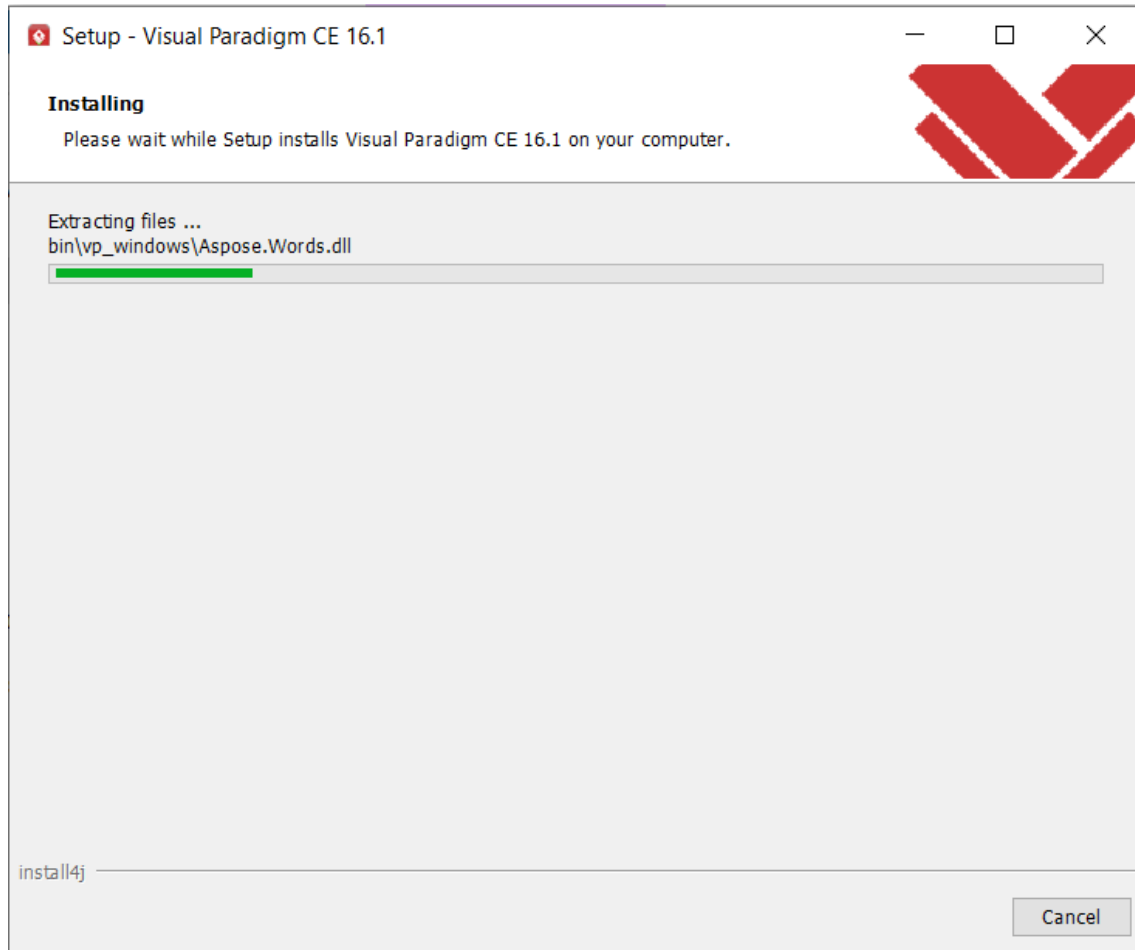


Práctica5c Eclipse-VisualParadigm.

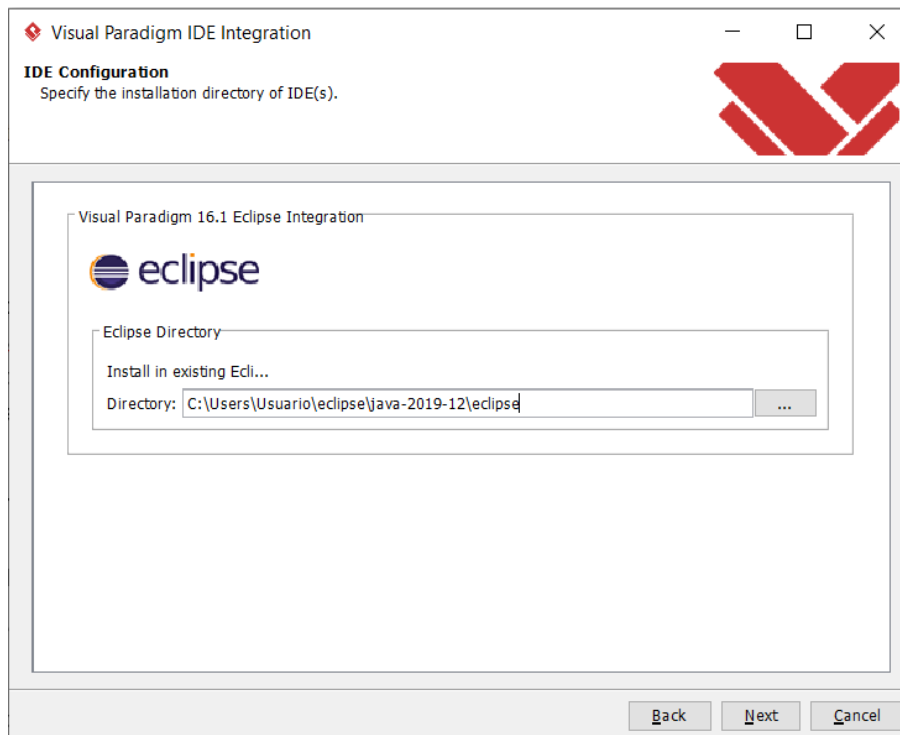
1.- Instalación de Visual Paradigm CE 16.1. Primero instalamos la Aplicación Visual Paradigm en nuestra máquina y posteriormente, la integramos en Eclipse, como Administrador del equipo.

1.1 Instala Visual Paradigm CE 16.1 como administrador usando el ejecutable que te proporciona el profesor.



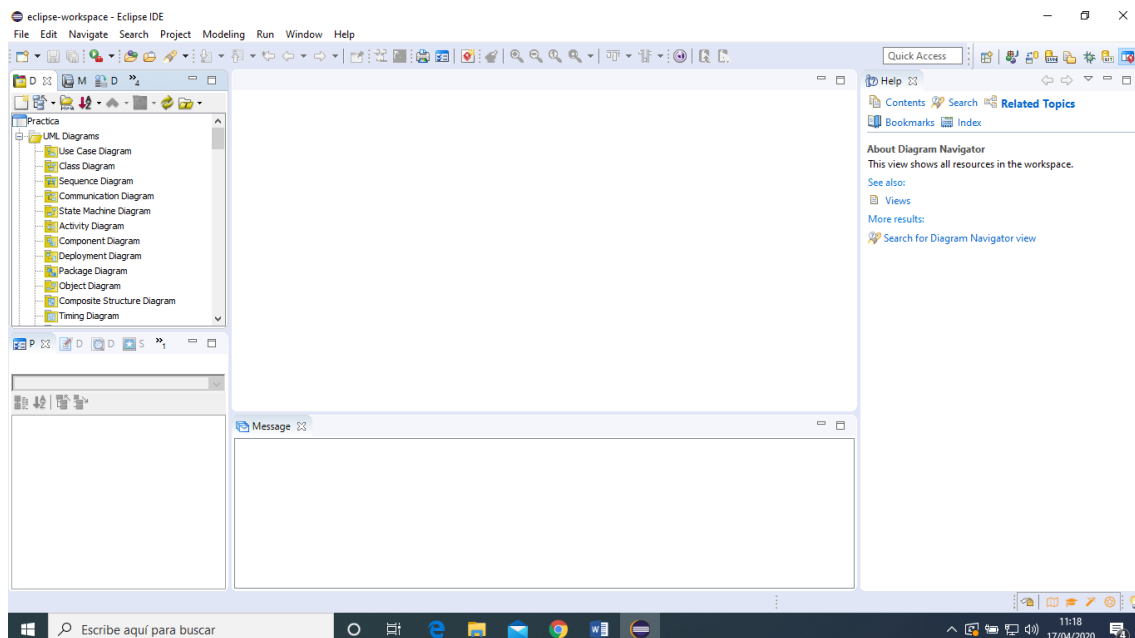
1.2 Integra Visual Paradigm en Eclipse. En Visual Paradigm, Windows> Integration>IDE Integration>Eclipse Integration. Indicamos la ruta de la carpeta donde se encuentra el ejecutable “eclipse.exe”, en mi PC es: “C:\Users\otro\eclipse\java-2019-09\eclipse”. Se instalarán en esta carpeta el plugin, para que podamos usar UML dentro de Eclipse. Cierra Visual Paradigm.

UML no contempla el tipo de dato “Date” de Java. Por ello vamos a asignar al atributo “fecha” en esta práctica el tipo “String”. En Visual Paradigm, se puede añadir el tipo de dato “Date” a UML, pero obliga a trabajar en un determinado proyecto, y desde Eclipse es complicado.

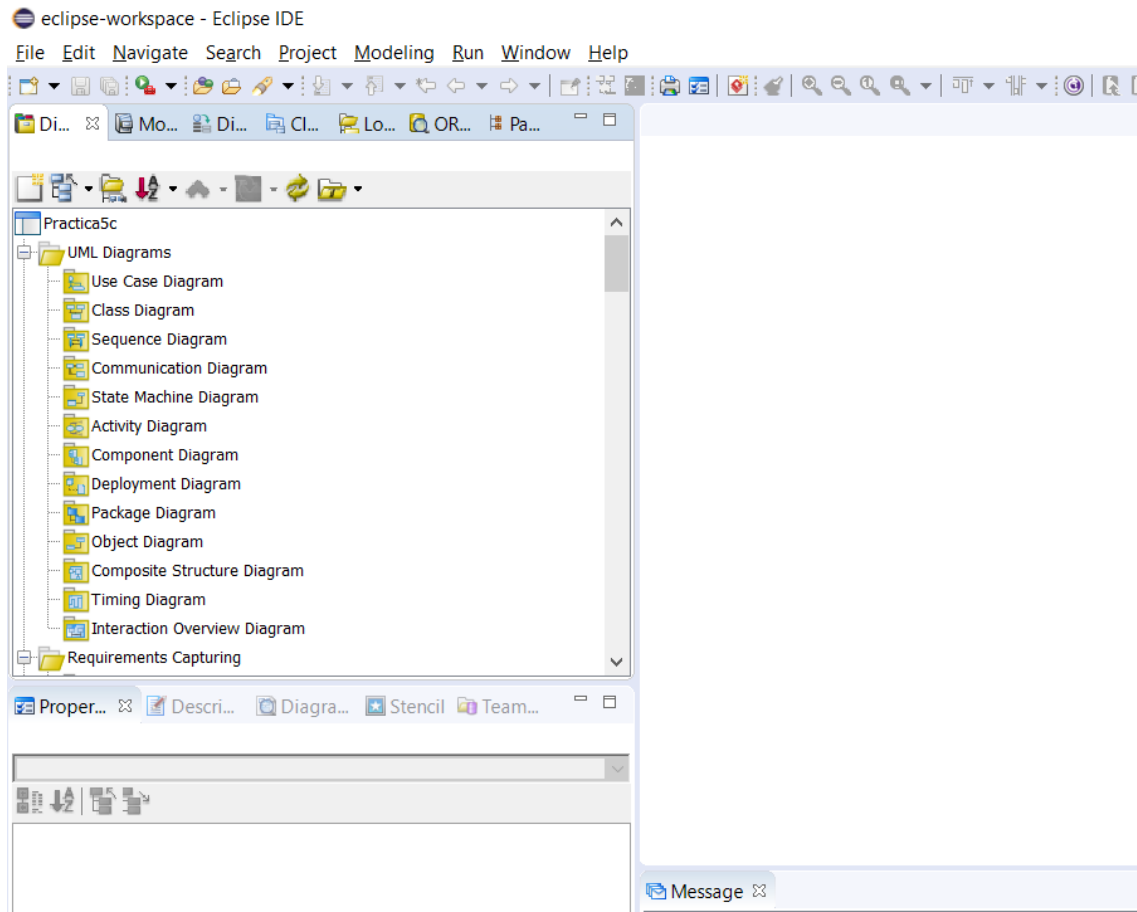


2.- Ejecuta Eclipse, y crea el proyecto Java “Práctica5c” desde la perspectiva Java por defecto.

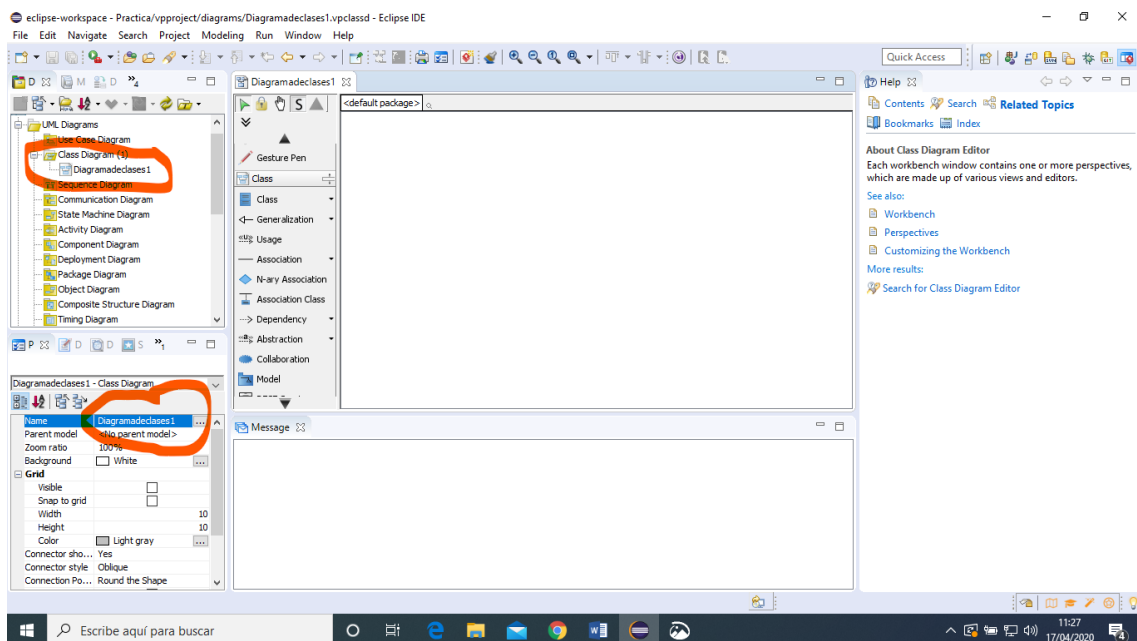
En este proyecto vamos a crear el “Diagramadeclases1”: Botón derecho de “Practica5c”>New>Open Visual Paradigm>Cambia a perspectiva “Modeling. Tendrás entonces la siguiente pantalla:

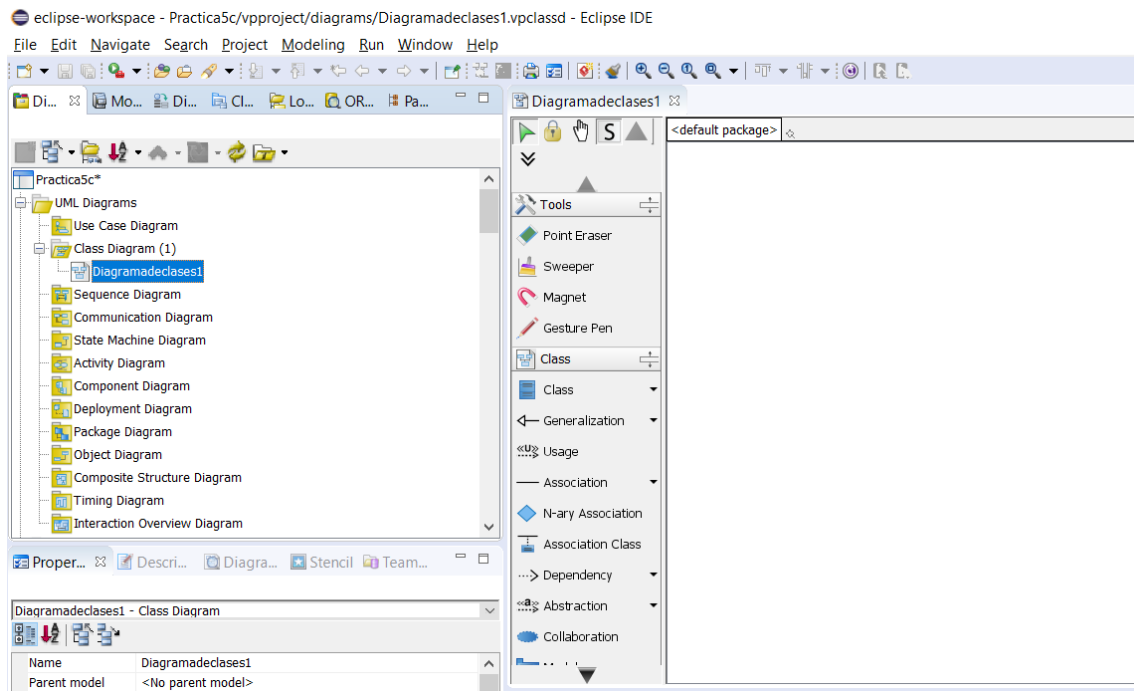


Tienes un tutorial aquí: <https://www.visual-paradigm.com/tutorials/eclipseuml.jsp>



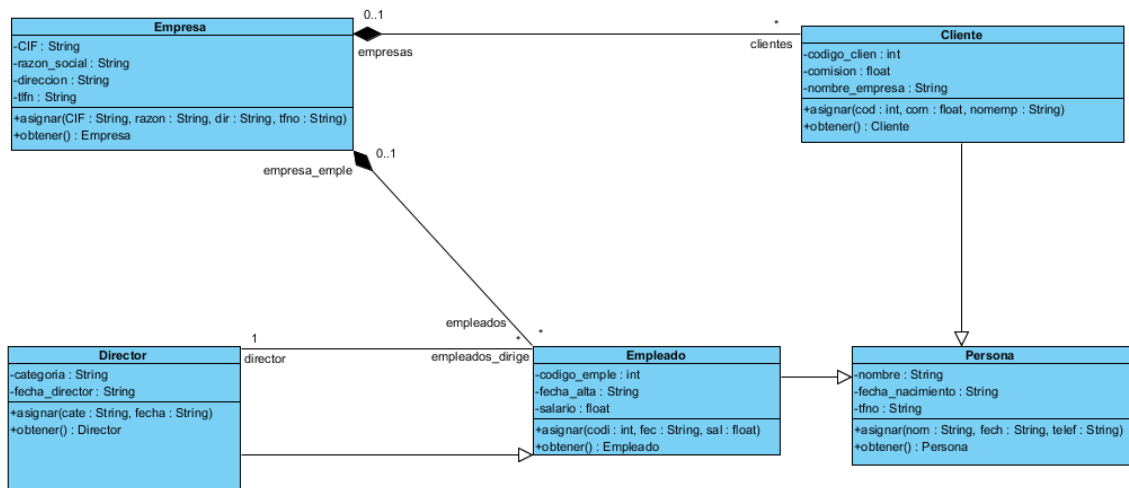
2.1 Selecciona en la pantalla anterior “Class Diagram”>New Class Diagram. Cambia el nombre por defecto del diagrama a “Diagramadeclasses1”. Debes obtener:





2.2 Con la ayuda del tutorial realiza un diagrama de clases para representar las relaciones entre Empresa, Empleado y Cliente, usando Composición y Herencia. Los requisitos son:

- La empresa se compone de clientes y de empleados. Usa 2 asociaciones de composición.
- Datos de empresa: CIF, razón social, dirección y teléfono.
- Datos de cliente: código de cliente, nombre, fecha de nacimiento, teléfono, empresa para la que trabaja y comisión.
- Datos de empleado: código de empleado, nombre, fecha de nacimiento, teléfono, fecha de alta en la empresa y salario.
- Dado que nombre, fecha de nacimiento, y teléfono, son comunes a las clases Empleado y Cliente, crea la clase Persona con estos atributos, y las clases Empleado(subclase)y Cliente (subclase) heredan la clase Persona (superclase).
- Un empleado puede ser director de varios empleados, y del director se necesita saber la categoría y la fecha de alta como director. El director hereda de empleado, y tiene una asociación 1...* con empleado.
- Para todas las clases crearemos 2 métodos, uno para asignar datos a los atributos (como si fuese el constructor), este método tendrá tantos parámetros como atributos, otro que se llame obtener que devolverá un objeto de la misma clase.



Como ayuda, aquí tienes varias capturas de las propiedades de las clases y asociaciones necesarias:

<default package>

Empresa

-CIF : String

-razon_social : String

-direccion : String

-tfn : String

+asignar(CIF : String, razon : String, dir : String, tfno : String)

+obtener() : Empresa

empresa_emple

Class Specification

General

Attributes

Operations

Receptions

Name	Classifier	Visibility	Type	Initial Value
CIF	Empresa	private	String	
razon_social	Empresa	private	String	
direccion	Empresa	private	String	
tfn	Empresa	private	String	

Operation Specification

General

Parameters

Name	Type	Default Value	Direction
cod	int		return
com	float		return
nomemp	String		return

×

Cliente

-codigo_clien : int

-comision : float

-nombre_empresa : String

+asignar(cod : int, com : float, nomemp : String)

+obtener() : Cliente



Association Specification

General

Name:

Visibility: <Unspecified>

Association End From

Role: ...

Element: ...

Multiplicity: v

Navigable: v

Association End To

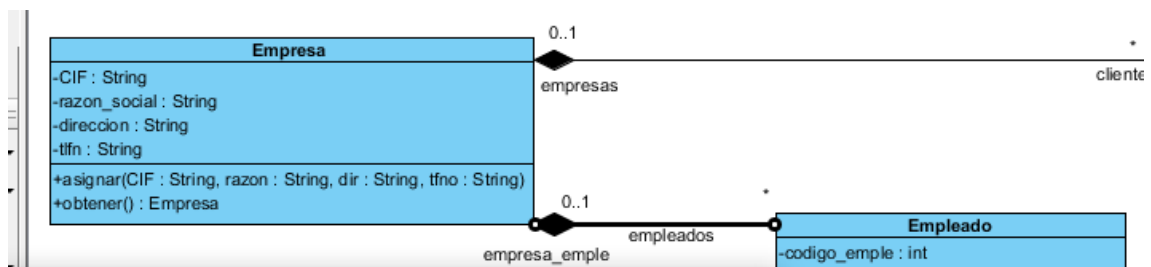
Role: ...

Element: ...

Multiplicity: v

Navigable: v

Description:



Association Specification

General

Name:

Visibility: <Unspecified>

Association End From

Role: ...

Element: ...

Multiplicity: v

Navigable: v

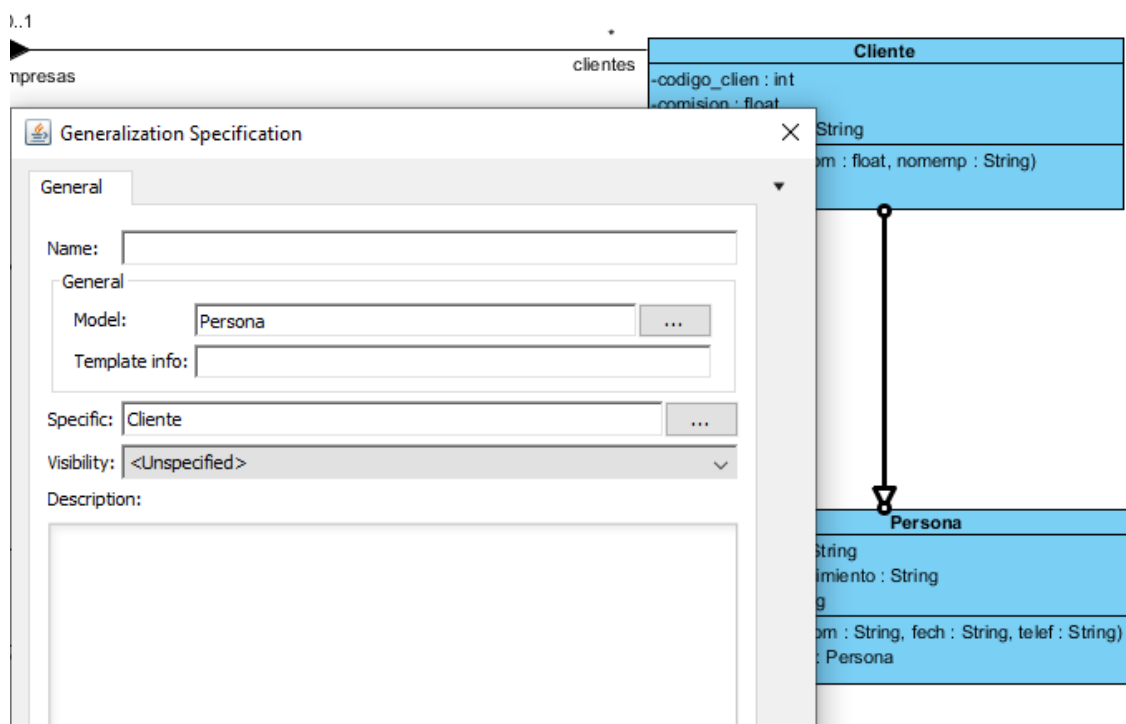
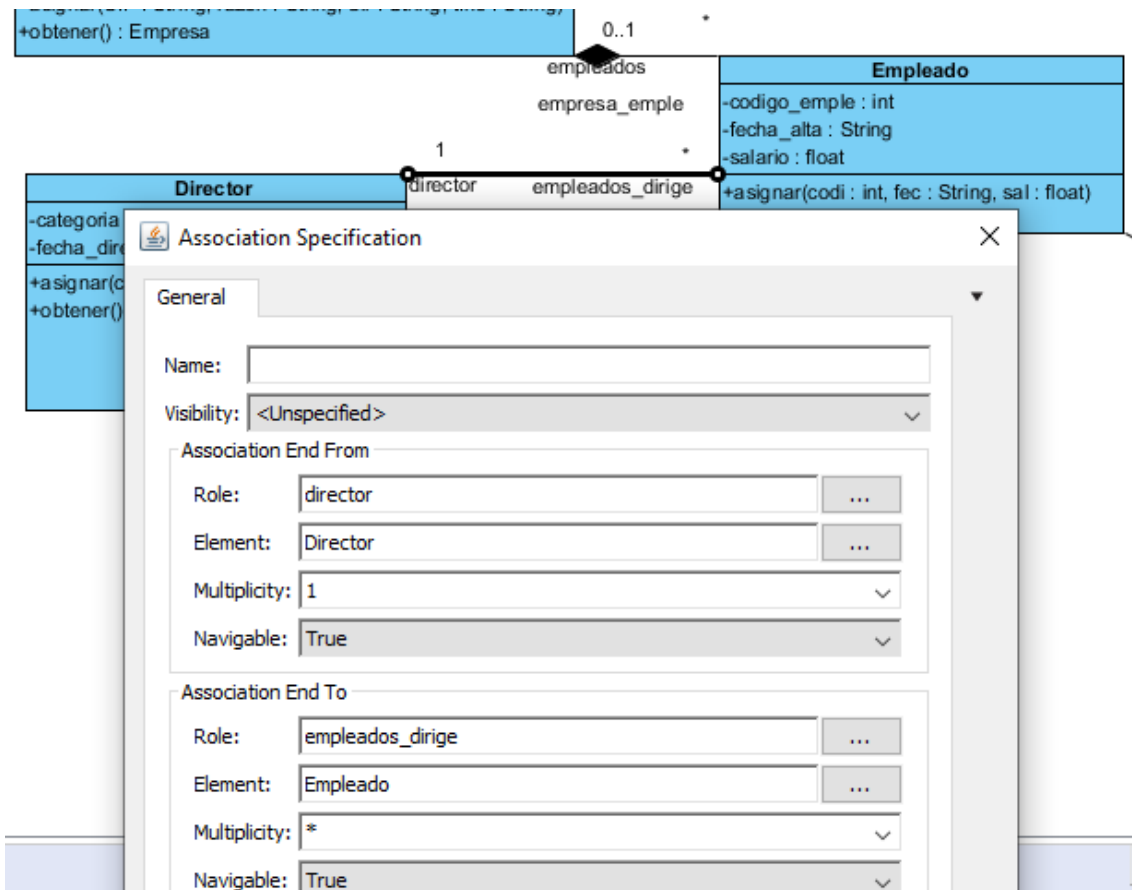
Association End To

Role: ...

Element: ...

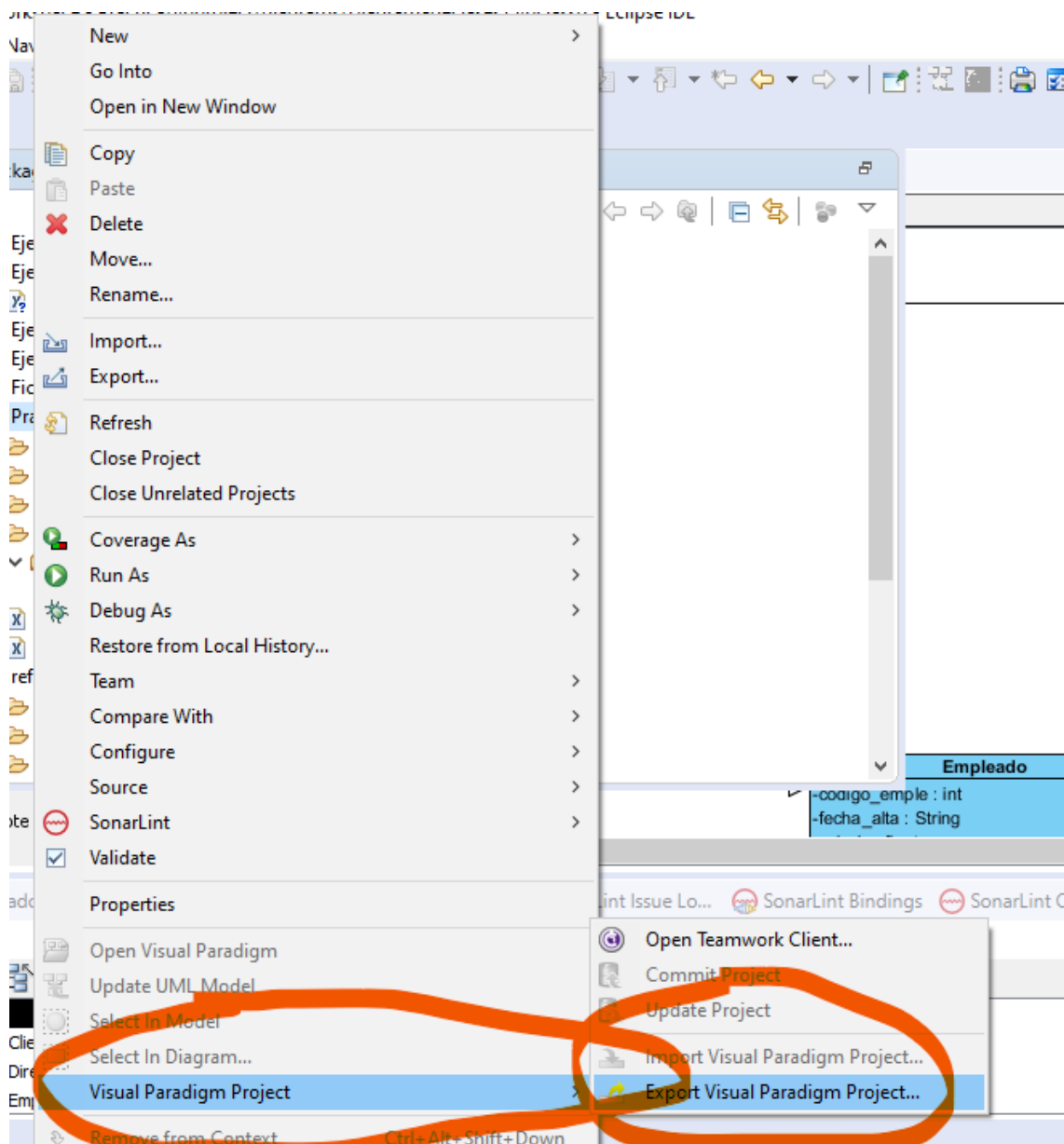
Multiplicity: v

Navigable: v

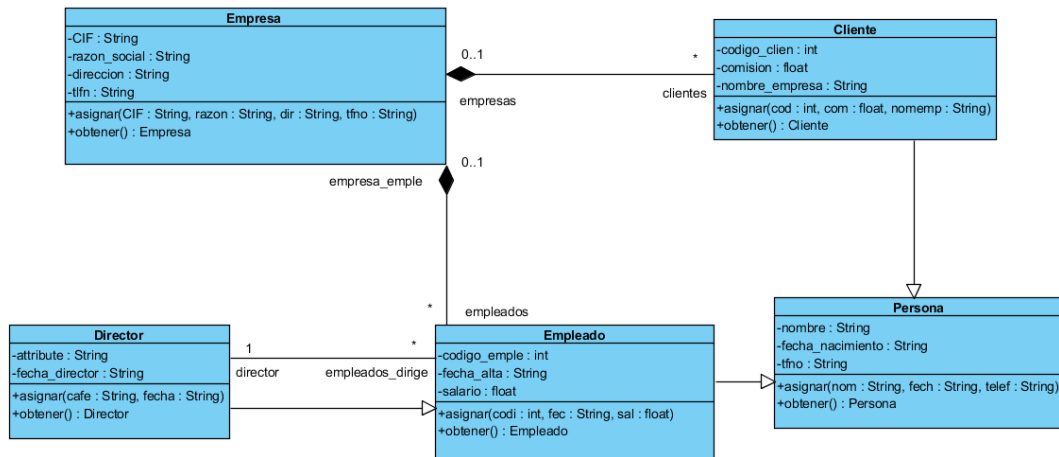


3 Salva el diagrama de clases realizado y exporta el proyecto "Práctica5c".

3.1 Desde el botón derecho de “Práctica5c”>Visual Paradigm Project>Export VP Project>



3.2 Guardar en una carpeta el archivo “Practica5ctuNombrey Apellidos.vpp”. Subir dicho archivo a la Tarea Practicas de Moodle.



4. Para no utilizar Visual Paradigm en Eclipse, debes abrir primero Visual Paradigm, y deshacer el punto 1.2. Esto lo haremos cuando finalicemos la UD5.