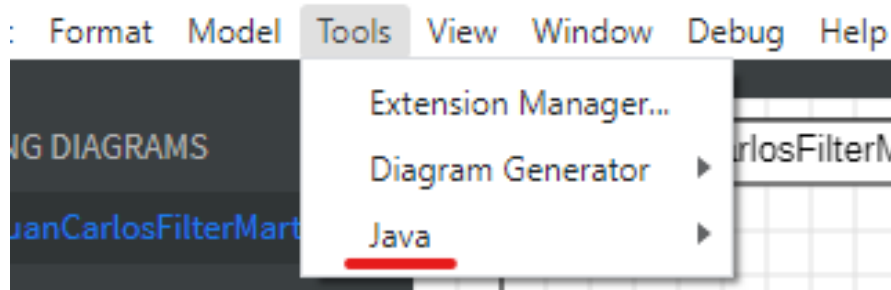
- ✖ **Tools > Extension Manager...**



x Buscamos java y procedemos a instalarlo

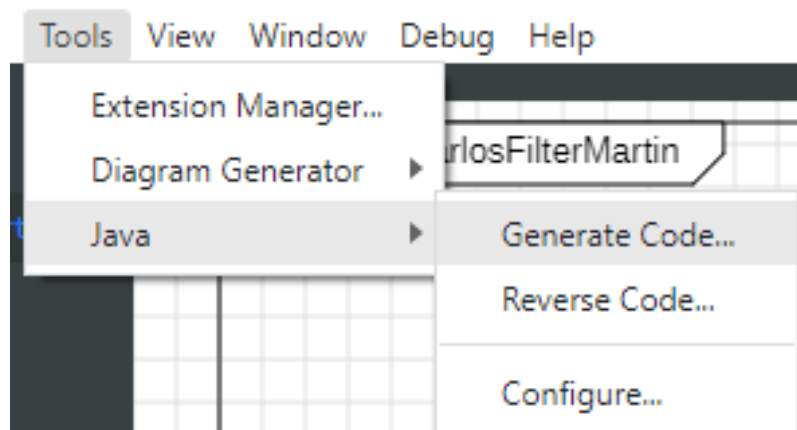


Una vez instalado ahora nos debe de aparecer Java en la pestaña Tools

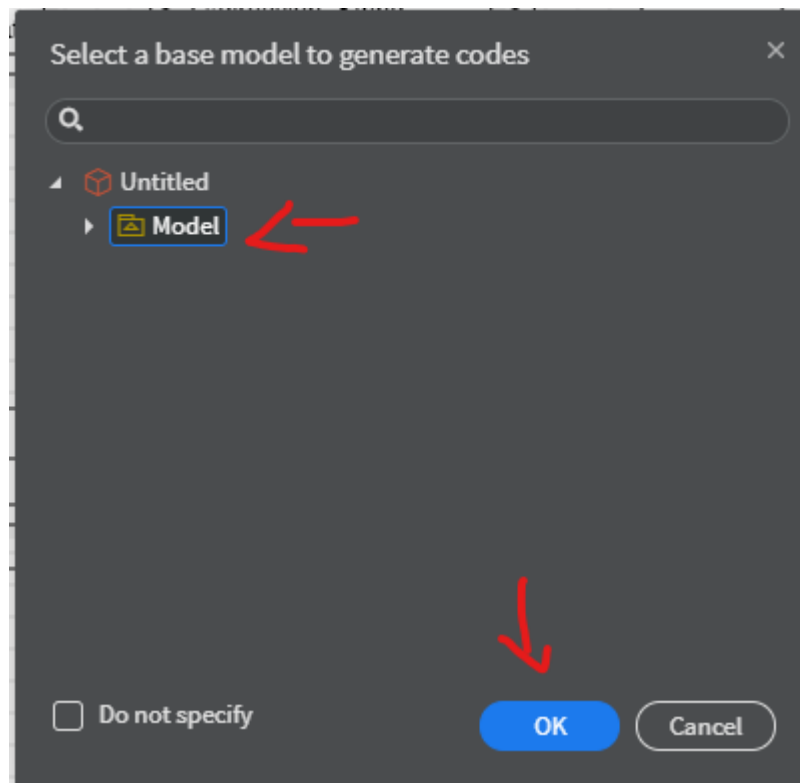


Para generar el código java pulsamos en:

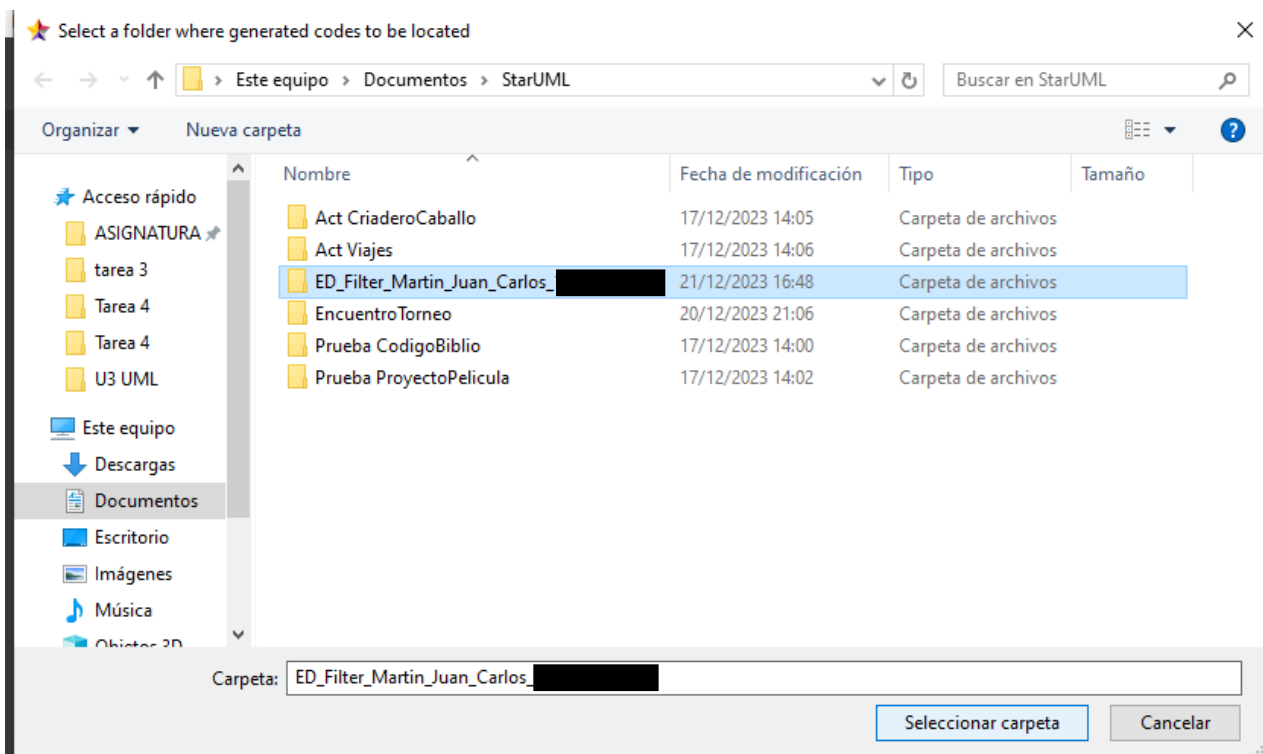
x **Tools > Java > Generate Code..**



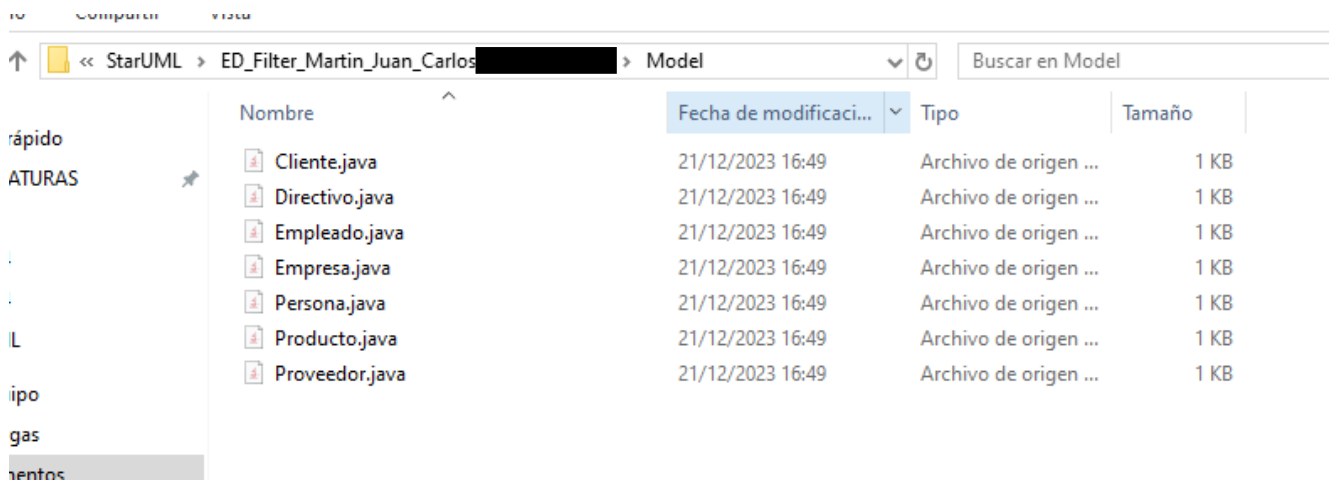
Nos saldrá la siguiente ventana donde elegimos el model y pulsamos en ok



Ahora nos dirá donde queremos guardar el código que se va a generar, le indicamos la carpeta/directorio

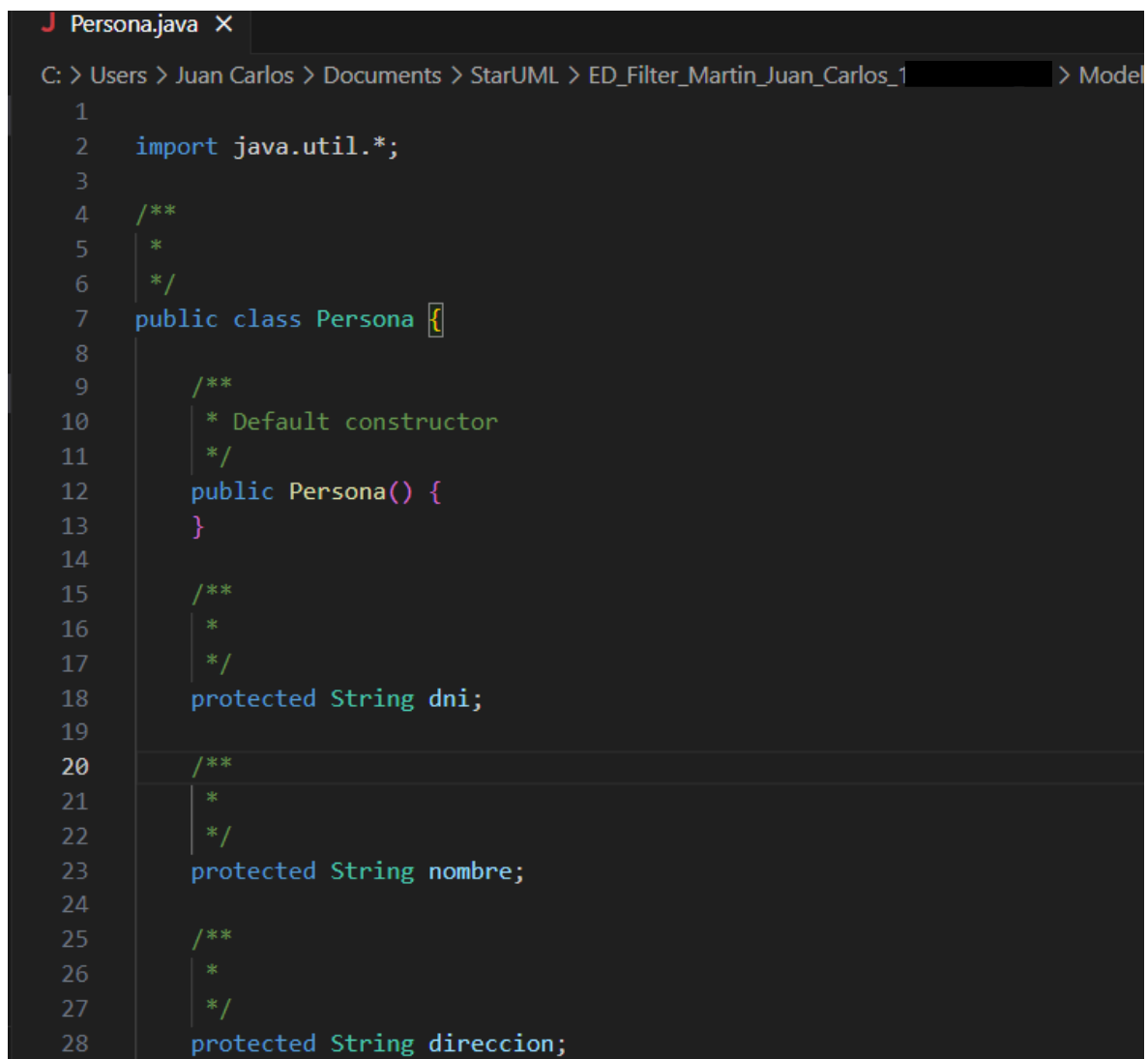


Ya tendríamos el código generado si ahora vamos a la carpeta y entramos en Model podemos ver todas las clases con su código generado



Nombre	Fecha de modificaci...	Tipo	Tamaño
Cliente.java	21/12/2023 16:49	Archivo de origen ...	1 KB
Directivo.java	21/12/2023 16:49	Archivo de origen ...	1 KB
Empleado.java	21/12/2023 16:49	Archivo de origen ...	1 KB
Empresa.java	21/12/2023 16:49	Archivo de origen ...	1 KB
Persona.java	21/12/2023 16:49	Archivo de origen ...	1 KB
Producto.java	21/12/2023 16:49	Archivo de origen ...	1 KB
Proveedor.java	21/12/2023 16:49	Archivo de origen ...	1 KB

y esto es un ejemplo de la clase persona:



```
1
2 import java.util.*;
3
4 /**
5  *
6  */
7 public class Persona {
8
9     /**
10      * Default constructor
11      */
12     public Persona() {
13     }
14
15     /**
16      *
17      */
18     protected String dni;
19
20     /**
21      *
22      */
23     protected String nombre;
24
25     /**
26      *
27      */
28     protected String direccion;
```