

Artefactos de diseño en UML usando StarUML 2

Introducción

Este es un segundo laboratorio de introducción a StarUML en el que crearemos algunos artefactos de diseño.

Objetivos

El objetivo de este laboratorio es el siguiente:

- Utilizar la herramienta StarUML para realizar diagramas de secuencia y obtener el correspondiente diagrama de clases de diseño.

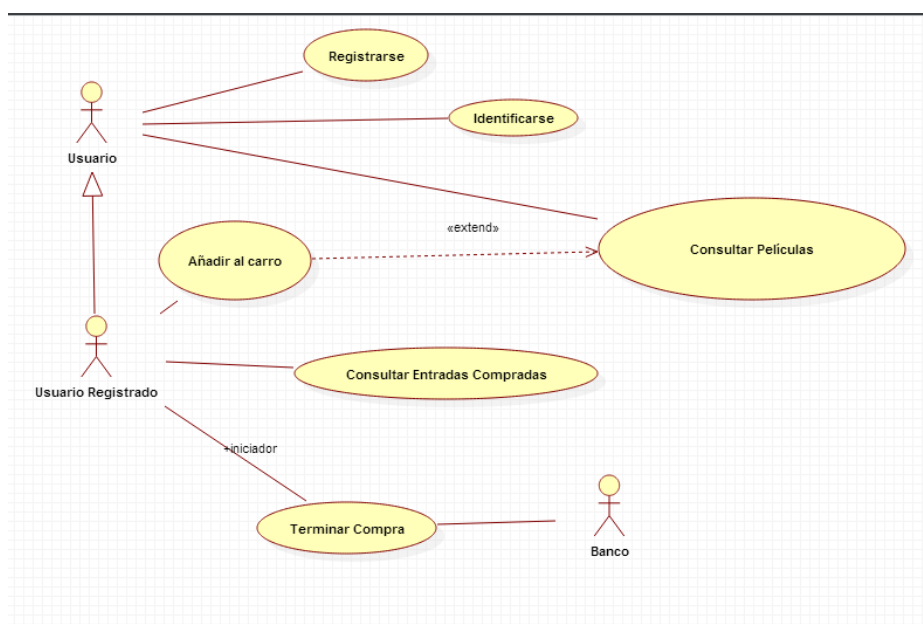
Pasos a seguir:

1. Instalación y ejecución de StarUML.

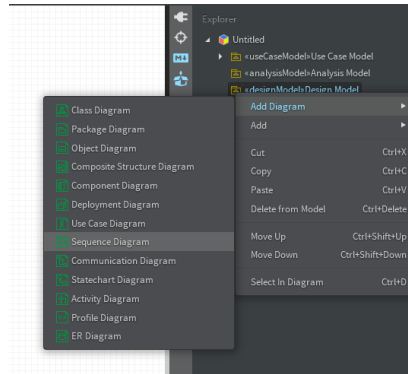
En la siguiente dirección podéis descargar la aplicación: <http://staruml.sourceforge.net/en/download.php>, si es que no está ya instalada.

2. Creación de diagramas de secuencia para los Casos de Uso.

En el sitio eGela de la asignatura está disponible el fichero zinemaldia2.mdj, que hay que descargar (con click dcho. => Guardar enlace, ya que con click izdo. se visualiza el fichero en formato JSON utilizado por StarUML para almacenar los modelos). Dicho fichero contiene el siguiente MCU:



Hacer click derecho en “Design Model” del “Explorer” para crear un nuevo diagrama de secuencia, que se añadirá al “Design Model” de la aplicación y dar el nombre, por ejemplo DS_CP.

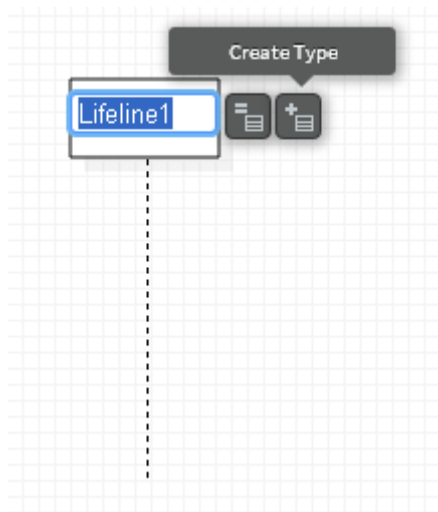


Crear los métodos y las clases:

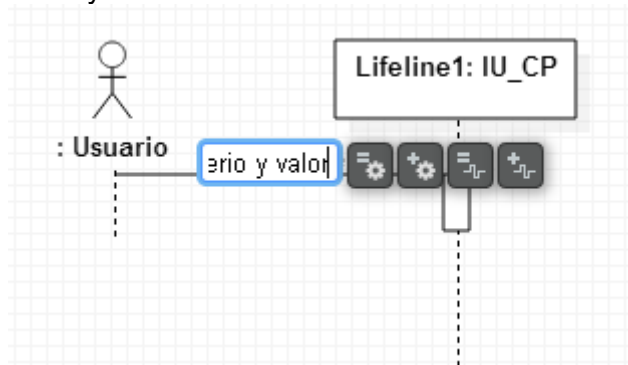
2.1 Crear las clases. A medida que se diseñan los diagramas de secuencia se van definiendo nuevas clases, y utilizando los actores y clases existentes. Por ejemplo, para crear el diagrama de secuencia para el caso de uso cuyo flujo de eventos es el siguiente:

- El usuario proporciona el criterio por el que quiere buscar películas (día, ciclo, título, director o sala), así como el día, o ciclo, o parte del título, o nombre de director o nombre de sala.
- Si el usuario quiere comprar entradas para esa proyección: extends AÑADIR AL CARRO

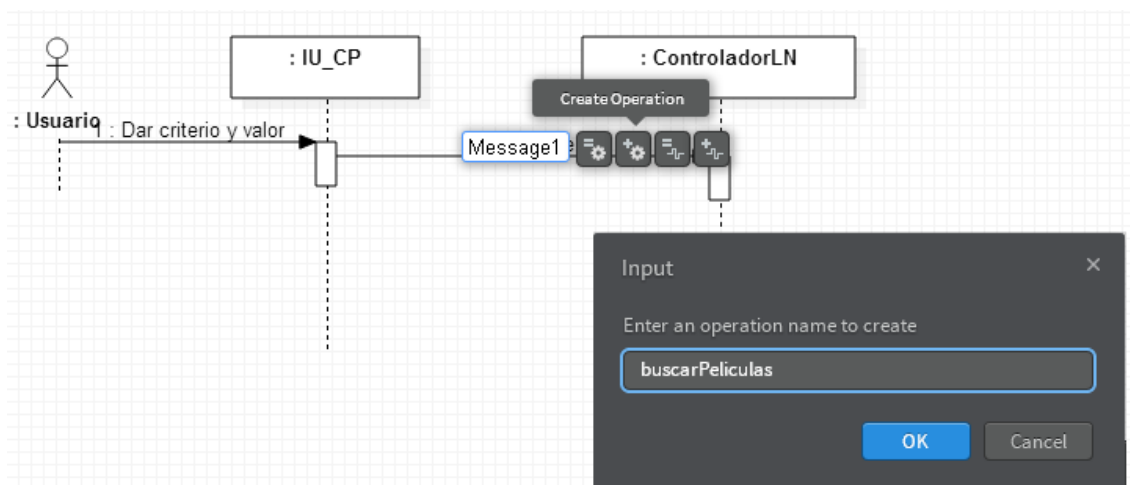
- 1) Arrastrar el actor ya existente “Usuario” desde el Modelo del Dominio que se encuentra en el “Use Case Model” del “Explorer” hasta el diagrama de secuencia. Se le puede quitar el nombre que le da por defecto “Lifeline1” de la propiedad name de properties.
- 2) Añadir un objeto de una nueva clase (arrastrando un objeto “Lifeline” del toolbox “Interactions (Basic)”) y darle el nombre de la interfaz, por ejemplo; IU_CP (tras posicionarse en el icono de clase que aparece a la derecha del objeto y que está anotado con “Create Type”). Se puede quitar también el nombre “LifeLine1” del objeto.



- 3) Crear una interacción entre el actor y la interfaz añadiendo un mensaje, Message del Toolbox Interactions (Basic), entre ambos y anotándolo con el texto “Dar criterio y valor”:

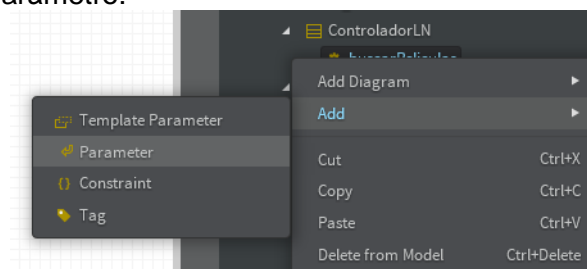


2.2 Definir los métodos. Crear un objeto de una nueva clase Controlador_LN (arrastrándolo desde “Lifeline” de “Interactions (Basic)” en “Toolbox”) y un nuevo método entre el objeto de la clase IU_CP y el objeto de la clase ControladorLN pulsando en el icono etiquetado con “Create Operation”. Dar el nombre buscarPelículas.

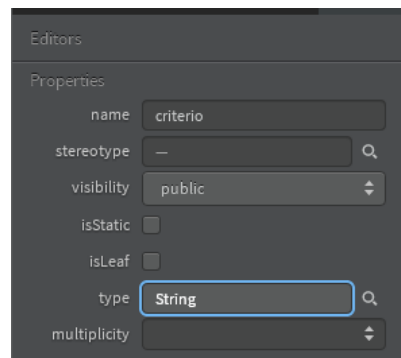


2.3 Definir los parámetros de los métodos. Para ello en el menú “Explorer”, posicionarse sobre el método “buscarPelículas” de la clase, hacer click en “Add” => “Parameter” y poner un nombre al parámetro. El tipo de datos se puede poner, seleccionando en “Explorer” el nombre del parámetro y escribiéndolo en el campo “type” del menú “Properties”, dar el nombre del tipo de datos. Añadir los dos parámetros: criterio y valor.

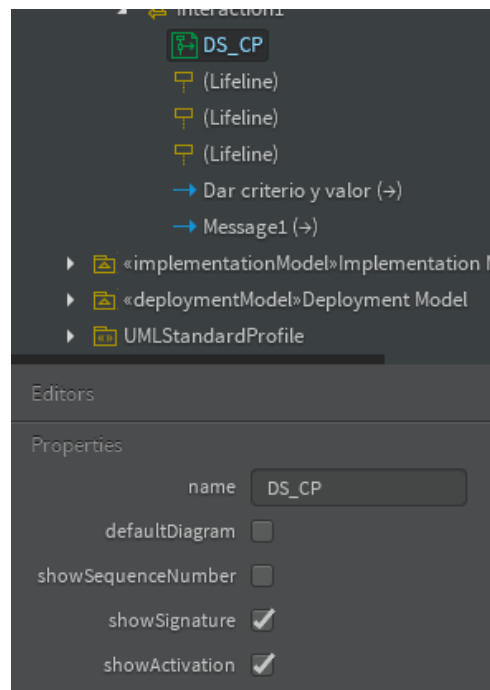
Para añadir un parámetro:



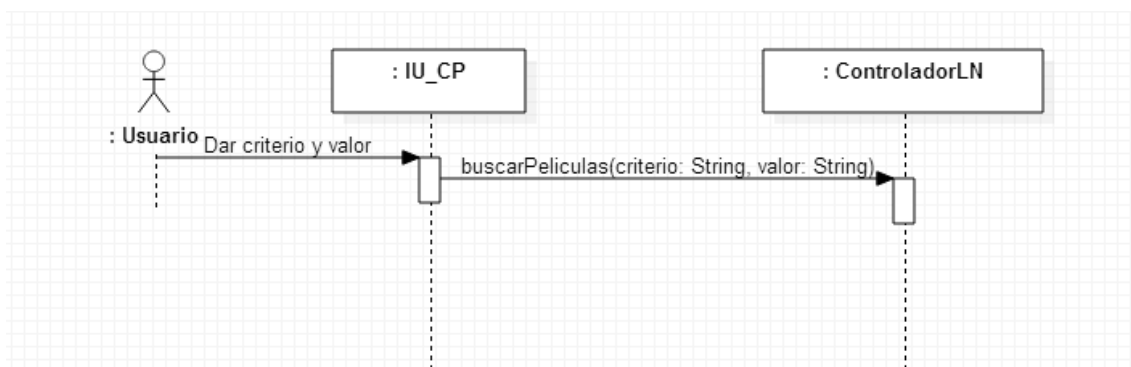
Para añadir el tipo del parámetro:



Para que los parámetros y tipos de los mismos sean visibles y no se muestre el número de secuencia en los mensajes, hay que seleccionar el diagrama de secuencia DS_CP en el "Explorer", para que aparezca su menú de propiedades ("Properties"). Y en el mismo, seleccionar "showSignature" y no seleccionar "showSequenceNumber"

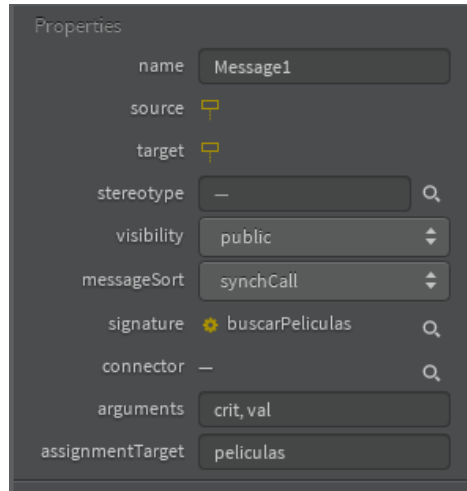


Para que aparezca la llamada al método así, con los parámetros y tipos formales:



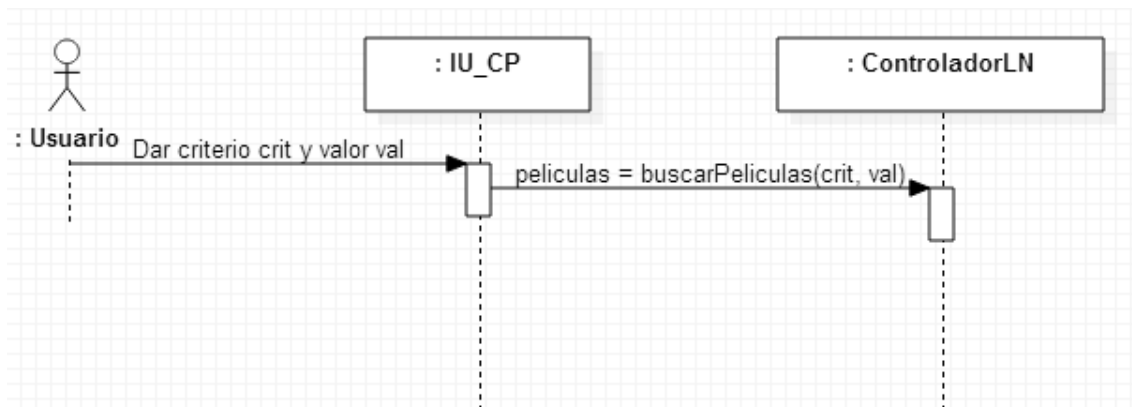
Sin embargo, se puede modificar lo que se muestra en la llamada al método. Se puede añadir una variable donde obtener la respuesta, y que los valores de los parámetros

actuales no coincidan con los parámetros formales. Para ello, se selecciona en el diagrama de secuencia, la llamada a dicho método, y en los campos “Arguments” y “assignmentTarget” del menú “Properties” se modifica lo que se desee.



| Properties | |
|------------------|-----------------|
| name | Message1 |
| source | [lifeline icon] |
| target | [lifeline icon] |
| stereotype | — |
| visibility | public |
| messageSort | synchCall |
| signature | buscarPeliculas |
| connector | — |
| arguments | crit, val |
| assignmentTarget | peliculas |

Que haría que se mostrara lo siguiente en el diagrama de secuencia:

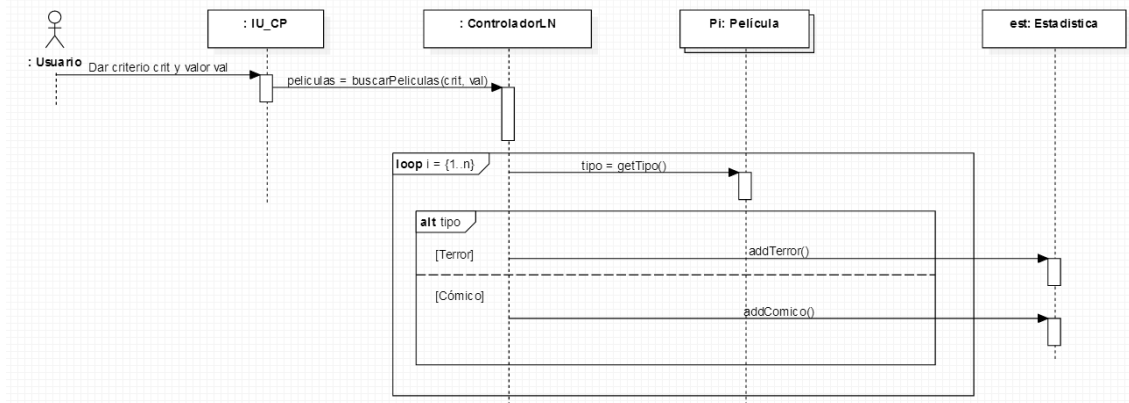


Nota: he modificado también el texto del mensaje entre :Usuario y :IU_CP para mostrar explícitamente en el diagrama que crit es el criterio y val el valor.

A continuación vamos a añadir un fragmento de diagrama de secuencia (que no aparece en el flujo de eventos del caso de uso anterior), con el único objetivo de utilizar “bucles” y “condiciones”. Podría corresponder al siguiente fragmento de flujo de eventos:

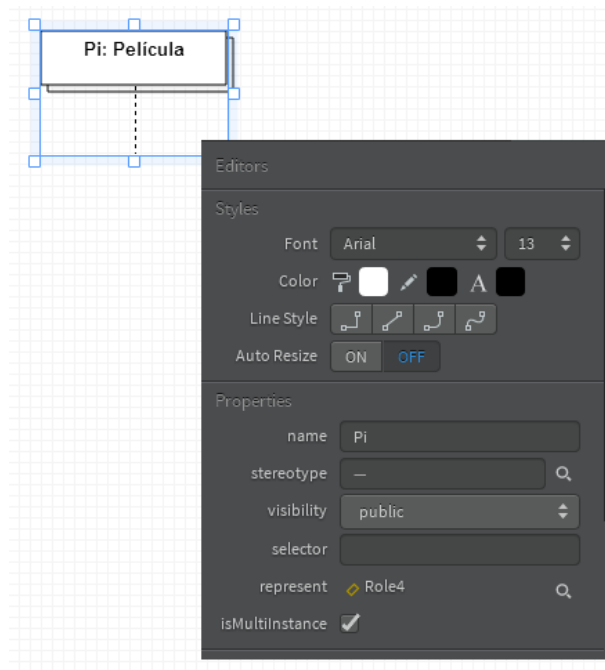
- Se van recogiendo estadísticas del número de películas de género de terror y cómico.

Lo vamos a diseñar así: se recorren las películas y dependiendo del tipo de película (de terror o cómica) se va modificando un objeto de una clase que va almacenando las estadísticas. No se considera necesario dar persistencia a dicha clase. Simplemente sería una estructura de datos en memoria principal que la lógica del negocio podría utilizar en otros casos de uso. Así quedaría el fragmento a añadir:

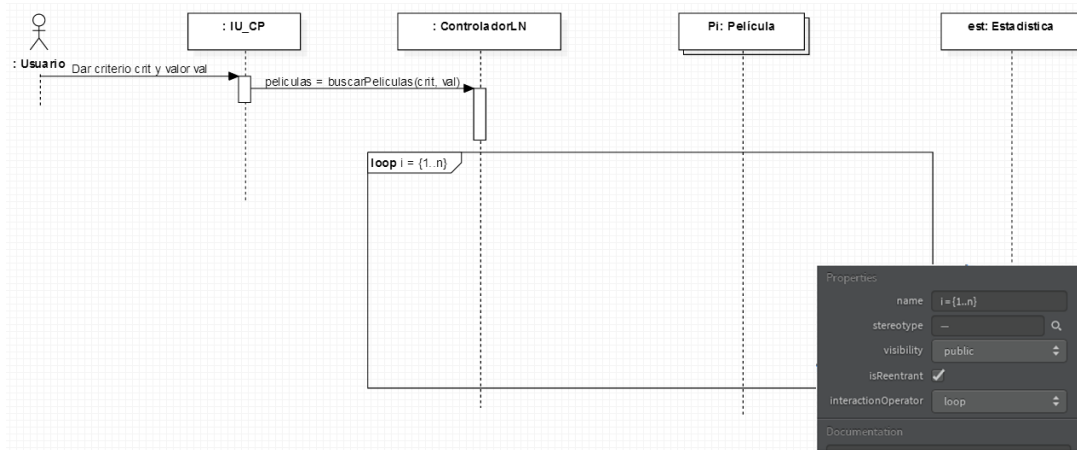


Pasos para hacerlo:

- Añadir un objeto de nombre Pi de la clase Película y otro objeto de nombre est de la clase Estadística (que no existe en el modelo del dominio y se crea arrastrándolo un “Lifeline” del “Toolbox” “Interactions (Basic)”). Para indicar que Pi son varios objetos de Película, hay que seleccionarlo el objeto Pi y en el menú Properties activar “IsMultiInstance”

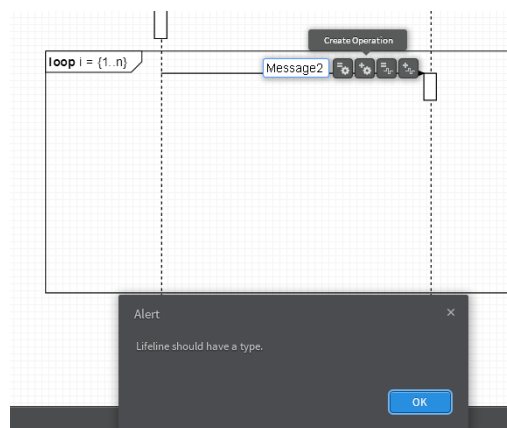


- 2) Añadir un bloque de repetición (CombinedFragment del “Toolbox” “Interactions (Advanced)”) de tipo “loop” (para indicar una repetición). El tipo “loop” se selecciona en el atributo interactionOperator, y la condición del bucle en el atributo “name” de “Properties”.

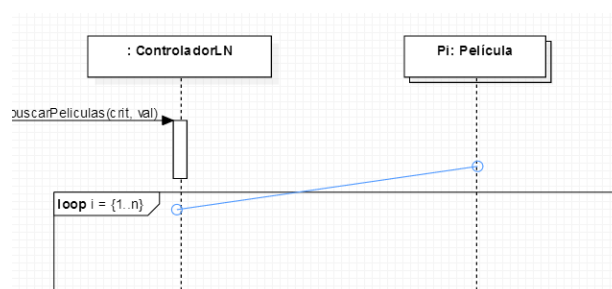


- 3) Añadir la llamada desde el objeto :ControladorLN a un nuevo método “getTipo()” de Pelicula (añadiendo un mensaje Message del Toolbox Interactions (Basic), entre ambos objetos)

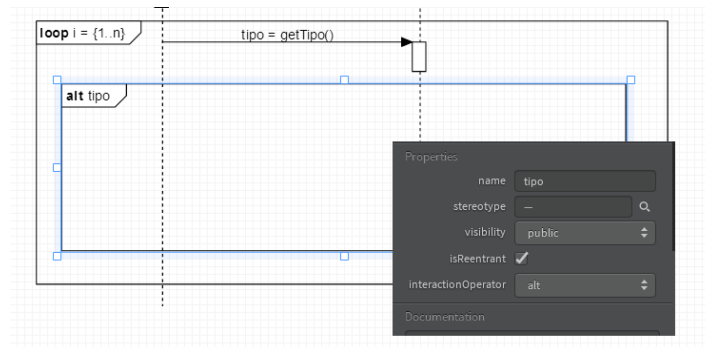
Nota: puede dar un problema como el siguiente si la herramienta StartUML no detecta bien el objeto al que se le envía el método o mensaje:



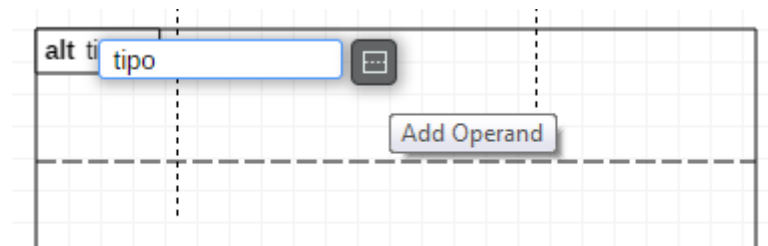
Para ello, intentar poner el destino del mensaje fuera del “loop” y asegurarnos de que el objeto al que se le envía el mensaje es una clase ya definida en alguno de los modelos (del dominio, de clases de diseño,...) porque si no, no podrá añadir un método a una clase inexistente:



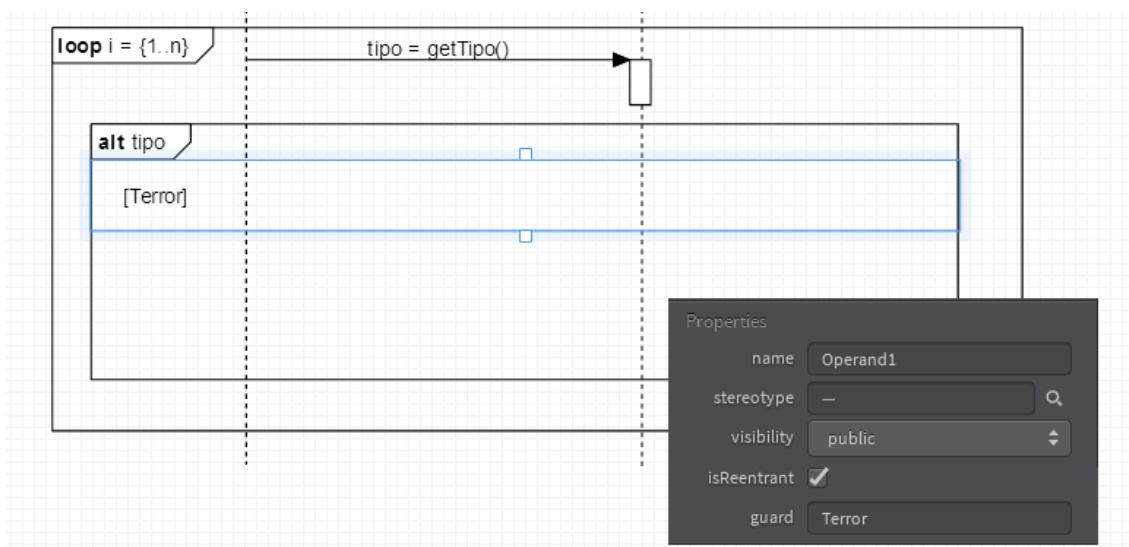
- 4) Añadir un bloque condicional (CombinedFragment del “Toolbox” “Interactions (Advanced)”) de tipo “alt” (para indicar una condición). El tipo “alt” se selecciona en el atributo InteractionOperator, y la condición “tipo” en el atributo “name” de “Properties”. Para indicar qué es “tipo” lo hemos puesto como la respuesta que devuelve el método `getTipo()` (poniendo “tipo” en la propiedad “assignmentTarget” correspondiente al mensaje “`getTipo()`”)



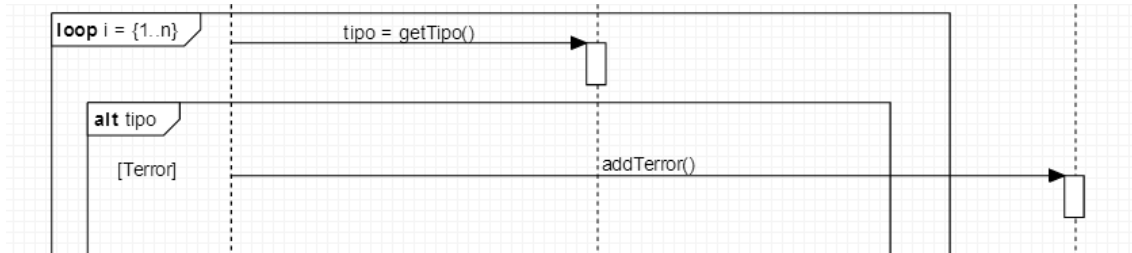
- 5) Añadir dentro del bloque alt dos bloques del tipo “Interaction Use” para indicar cada una de las alternativas (etiquetadas como Terror o Cómico en el atributo Guard). El bloque “alt” ya viene con una condición (alternativa). Para añadir la segunda, hay que hacer click en el icono “Add Operand” (el cual aparece cuando se cambia el nombre al bloque “alt” o al hacer doble click izquierdo dentro del bloque alt).



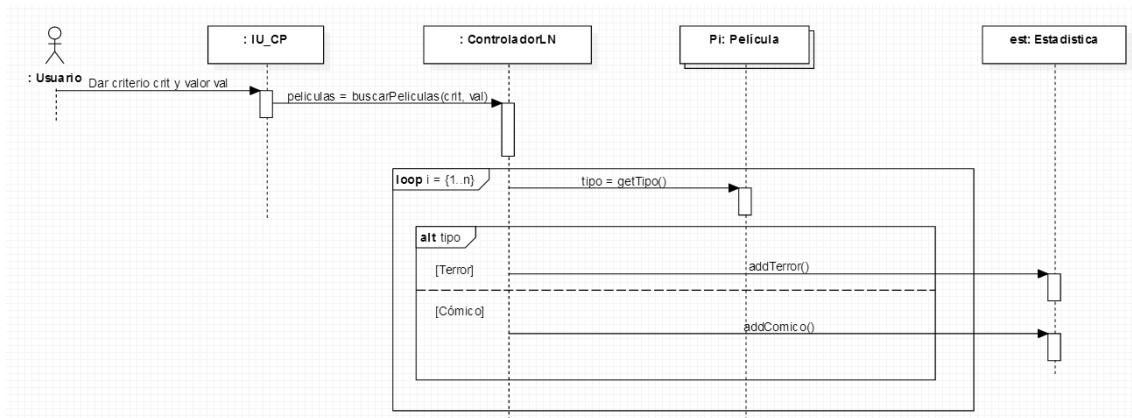
Las condiciones “Terror” (y “Cómico”) se ponen respectivamente en la propiedad “guard” que aparece al seleccionar cada uno de los bloques.



Y a continuación se puede añadir el mensaje “addTerror” a ejecutar cuando tipo=“Terror”



Finalmente, tras añadir la última llamada a un nuevo método addComico, se obtendría el diagrama de secuencia deseado:



3. Generación de código

Para poder generar código de manera automática hay que tener instalada la extensión “Java code generation and reverse engineering”. Se puede comprobar/instalar en Tools => Extension Manager

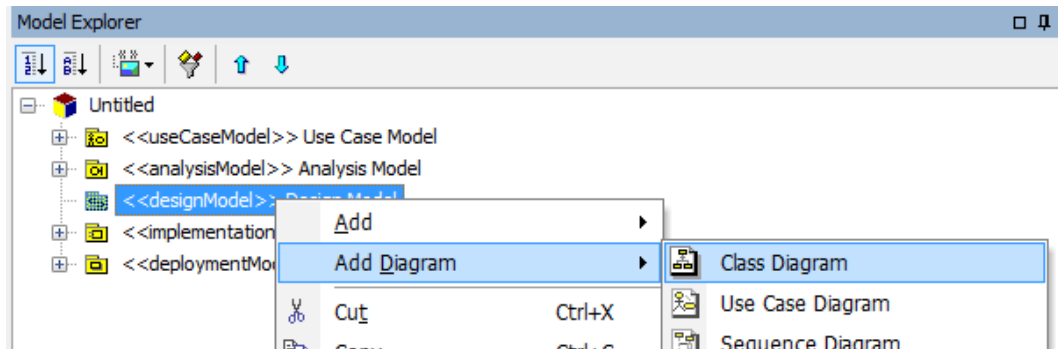
Una vez instalada, la herramienta StarUML nos permite generar algo de código a partir de las clases definidas. Dicho código consiste básicamente en los esqueletos de las clases Java con los nombres de los atributos y las signatures de los métodos. No genera el código correspondiente a los diagramas de secuencia.

Para ello:

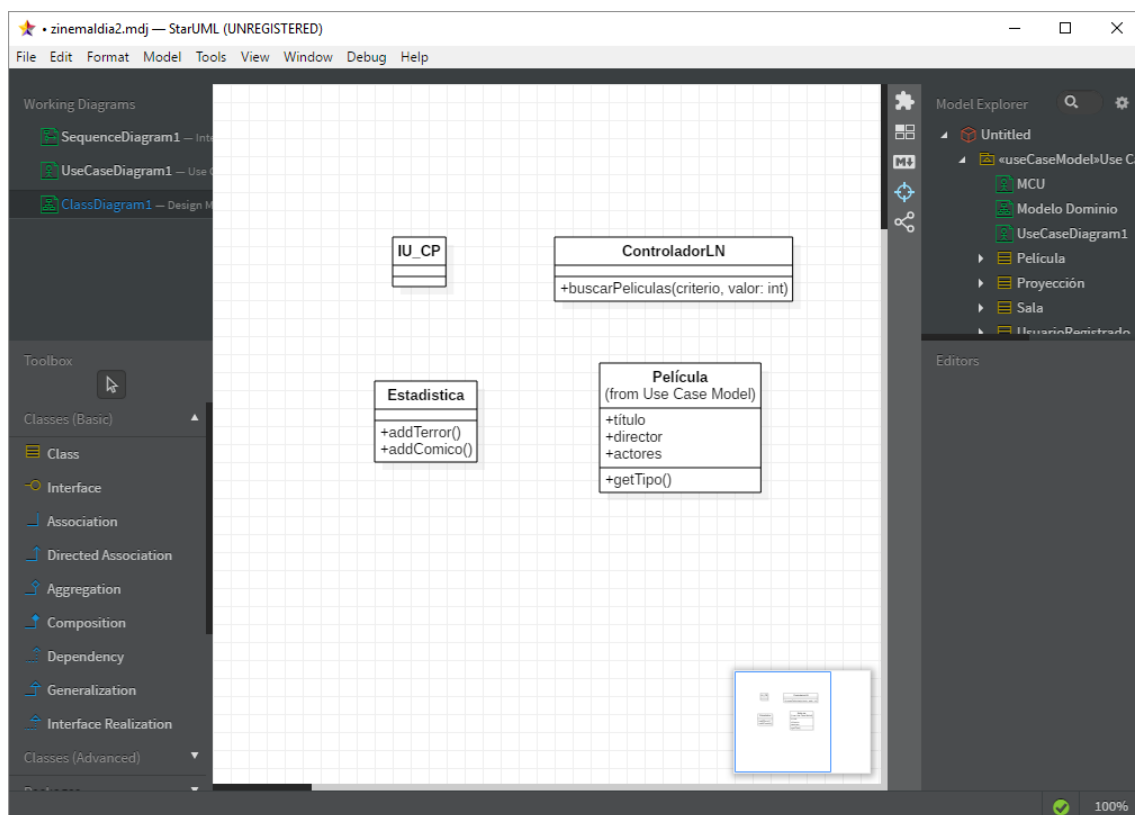
Tools => Java => Generate Code => Escoger por ejemplo el modelo del diseño => Seleccionar una carpeta donde se almacenarán las clase Java

4. Definir el modelo del diseño

Las clases del diseño se pueden mostrar juntas en un diagrama de diseño. Para ello, crearemos un nuevo diagrama de clases en el apartado <<designModel>> del “Model explorer”



Sobre dicho diagrama arrastramos las clases creadas y/o modificadas durante la realización del diagrama de secuencia y se obtendría el siguiente diagrama:



Sobre el diagrama se pueden modificar o añadir nuevas clases, atributos y asociaciones.