

TAREA UNIDAD 5
Diseño orientado a objetos. Diagramas



CFGS DAW. Módulo: Entornos de desarrollo (2023/24)
Dayro Morales Cruz

Elaborar un documento de texto en el que se **describa** y visualice a través de capturas de pantalla (**siempre con tu usuario de la plataforma de fondo** en todas ellas), el proceso que has seguido para elaborar cada ejercicio de la tarea.

Cada vez que aparezca en el enunciado de la tarea **XXX** tendrá que ser sustituido por tus datos personales (Apellido1Apellido2Nombre).

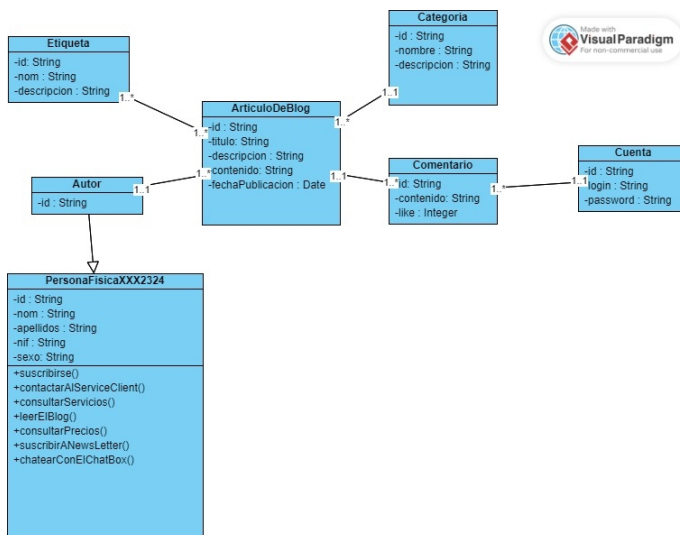
De no realizarse así la tarea no será corregida.

Todos los diagramas UML solicitados en la tarea deben ser creados con Visual Paradigm en un único proyecto que debe llamarse Apellido1_Apellido2_Nombre_ED05.

Parte 1: Diagramas estructurales

Ejercicio 1

Elabora el siguiente diagrama de clases mediante el programa Visual Paradigm 17.1. (Breve descripción de los pasos a seguir incluyendo capturas de pantalla).



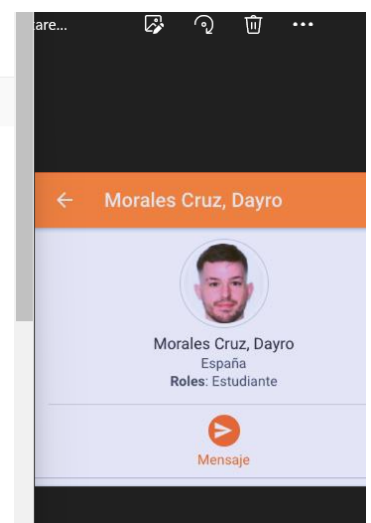
Primero descargamos el programa Visual Paradigm 17.1 desde su [página oficial](#)

Thank you for downloading Visual Paradigm

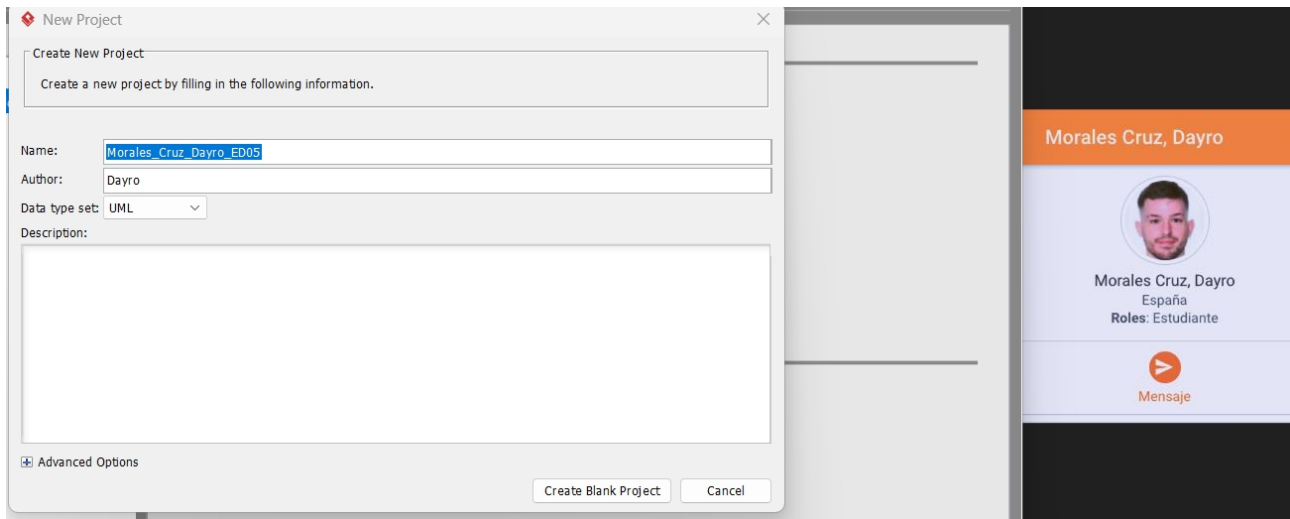
Run the downloaded installer and perform the steps below.

If your download does not begin, please use this [direct link](#)

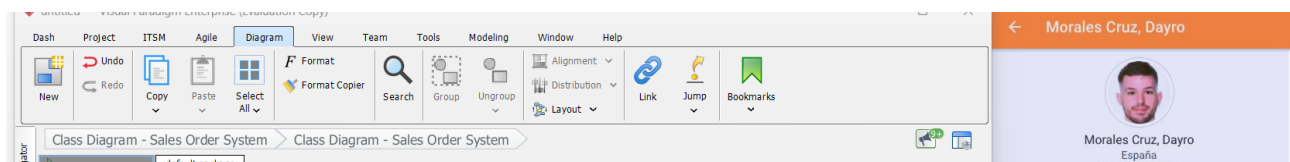
Or you can select other download sites.



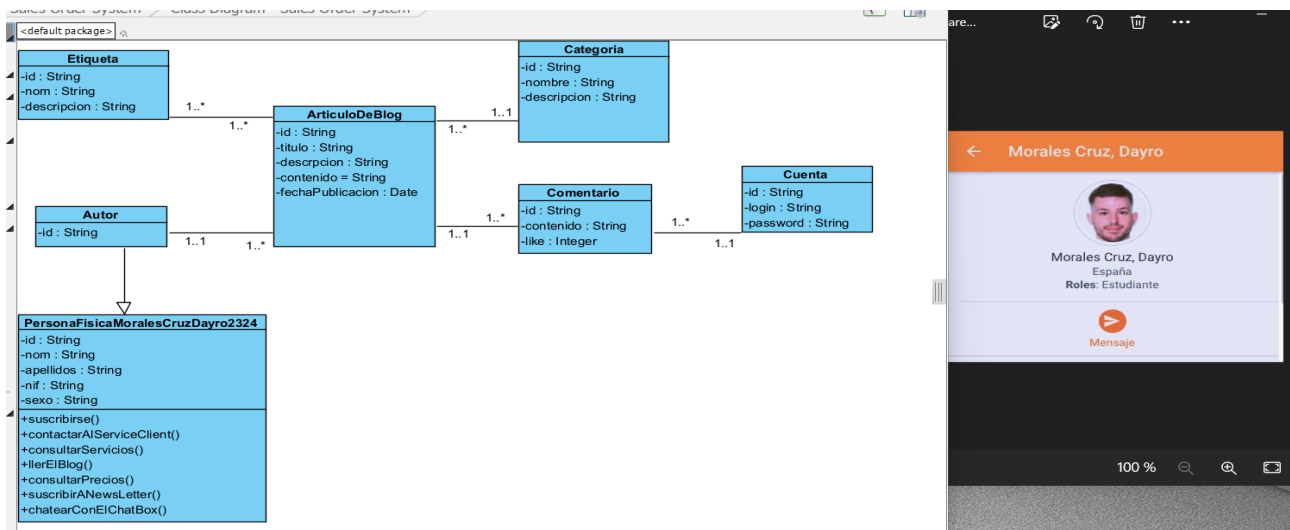
Una vez instalado el programa creamos nuestro proyecto en **New Project**



Para crear nuestro diagramas, vamos a **Diagrams** → **New**



Esta es la captura del diagrama realizado con el programa indicado en la tarea:

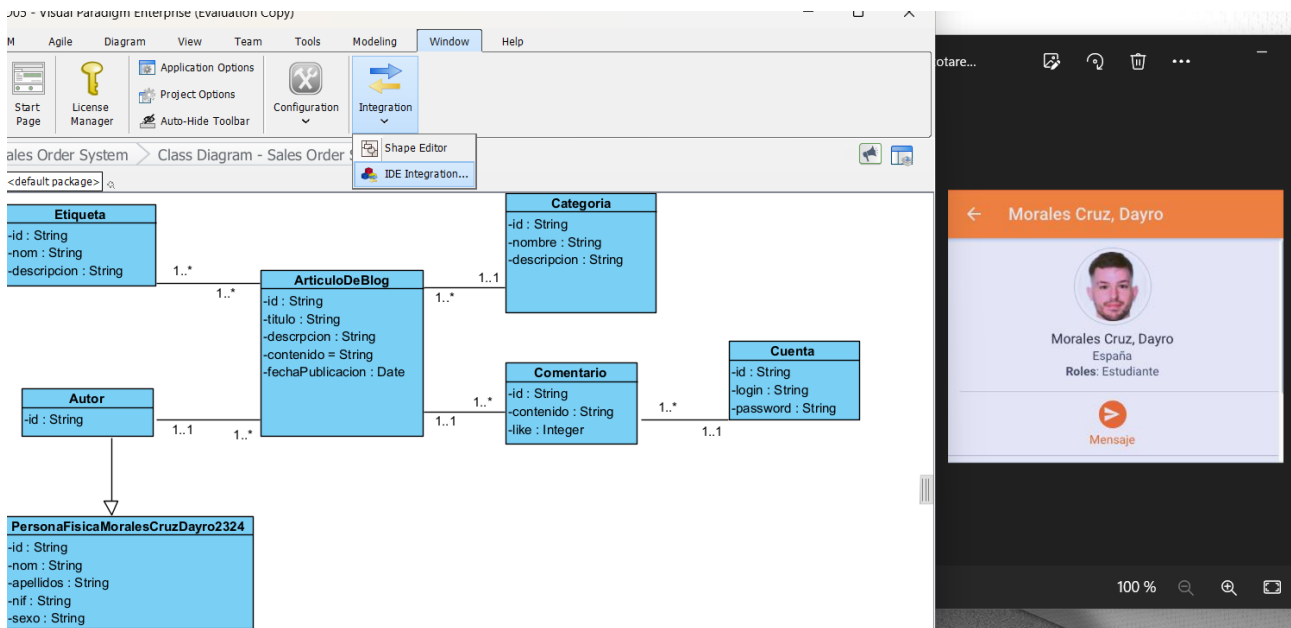


Ejercicio 2

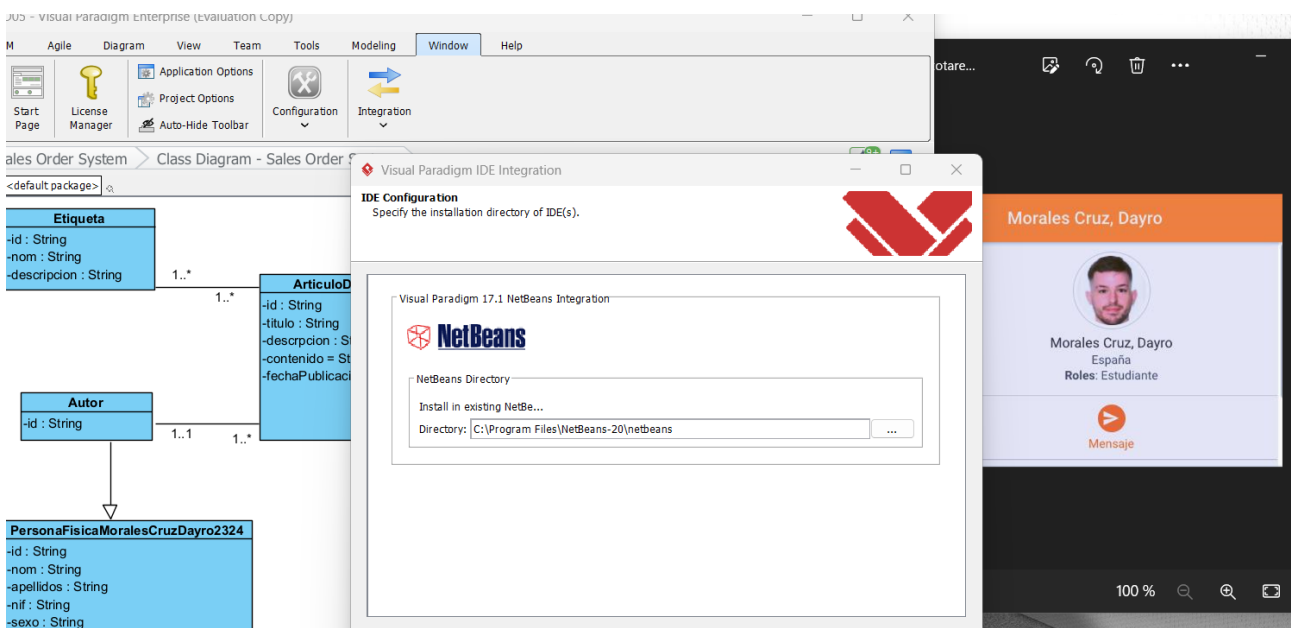
Importación del proyecto VP-UML en un proyecto de NetBeans. (Breve descripción de los pasos a seguir incluyendo capturas de pantalla).

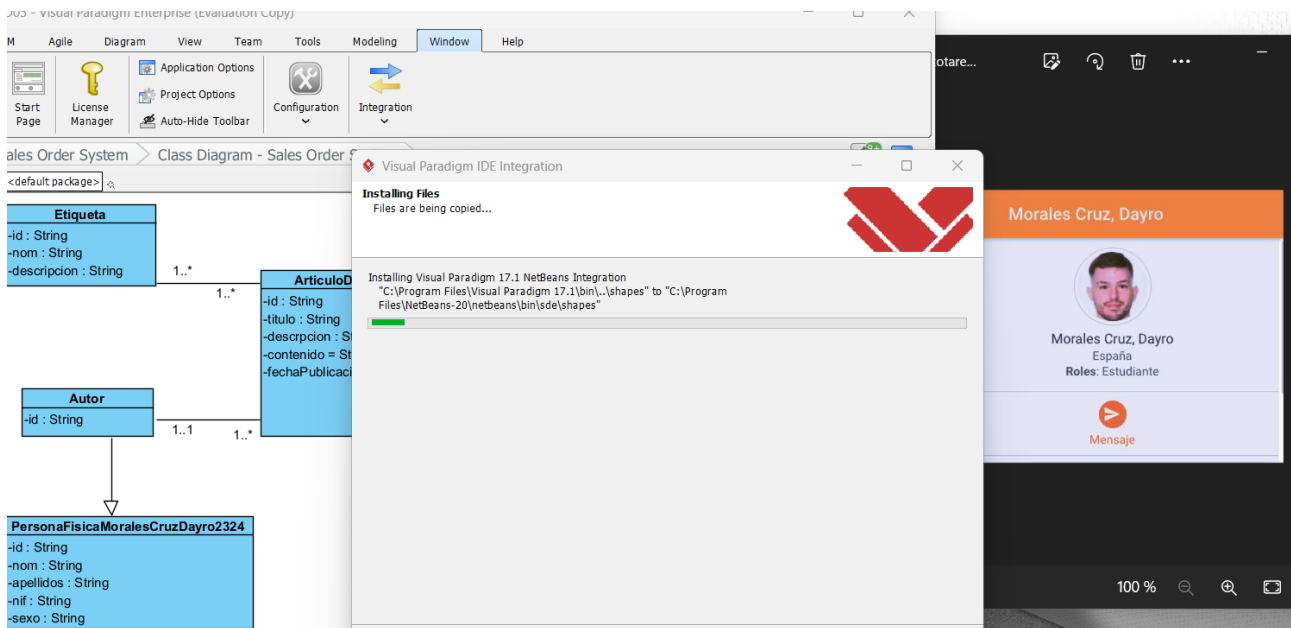
Antes de nada, para poder realizar la importación es necesario realizar la integración de Visual Paradigm con Netbeans desde **Visual Paradigm**.

Vamos a la pestaña **windows** → **Integration** → **IDE Integration**

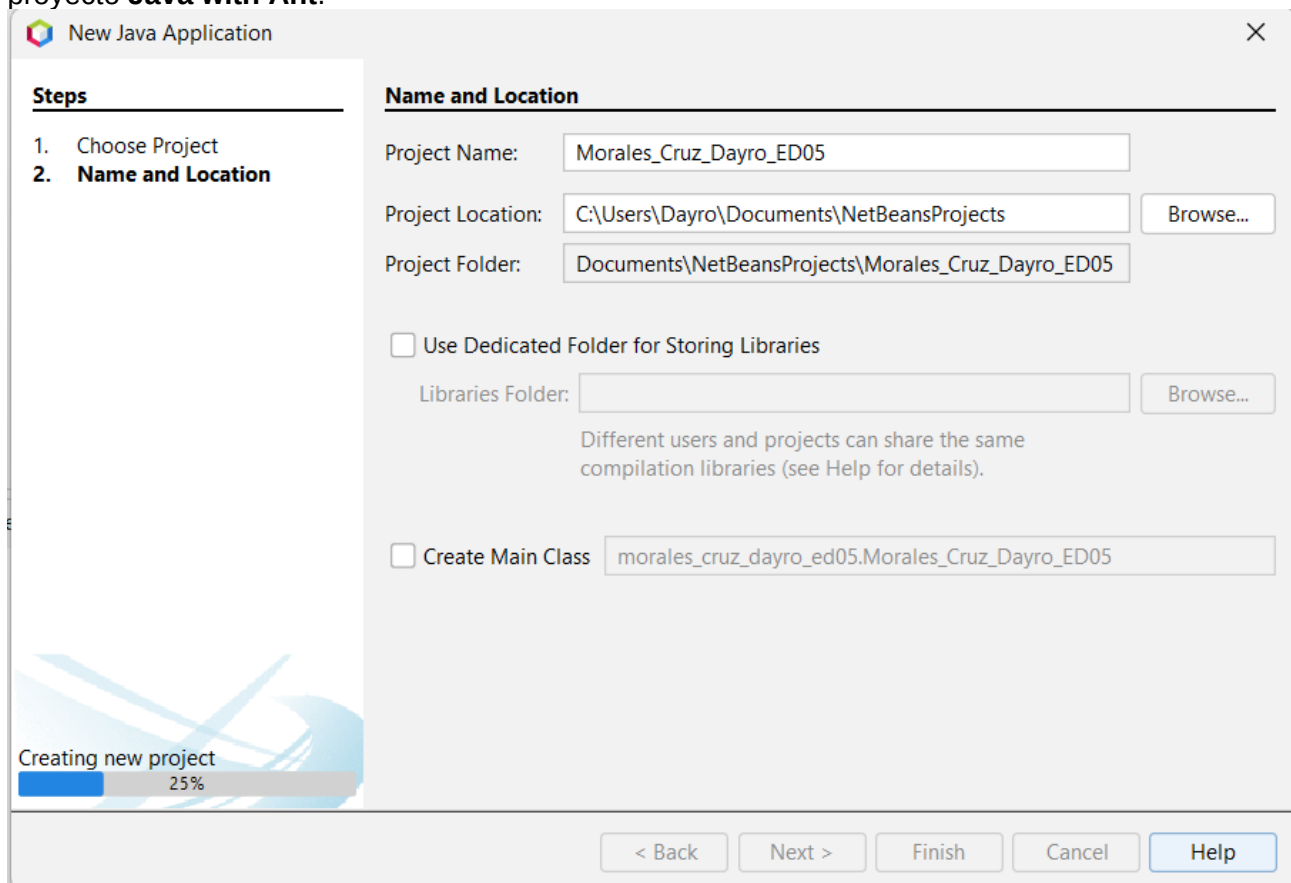


A continuación, deberemos elegir netbeans y meter la ubicación de nuestro IDE y esperamos a que se instale

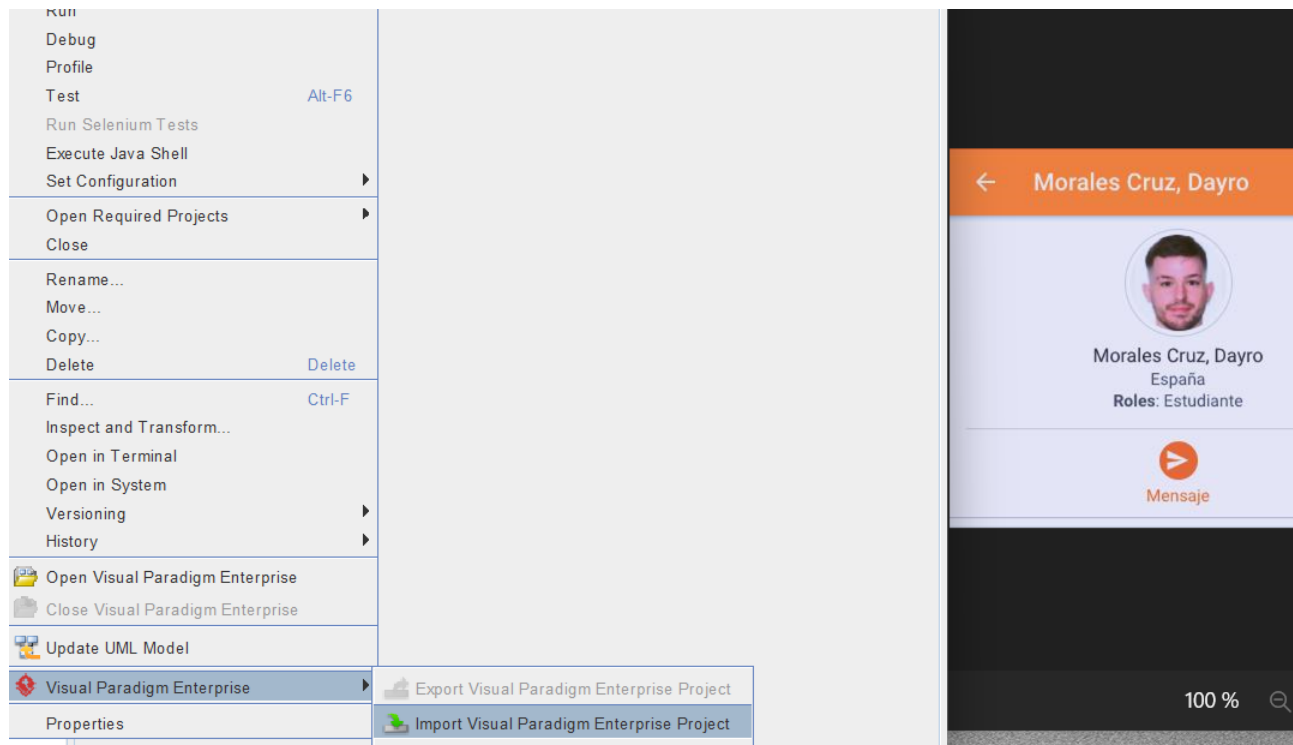




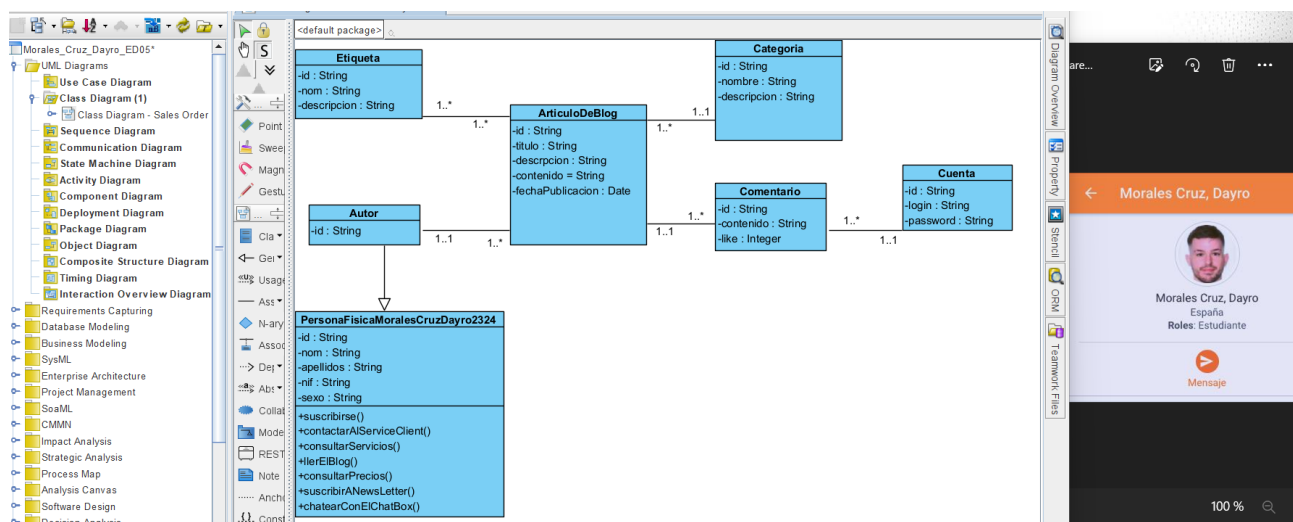
Una vez realizada la integración de Visual Paradigm con NetBeans, creamos un proyecto **Java with Ant**.



Hacemos click con el botón derecho del ratón sobre el nombre del proyecto y seleccionamos **Visual Paradigm Enterprise** → **Import Visual Paradigm Enterprise Project**.



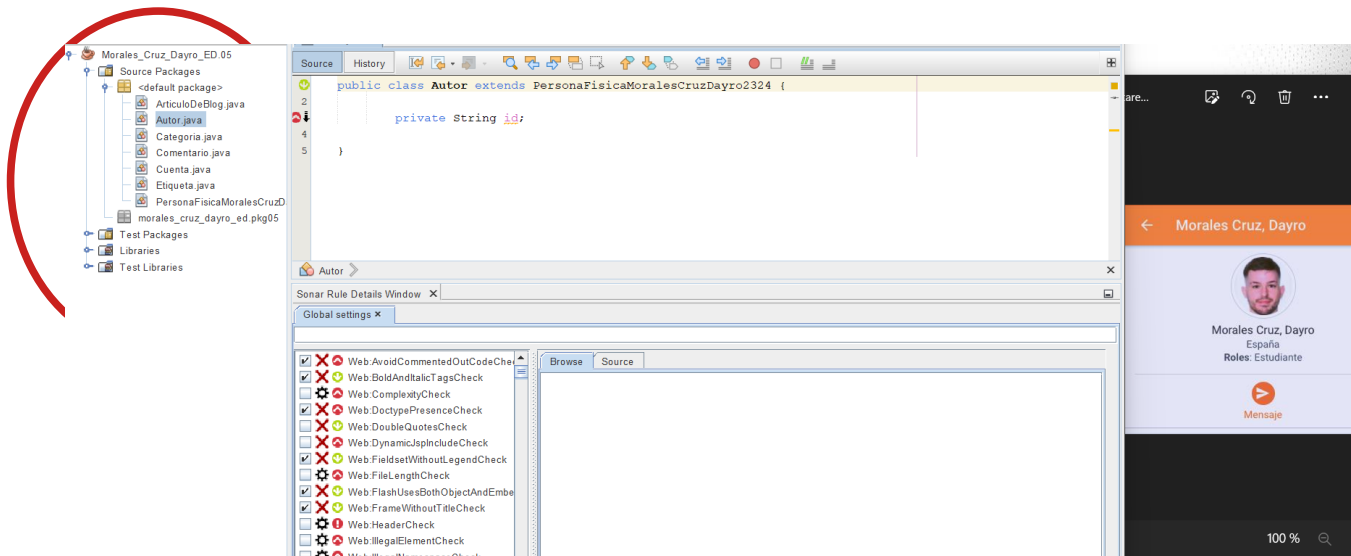
A la derecha de la pestaña Projects tenemos una pestaña denominada Diagram Navigator, en la cual, seleccionando **UML Diagrams** y **Class Diagram** podemos localizar el diagrama que acaba de importar.



Ejercicio 3

Generación del código a partir del diagrama de clases realizado. (Breve descripción de los pasos a seguir incluyendo capturas de pantalla).

En la pestaña **Diagram Navigator** hacemos click con el botón derecho del ratón sobre el nombre del proyecto y seleccionamos **Update Project to Code**. Como comprobamos en la captura, se ha generado todas las clases del código correctamente.

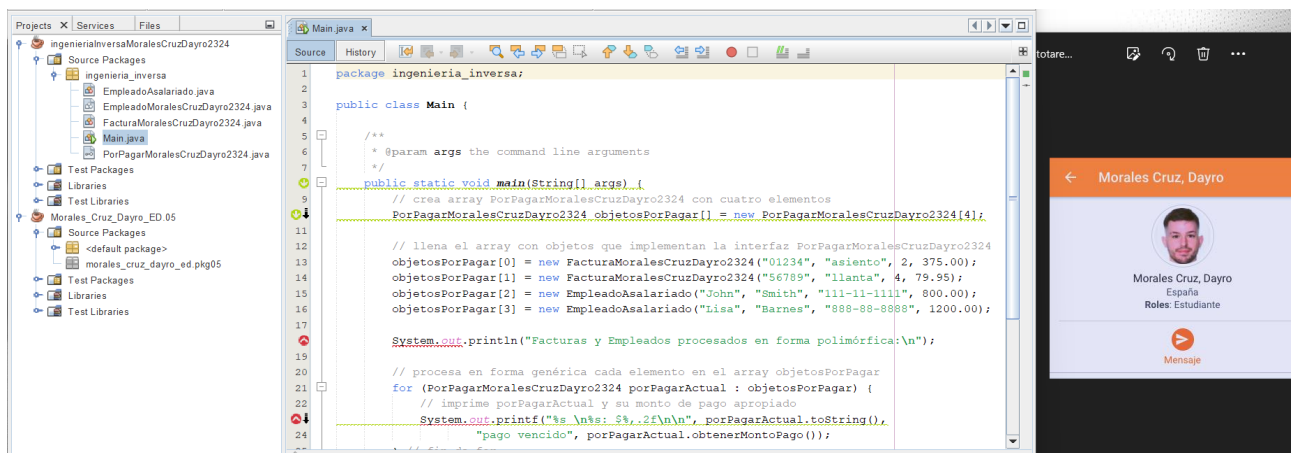


Ejercicio 4

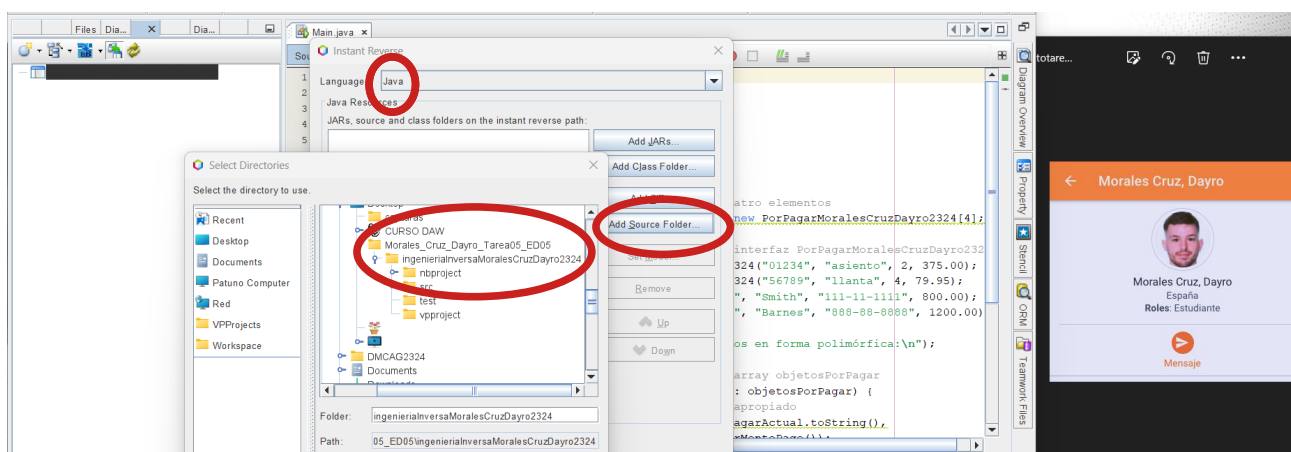
Mediante el proceso de ingeniería inversa, genera el diagrama de clases a partir del siguiente [proyecto Java](#). Si al realizar este proceso faltan relaciones, tendrás que establecerlas a mano. (No olvides refactorizar y cambiar **XXX** por tus datos para cumplir el **requisito de corrección**). Puedes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

Una vez importamos el proyecto a NetBeans, haciendo click con el boton derecho seleccionamos la opción **Update UML Model** y refactorizamos sustituyendo **XXX** por **MoralesCruzDayro**.

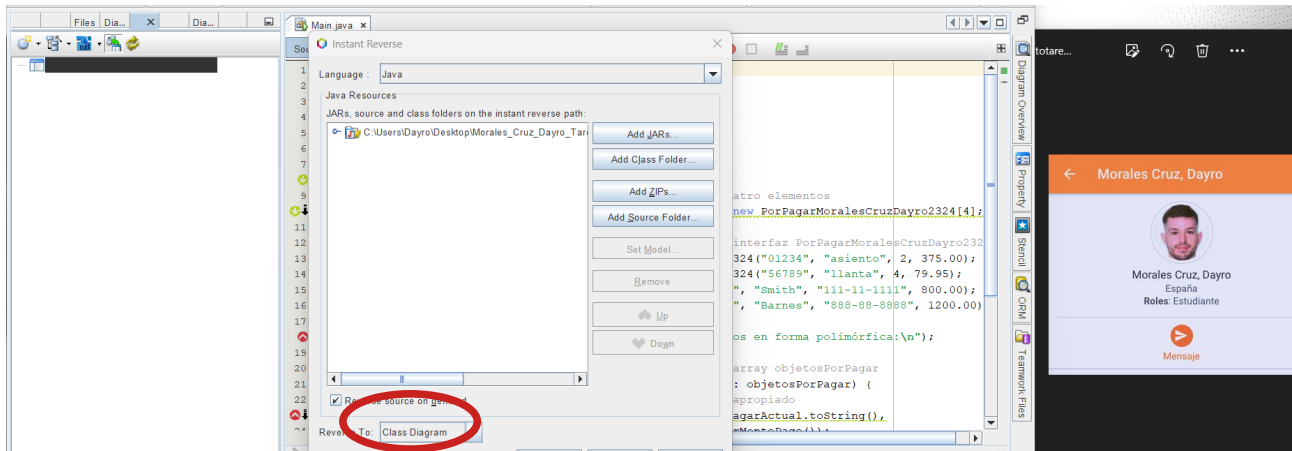
Cuando finalice, comprobamos que se generó correctamente.



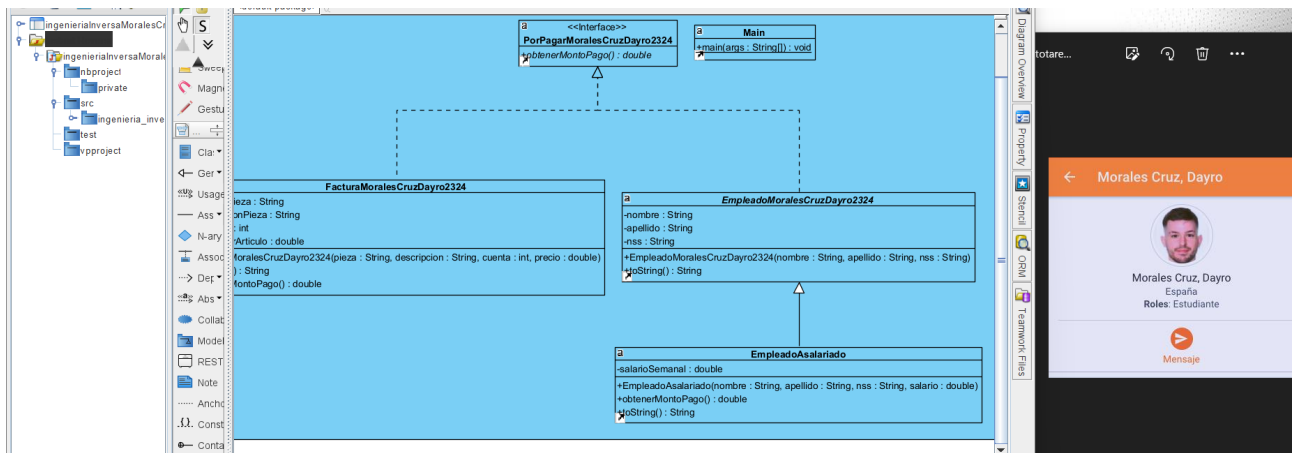
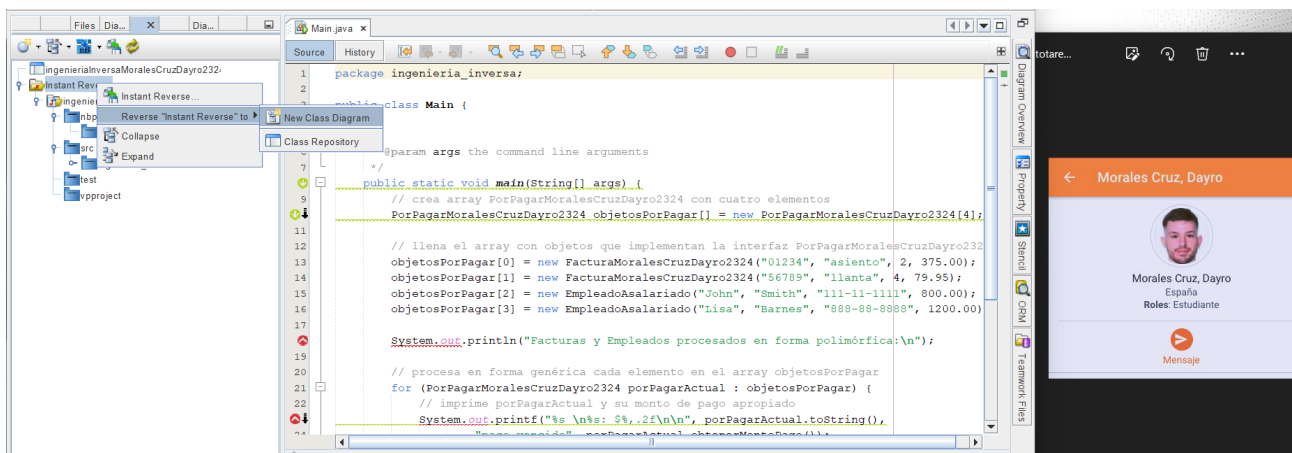
A continuación, hacemos click sobre **Instant Reverse** y pulsamos sobre **Add Source Folder** y seleccionamos nuestro proyecto.



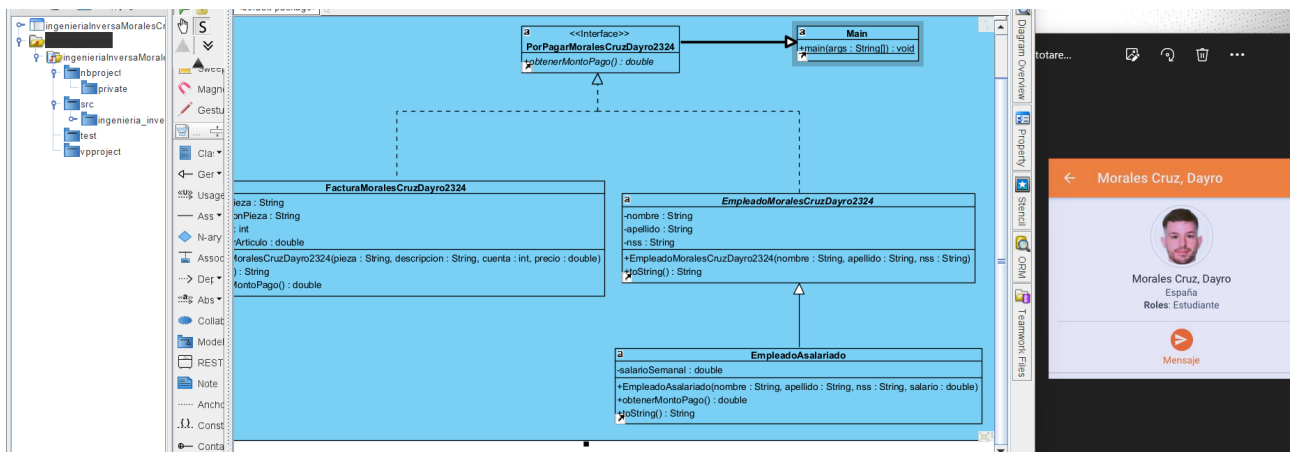
Nos aseguramos de tener seleccionado el lenguaje **Java**



Podemos comprobar que ha generado el inverso y haciendo click derecho sobre la carpeta seleccionamos la opción **Reverse "Instant Reverse to" → New Class Diagram** y generamos el diagrama UML automaticamente



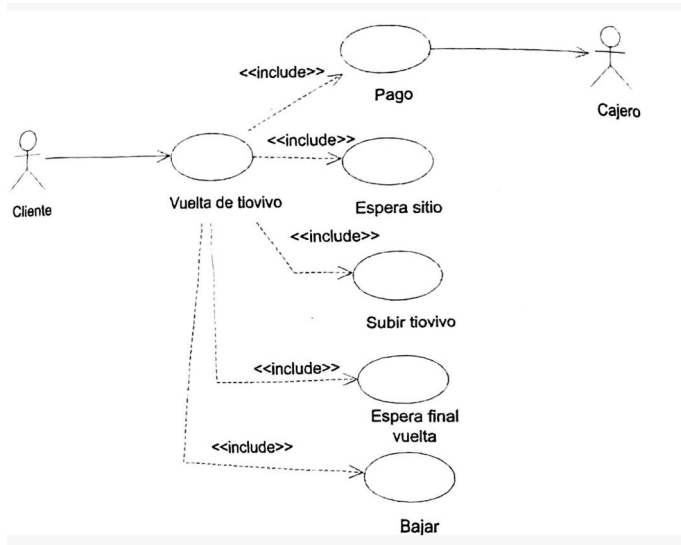
Sólo tendríamos que relacionar el **Main** con **PorPagarMoralesCruzDayro2324**



Parte 2: Diagramas de comportamiento

Ejercicio 5

Identifica el tipo de diagrama e interpreta lo que está representando:



El diagrama mostrado en la imagen anterior se trata de un **Diagrama de Caso de Uso** ya que es una representación que muestra como los usuarios interactúan con un sistema y cuáles son esas funciones.

El diagrama incluye:

- **Actores:** Existen 2 actores *Cliente* y *Cajero*. Son los que interactúan con el sistema
- **Casos de Uso:** Son las acciones que los actores pueden realizar. Los casos son: *Pago*, *Espera sitio*, *Subir tiovivo*, *Espera final vuelta* y *Bajar*
- **Inclusión:** La acción principal titulada *Vuelta de tiovivo* está conectada a 4 acciones secundarias (líneas punteadas) y las etiquetas *<include>*.

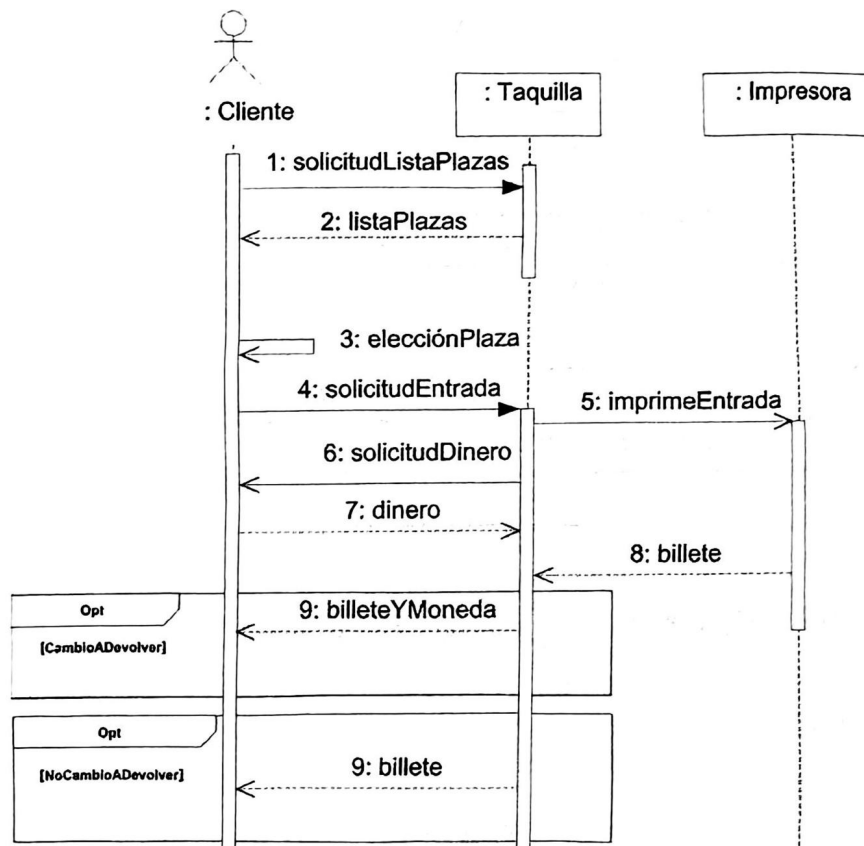
El diagrama dice que un cliente realiza un pago, espera un sitio, sube al tiovivo, espera que termine la atracción y luego baja del tiovivo.

Todo forma parte de *Vuelta de tiovivo*


El *Cajero* forma parte del proceso de pago

Ejercicio 6

En el mismo proyecto en el que se realizó el ejercicio 1 crea el siguiente diagrama: (Breve descripción de los pasos a seguir incluyendo capturas de pantalla).




Elegimos un tipo de diagrama Secuencial



Sequence Diagram

Model how users, objects and systems interact with one another and in what order. Interactions are arranged from top to bottom, following their order of occurrence.



Arrow Diagram

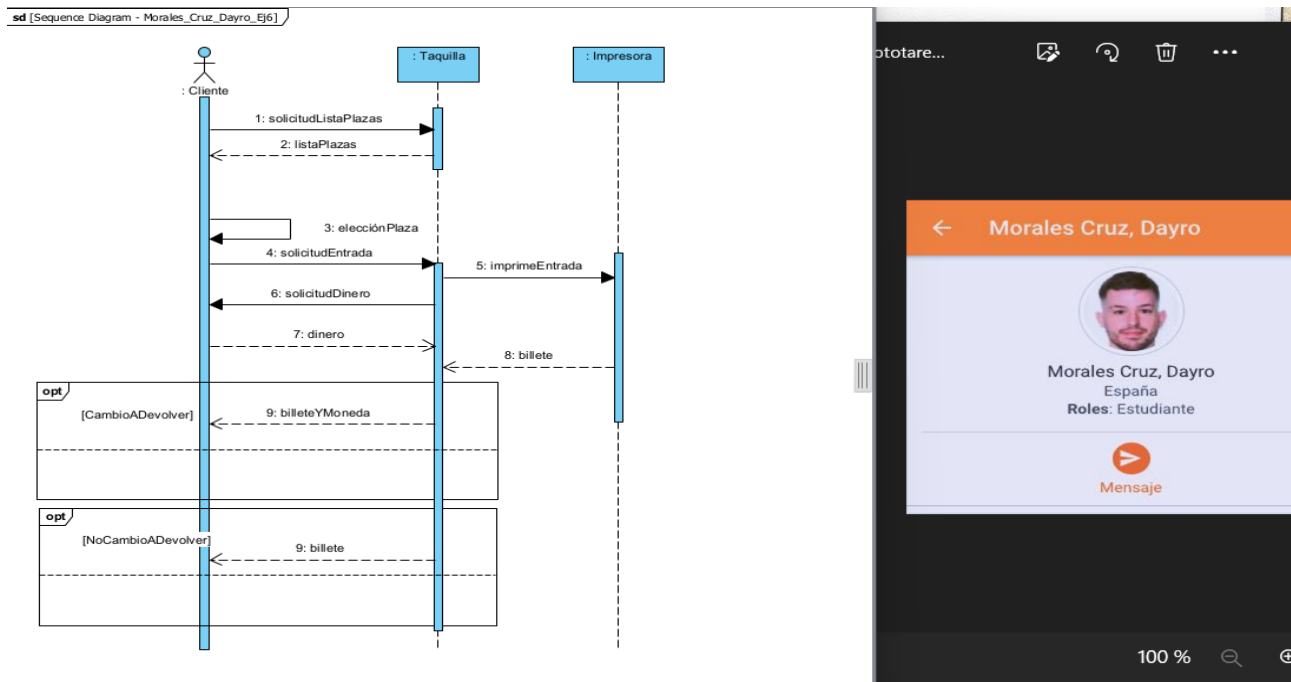
Determine optimal sequence of events, and the interconnectivity also known as activity network diagram, node diagram & CPM chart.

Morales Cruz, Dayro



Morales Cruz, Dayro
España
Roles: Estudiante

Este es el diagrama realizado en Visual Paradigm:





Ejercicio 7


Realiza el diagrama de estados en Visual Paradigm (dentro del proyecto anterior) para la siguiente operatoria de una máquina de café de cápsulas: (Breve descripción de los pasos a seguir incluyendo capturas de pantalla).

Cuando se pulsa el botón la máquina empieza a calentarse, y al pasar 15 segundos la máquina ya está preparada. Si mientras está preparada se vuelve a pulsar el botón empieza a echar café durante 12 segundos y se para. Si durante los 12 segundos que está echando café se pulsa el botón, la máquina deja de echar café (por si se quiere más corto) y se para, siguiendo preparada por si se quiere hacer más café. Si transcurren 5 minutos sin que se haya hecho ningún café la máquina automáticamente se apaga. Añade al nombre de los estados XXX2324 (recuerda sustituir XXX por tus datos).

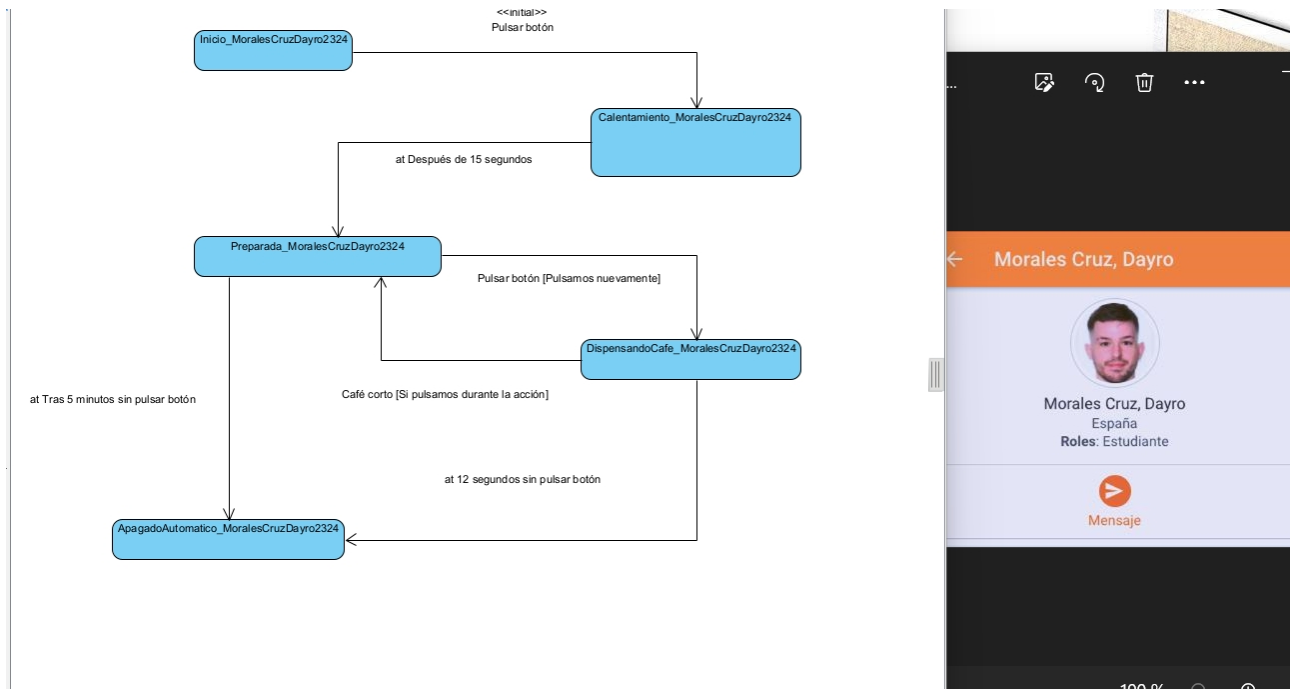
Seleccionamos un tipo de diagrama de estados

**State Machine Diagram**
Model the critical states within a system and the events that may trigger a state change.

**Analysis Canvas**
Innovate and create your own strategic model canvases with VP's configurable canvas designer to fit your business needs for the ever-changing environment.



Aquí la captura del diagrama realizado en Visual Paradigm:



Los pasos detallados son:

1. **Inicio_MoralesCruzDayro2324**: Estado inicial cuando la máquina está apagada.
2. **Calentamiento_MoralesCruzDayro2324**: Al pulsar el botón, transición al estado de calentamiento. Se establece un temporizador de 15 segundos.
3. **Preparada_MoralesCruzDayro2324**: Después de 15 segundos, transición al estado preparada. La máquina está lista para hacer café.
4. **DispensandoCafe_MoralesCruzDayro2324**: Si se pulsa el botón mientras la máquina está preparada, transición al estado dispensando café. Se establece un temporizador de 12 segundos.
5. **Bifurcación** que vuelva a preparada llamada **Café corto** y otra que si pasa **12 segundos** se apaga.
6. **ApagadoAutomatico_MoralesCruzDayro2324**: Si no se hace café durante 5 minutos, transición al estado apagado automático.