Tarea 5: Diseño Orientado a Objetos. Elaboración de Diagramas

Francisco Javier Sueza Rodríguez

17 de mayo de 2023

Centro:	IES Aguadulce
Ciclo Formativo:	Desarrollo Aplicaciones Web (Distancia)
Asignatura:	Entornos de Desarrollo
Tema:	Tema 5 - Diseño Orientado a Objetos. Elaboración de Diagramas

Índice

Índice de figuras

1. Caso Práctico

Juan va a realizar una aplicación para un cliente. Ada le indica que antes de empezar debe obtener diversos datos para realizar un buen análisis, ya que es una fase fundamental en el desarrollo de software. María, aprovecha para explicarle las ventajas de realizar un buen diagrama de clases y demás diagramas. Juan entiende que tiene mucho sentido realizarlo y se pone manos a la obra.

2. Parte 1: Diagramas de Clases

2.1. Ejercicio 1

2.1.1. Enunciado

Elabora el siguiente diagrama de clases mediante el programa Visual Paradigm. Debes crear el proyecto VP-UML.

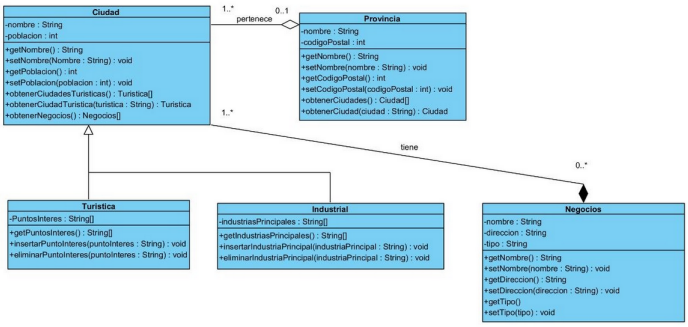

Figura 2.1: Diagrama de Clases

2.1.2. Solución

En este primer ejercicio, hemos creado un proyecto en Visual Paradigm. El proyecto se incluye dentro de los archivos entregados, pero a continuación se muestra la diagrama creado en Visual Paradigm. Se han redistribuido un poco las clases para que quede más ordenado.

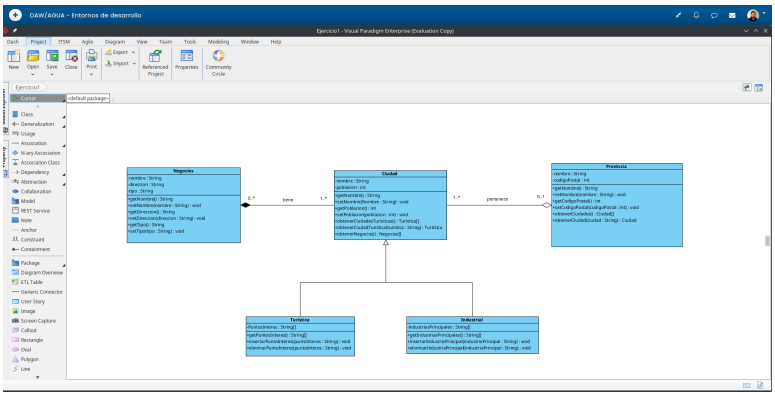

Figura 2.2: Diagrama creado en Visual Paradigm

2.2. Ejercicio 2

2.2.1. Enunciado

Importación del proyecto VP-UML en un proyecto de NetBeans. (Breve descripción de los pasos a seguir incluyendo capturas de pantalla).

2.2.2. Solución

En este ejercicio vamos a importar el proyecto que hemos creado en el anterior en VP a Netbeans. Para ellos, vamos a seguir una serie de pasos los cuales describimos a continuación.

1. En primer lugar, tenemos que realizar la **integración** de Visual Paradigm en Netbeans. Para ello abrimos VP y pulsamos en la opción del menú superior "*Windows*, lo cual nos mostrará diversas opciones en la barra superior. Tenemos que pulsar en la opción "*Integration* y dentro de las dos opciones que se nos despliegan, pulsar en "*IDE Integration...*, como podemos ver en la siguiente captura.

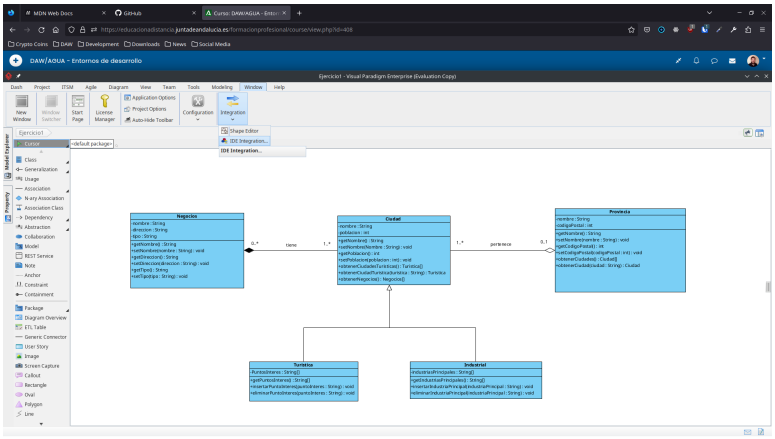

Figura 2.3: Integración de VP en Netbeans (1)

Una vez pulsada en la opción anterior, se nos mostrará una ventana donde deberemos seleccionar el IDE en el que queremos realizar la integración, en nuestro caso, seleccionaremos **Netbeans** y pulsamos en "*Next*".

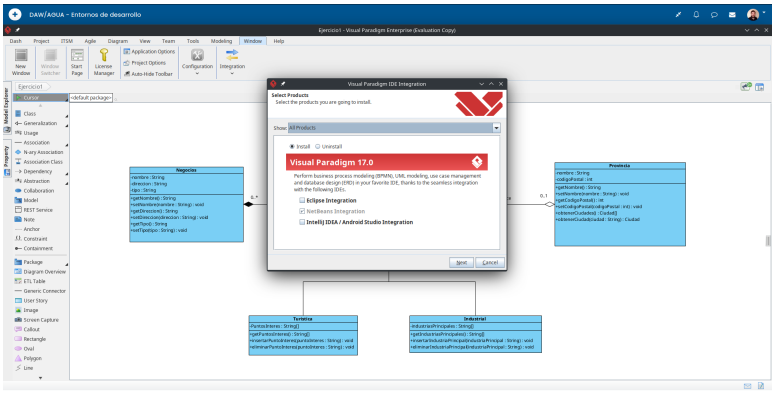

Figura 2.4: Integración de VP en Netbeans (2)

Una vez hecho esto, se copiarán todos los archivos necesarios en el directorio de instalación de Netbeans y la integración se habrá realizado con éxito.

2. El siguiente paso deberemos darlo en Netbeans. En primer lugar abrimos **Netbeans** y **creamos un proyecto**, ya que para poder usar VP en Netbeans debemos hacerlo sobre un proyecto que hayamos creado. Para ello, pulsamos en la opción "*File–>New project*" del menú de la barra superior de Netbeans, lo que nos abrirá la ventana para la creación del nuevo proyecto.

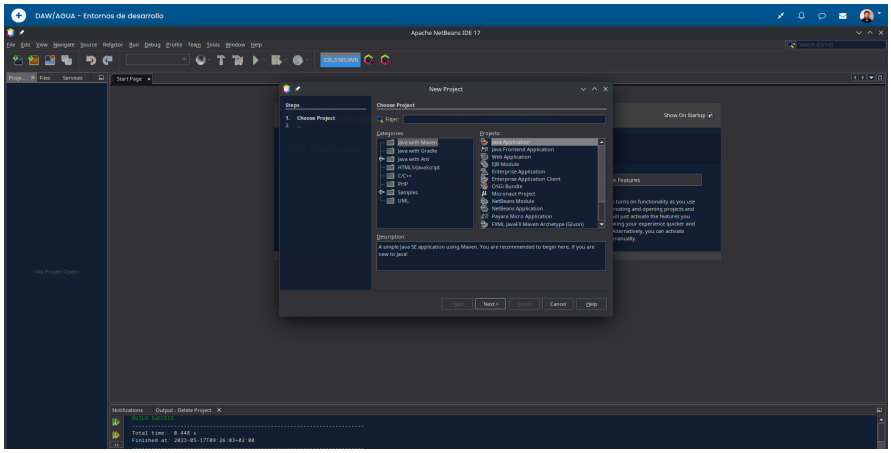

Figura 2.5: Creación de proyecto en Netbeans (1)

En la ventana que se nos abre, seleccionamos la opción "*Java Aplication*" para crear una aplicación java y pulsamos en "*Next*", lo que nos mostrará la siguiente ventana donde podemos incluir información sobre el proyecto, como su nombre, la carpeta donde se va a almacenar, etc... Nosotros hemos nombrado el proyecto según las especificaciones de la tarea con nuestro nombre y apellidos.

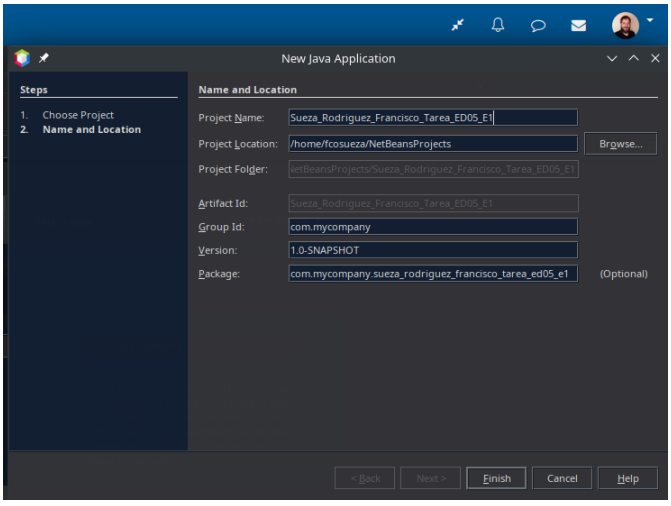

Figura 2.6: Creación de proyecto en Netbeans (2)

Una vez introducidos los datos, el proyectos se creará con los archivos por defecto y se nos abrirá automáticamente el fichero principal, donde se encuentra el método main.

3. A continuación, debemos **inicializar VP en Netbeans**. Para ello, pulsamos con el botón derecho sobre el proyecto que acabamos de crear y pulsamos en la opción "*Open Visual Paradigm Enterprise*.

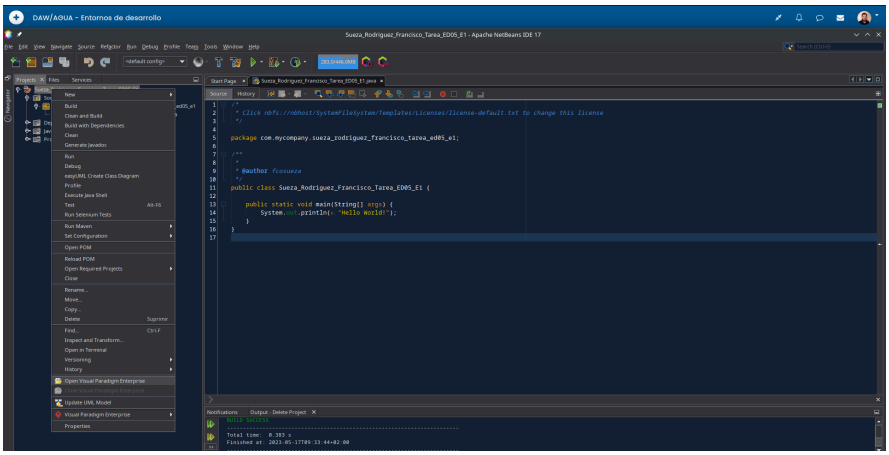

Figura 2.7: Inicialización de VP en Netbeans (1)

Una vez pulsado, **comenzará la carga de Visual Paradigm**, lo que se nos mostrará con la ventana de carga por defecto de esta aplicación. Este proceso puede tardar más o menos, ya dependiendo de las especificaciones de hardware de tu dispositivos o del SO donde lo estés realizando.

Una vez realizada la carga de VP, nos aparecerán nuevas opciones, en la barra superior de Netbeans, para la creación de los diferentes tipos de diagramas, así como nuevas pestañas encima del proyecto con diferentes opciones, etc.., como podemos ver en la siguiente captura de pantalla.

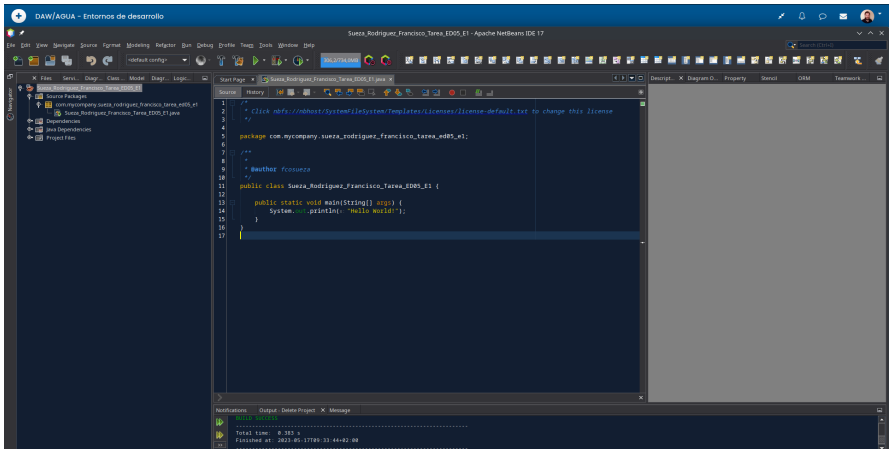

Figura 2.8: Inicialización de VP en Netbeans (2)

4. El siguiente paso será **importar el proyecto de VP**. Para ello, tenemos que pulsar el la opción "*File*" del menú superior y dentro de este en las opciones "*Visual Paradigm Enterprise Import–>UML Model*", como vemos en la siguiente imagen.

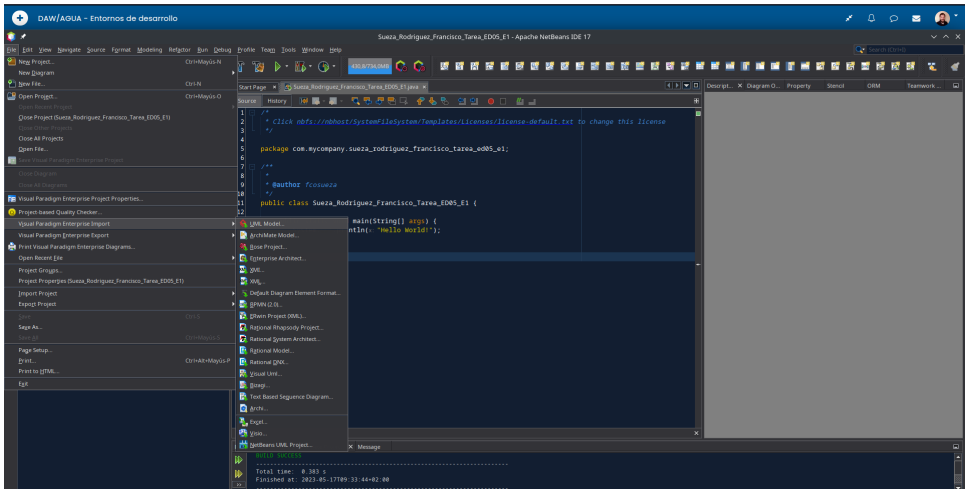

Figura 2.9: Importación del proyecto VP (1)

Esto nos mostrará una ventana donde podemos **seleccionar el archivo** que queremos importar. Seleccionamos el archivo de nuestro proyecto de VP y pulsamos el "*Open*".

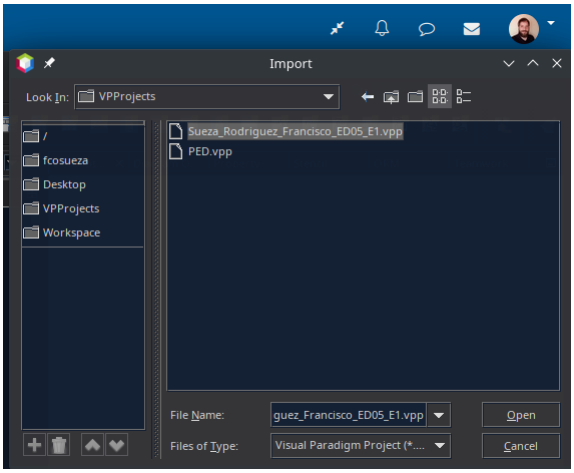

Figura 2.10: Importación del proyecto VP (2)

5. Una vez realizado el paso anterior, el proyecto se habrá **importado correctamente**, pero aún no veremos el proyecto en la pantalla de Netbeans.

   Para que podamos ver el proyecto, debemos seleccionar la pestaña "*Diagramas*", que nos habrá aparecido encima de la ventana del proyecto cuando cargamos VP. En esta pestaña, deberemos seleccionar la opción "*Class Diagram*" y hacer doble-click sobre el nombre de nuestro proyecto VP.

   Hay que tener en cuenta que el nombre que aparecerá no es el del archivo de nuestro proyecto, sino el nombre que le hayamos puesto al proyecto dentro de las propiedades en VP.

   Una vez hecho esto, nuestro proyecto de VP se mostrará en la ventana principal de Netbeans, como vemos en la siguiente captura.

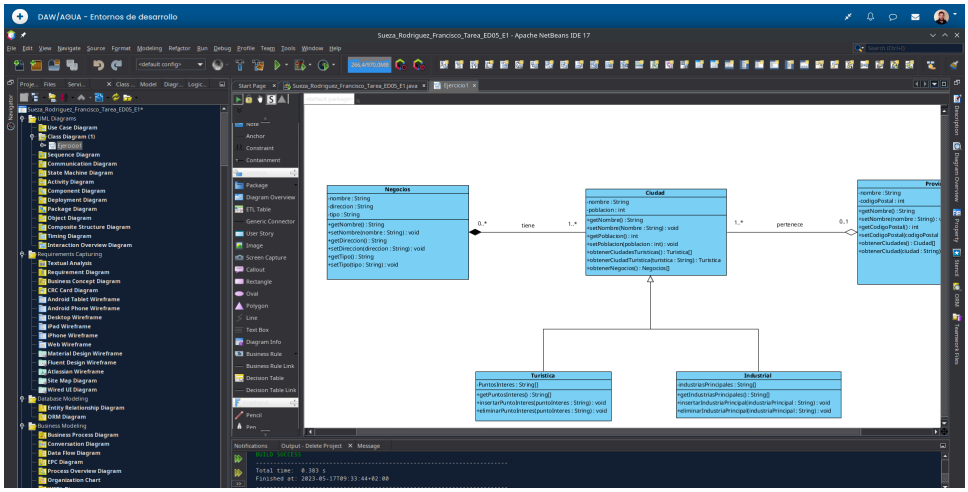

Figura 2.11: Proyecto VP cargado en Netbeans

2.3. Ejercicio 3

2.3.1. Enunciado

Generación del código a partir del diagrama de clases realizado.

2.3.2. Solución

En este ejercicio vamos a generar el código Java a partir del diagrama que hemos creado. Podemos hacerlo desde Visual Paradigm o desde Netbeans. Para que se continúe con el ejercicio anterior, y ya que no se especifica en el enunciado del ejercicio de que forma hacerlo, nosotros lo vamos a hacer desde **Netbeans**, ya que así, además, tendremos el proyecto Java generado cargado directamente en nuestro IDE.

Para generar el código, partiendo de la pestaña "*Diagramas*" que se nos creo encima de la ventana del proyecto al cargar VP en Netbeans, pulsamos en el **icono azul** justo encima de la ventana donde seleccionamos los diagramas, como vemos en la siguiente imagen, y seleccionamos la opción "*Update code*".

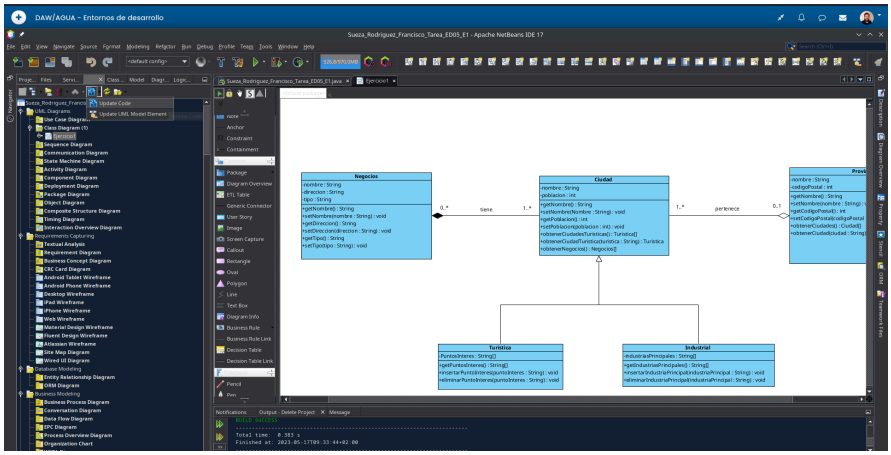

Figura 2.12: Generación de código a partir del proyecto VP (1)

Es posible que nos muestre algún error, si hay algo que no esta bien en el diagrama, como puede ser un tipo de retorno erróneo en algún método. Si fuera el caso, corregimos el error, y volvemos a pulsar en esta opción. Esto, generará un conjunto de clases acordes al diagrama de clases. Es posible que algunas relaciones no se hayan traducido, por lo que tendremos que hacerlo de forma manual.

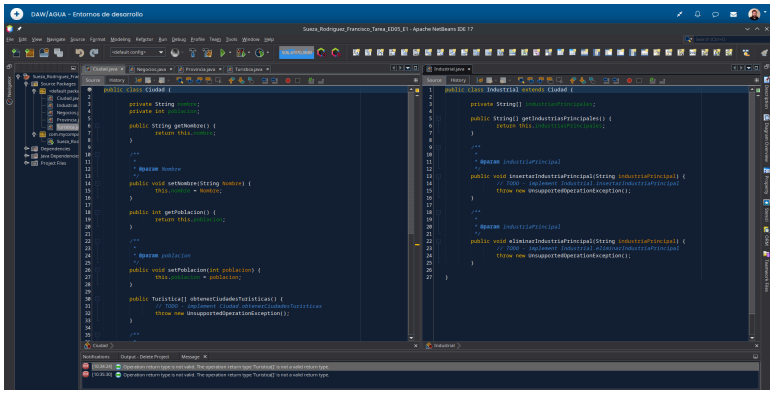

Figura 2.13: Generación de código a partir del proyecto VP (2)

2.4. Ejercicio 4

2.4.1. Enunciado

Mediante el proceso de ingeniería inversa, genera el diagrama de clases a partir del siguiente proyecto Java. Si al realizar este proceso faltan relaciones, tendrás que establecerlas a mano. (No olvides refactorizar y cambiar XXX por tus datos para cumplir el requisito de corrección). Puedes hacer uso del plugin EasyUML (enlaces en la sección Información de interés).

2.4.2. Solución

En este ejercicio vamos a realizar el proceso de ingeniería inversa con el proyecto Java que se nos ha proporcionado. Para realizar este proceso, vamos a usar el plugin de Netbeans **EasyUML**, el cual podemos descargar desde `https://github.com/mgeee35/easyUML`.

Este proceso es bastante sencillo, y solo debemos seguir los siguientes pasos para llevarlo a cabo.

1. En primer lugar, y teniendo ya el proyecto Java que se nos especifica cargado, debemos **crear otro proyecto**, de tipo **UML** en Netbeans. Para ello, pulsamos en la opción "*File–>New Project...* del menú superior y seleccionamos UML como el tipo de proyecto a crear.

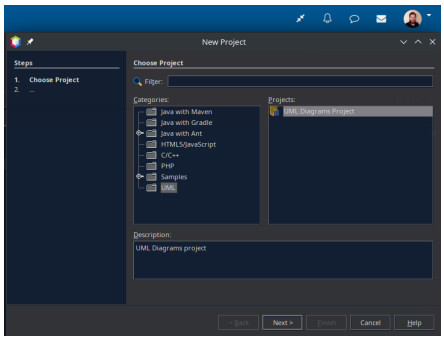

Figura 2.14: Ingeniería Inversa: Creación proyecto UML

2. Una vez creado el proyecto UML, pulsamos sobre el **proyecto Java** con el botón derecho para que se despliegue el menú contextual y seleccionamos la opción "*easyUML Create Class Diagram*".

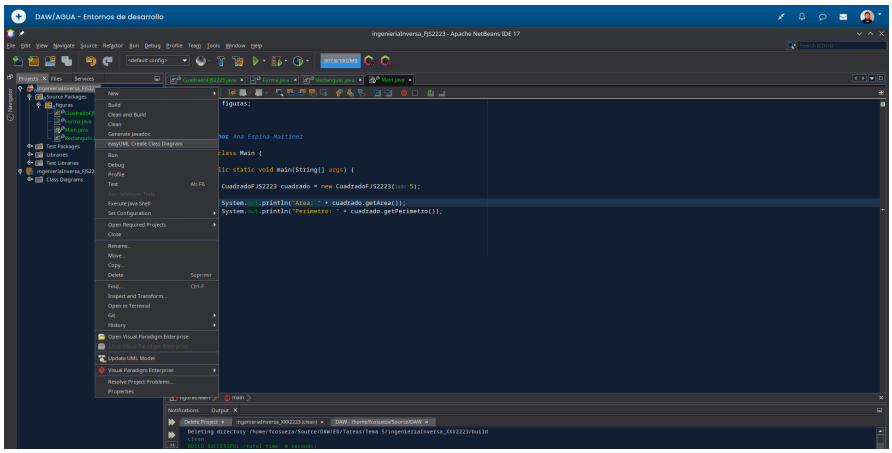

Figura 2.15: Ingeniería Inversa: Seleccionar opción de easyUML

3. Esto nos mostrará una ventana donde debemos seleccionar el proyecto UML donde queremos que se generen los diagramas, que será el proyecto que hemos creado.

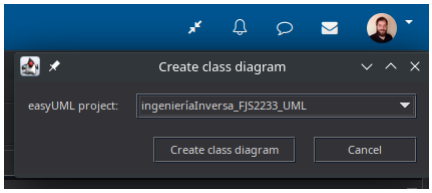

Figura 2.16: Selección de proyecto UML

4. Tras el último paso, se nos habrá generado correctamente el **diagrama de clases** del proyecto Java. Podría ser que algunas relaciones falte en el diagrama generado, aunque no es nuestro caso, ya que se muestran las relaciones entre las clases correctamente, como podemos ver en la siguiente captura.

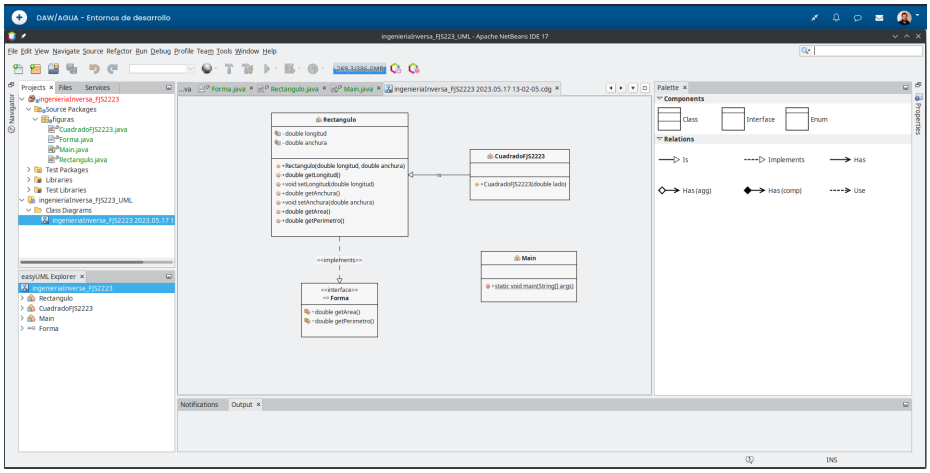

Figura 2.17: Diagrama generado con easyUML

NOTA: en la última captura se muestra Netbeans con otro tema diferentes, ya que con el tema anterior, Dark Metal, no se apreciaban correctamente el nombre de las clases generadas con easyUML.

3. Parte 2: Diagramas de Comportamiento

3.1. Ejercicio 5

3.1.1. Enunciado

Identifica el tipo de diagrama e interpreta lo que está representando sobre pedidos de pizza.

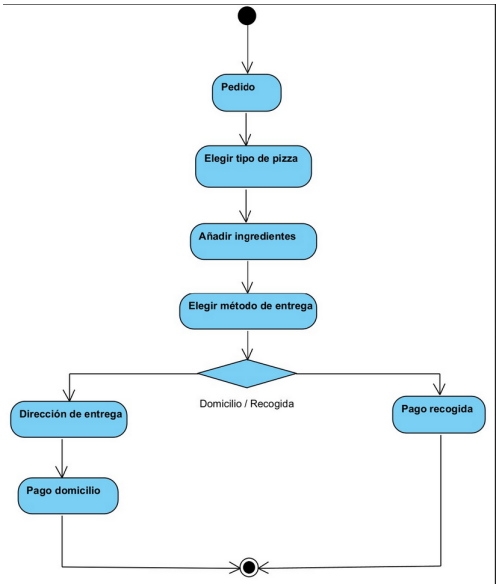

Figura 3.1: Diagrama de comportamiento

3.1.2. Solución

El diagrama que se muestra es una **diagrama de estados**, que muestra el comportamiento de un sistema en respuesta a diferentes eventos y los estados en los que se encuentra este en cada después de dichos eventos. En nuestro caso, el diagrama muestra un sistema para realizar pedidos de pizzas.

En este sistema podemos encontrar los siguientes estados:

- **Pedido**: es el primer estado después del inicial, que representa el inicio del proceso de pedido.
- **Elegir tipo de pizza**: es el siguiente estado, donde se deberá elegir el tipo de pizza que se quiere pedir.
- **Añadir ingredientes**: tras haber seleccionado el tipo de pizza, en este estado podemos añadir ingredientes a esta.
- **Elegir método de entrega**: en este estado debemos realizar la elección del método de entrega de la pizza, dependiendo del método que elijamos, el sistema pasará a uno estado u otro, lo que se representan con un rombo y las dos opciones que podemos elegir: **Domicilio** ó **Recogida**.
- Si **elegimos Recogida**:

- **Pago en Recogida**: en este estado el sistema establecerá que el pago se realizará a la recogida de la pizza en el local.
- **Estado final**: la ejecución de la operación a concluido.

- Si elegimos **domicilio**:
  - **Dirección de entrega**: en este estado se debe introducir la dirección de entrega de la pizza, para poder llevarla al domicilio.
  - **Pago domicilio**: aquí se establece el método de pago de la pizza para que se realice en el domicilio a la entrega de esta.
  - **Estado final**: la ejecución de la operación a concluido.

3.2. Ejercicio 6

3.2.1. Enunciado

Realiza el diagrama de estados para la siguiente operatoria de un ascensor:

El ascensor dispone de los siguientes estados: ReposoXXX2223, Subiendo, Bajando, Puertas abiertas y Puertas cerradas.

La posición inicial parte del estado Reposo, en el cual se mantiene hasta que se pulsa un botón, pasando al siguiente estado subiendo o bajando, dependiendo de la dirección del piso en el que se ha pulsado el botón o abriendo la puerta si se pulsa el botón.

Cuando llega al piso, vuelve a la posición de Reposo, y abre las puertas, cambiando a este estado. A continuación, cierra las puertas, cambiando a dicho estado, y vuelve a estar en Repos a la espera de que se pulse un botón.

3.2.2. Solución