Artefactos de diseño en UML usando StarUML

Introducción

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

Objetivos

El objetivo de este laboratorio es el siguiente:

- Utilizar la herramienta StarUML para realizar diagramas de secuencia.
- Creación de diagrama de clases de diseño.

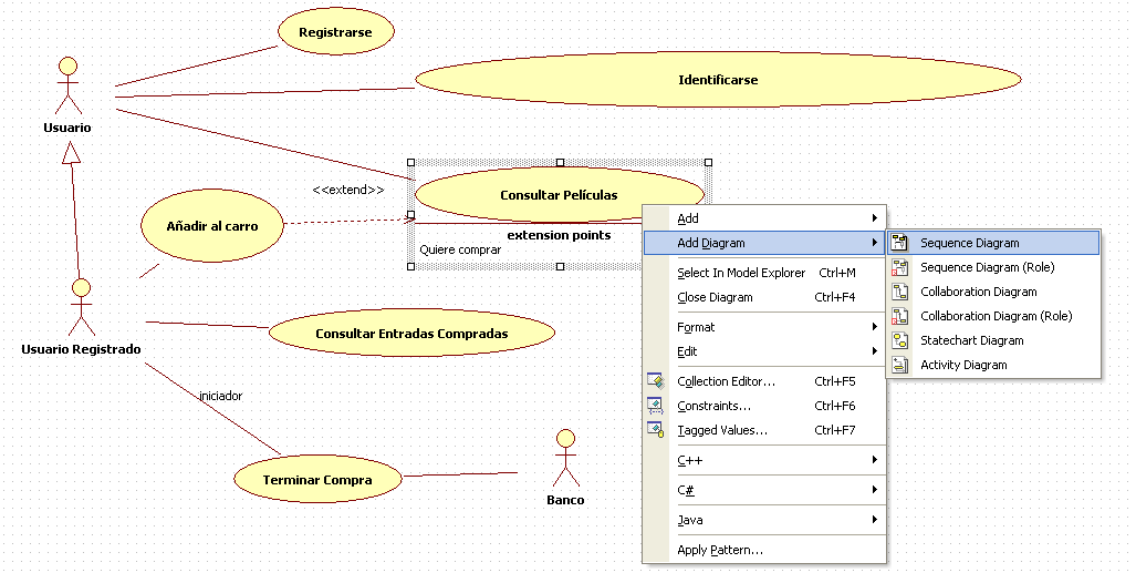
Pasos a seguir:

1. Instalación y ejecución de StarUML.

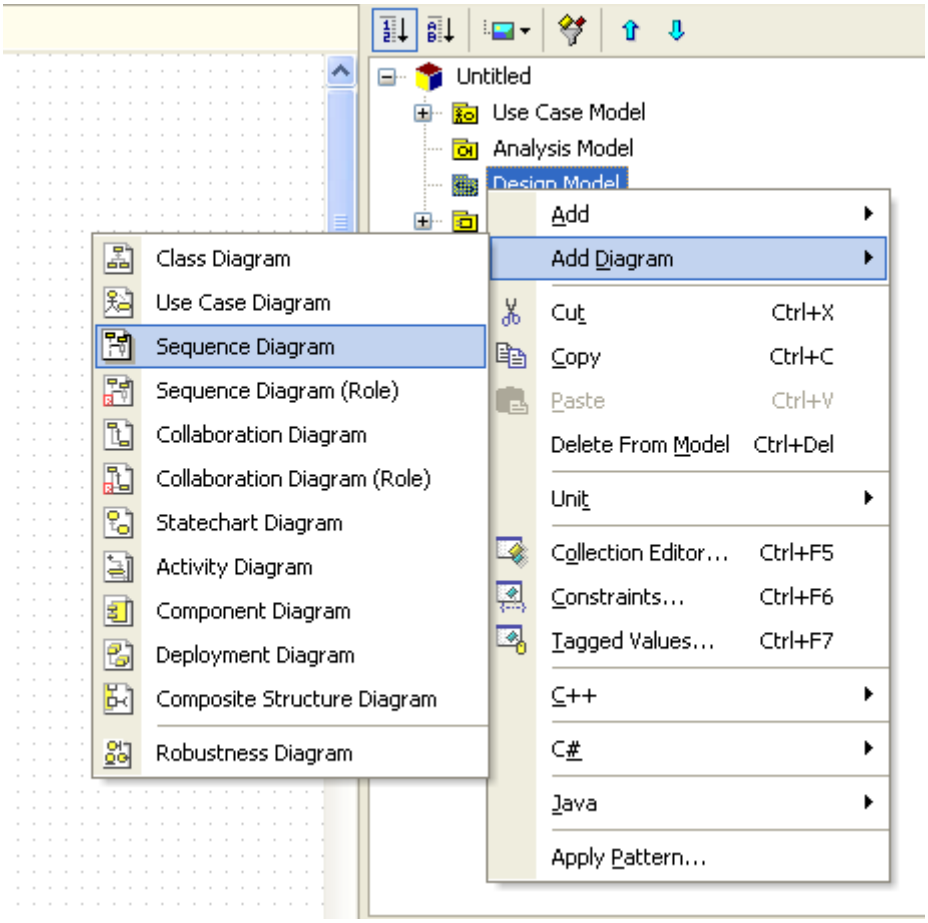
En la siguiente dirección podéis descargar la aplicación: http://staruml.sourceforge.net/en/download.php, si es que no está ya instalada.

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

Abrir en UML el fichero zinemaldia2.uml que tenéis disponible en el Moodle, y haced click derecho en el caso de uso “Consultar Películas”, para crear su diagrama de secuencia (dándole el nombre que se desee, por ejemplo DS_CP),



o bien, haced click derecho en “Design Model” del “Model Explorer” para crear un nuevo diagrama de secuancia, que se añadirá al “Design Model” de la aplicación.,



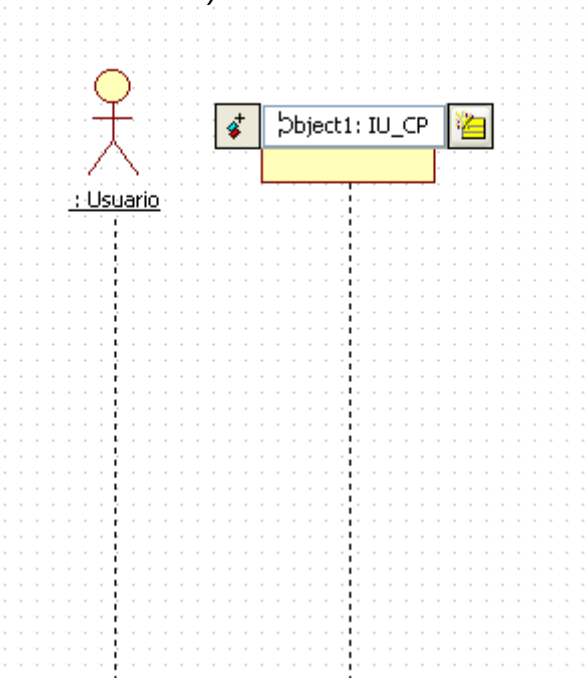
Crear los métodos y las clases:

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:

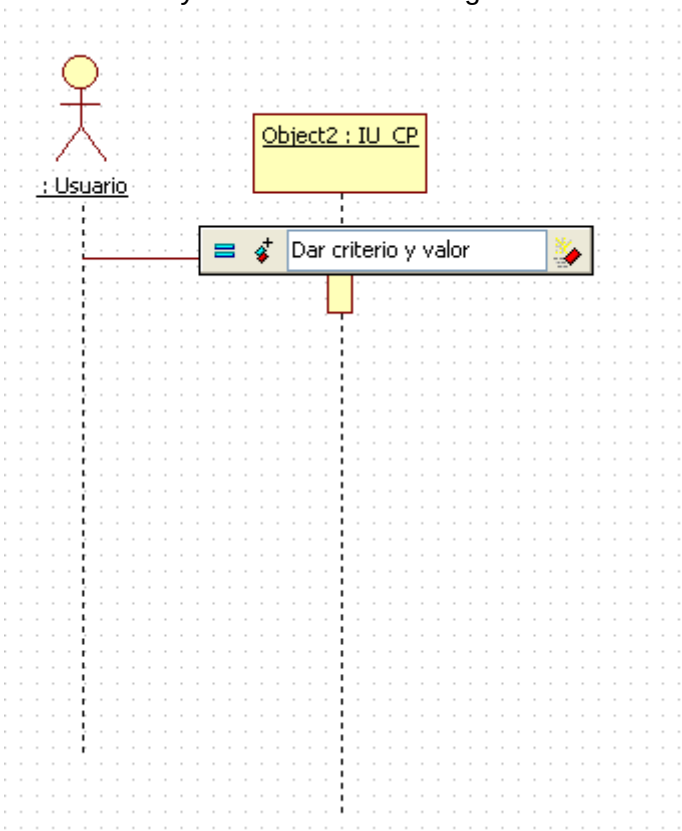
- El usuario proporciona el criterio por el que quiere buscar películas (día, ciclo, título, director o sala), así como el día, o ciclo, o parte del título, o nombre de director o nombre de sala.
- El sistema muestra una lista de películas (el nombre de cada película) que cumplen el criterio de búsqueda anterior.
- El usuario selecciona una película de la lista, y el sistema muestra la siguiente información de la misma: título, nombre del director, actores y una lista de las horas y las salas en las que se proyecta esa película en el día actual.
- Si el usuario quiere comprar entradas para esa proyección: extends AÑADIR AL CARRO

1) Arrastrar el actor ya existente “Usuario” desde el “Model Explorer” hasta el diagrama de secuencia.

2) Añadir un objeto de una nueva clase (arrastrando un “Object” del toolbox “Sequence”) y dadle 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 “Add Class”).

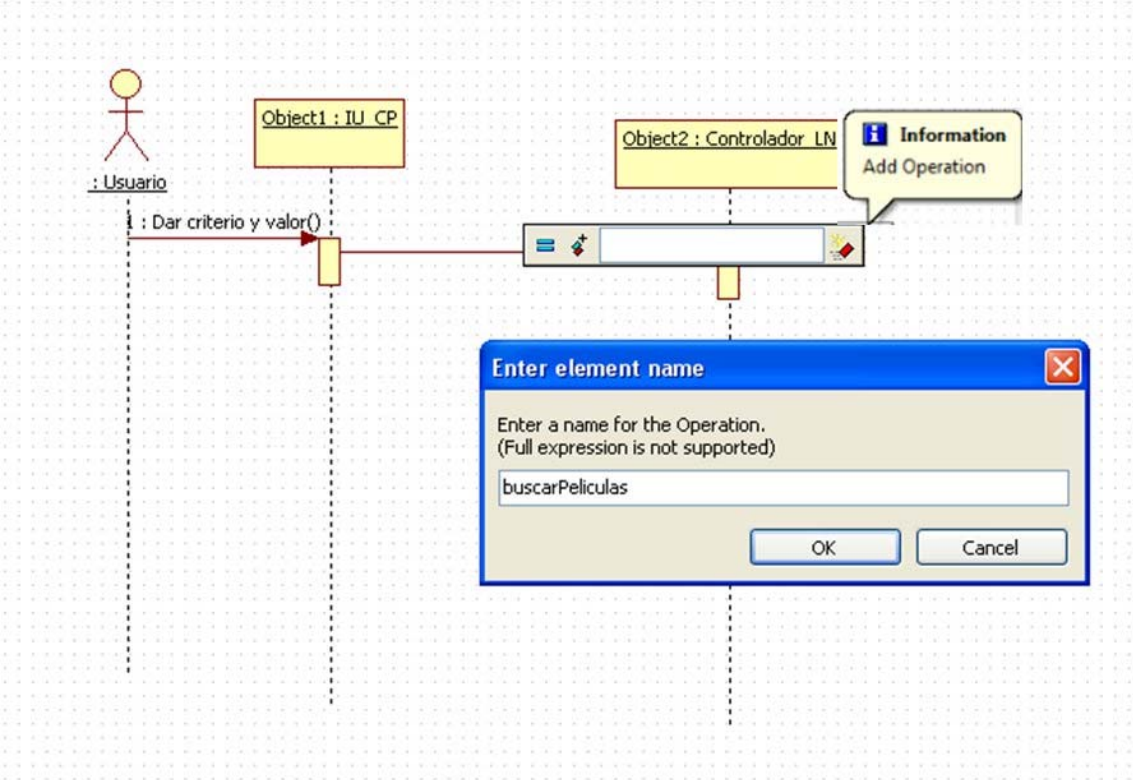


3) Crear una interacción entre el actor y la interfaz añadiendo un mensaje (stimulus) entre ambos y anotándolo con el siguiente texto:



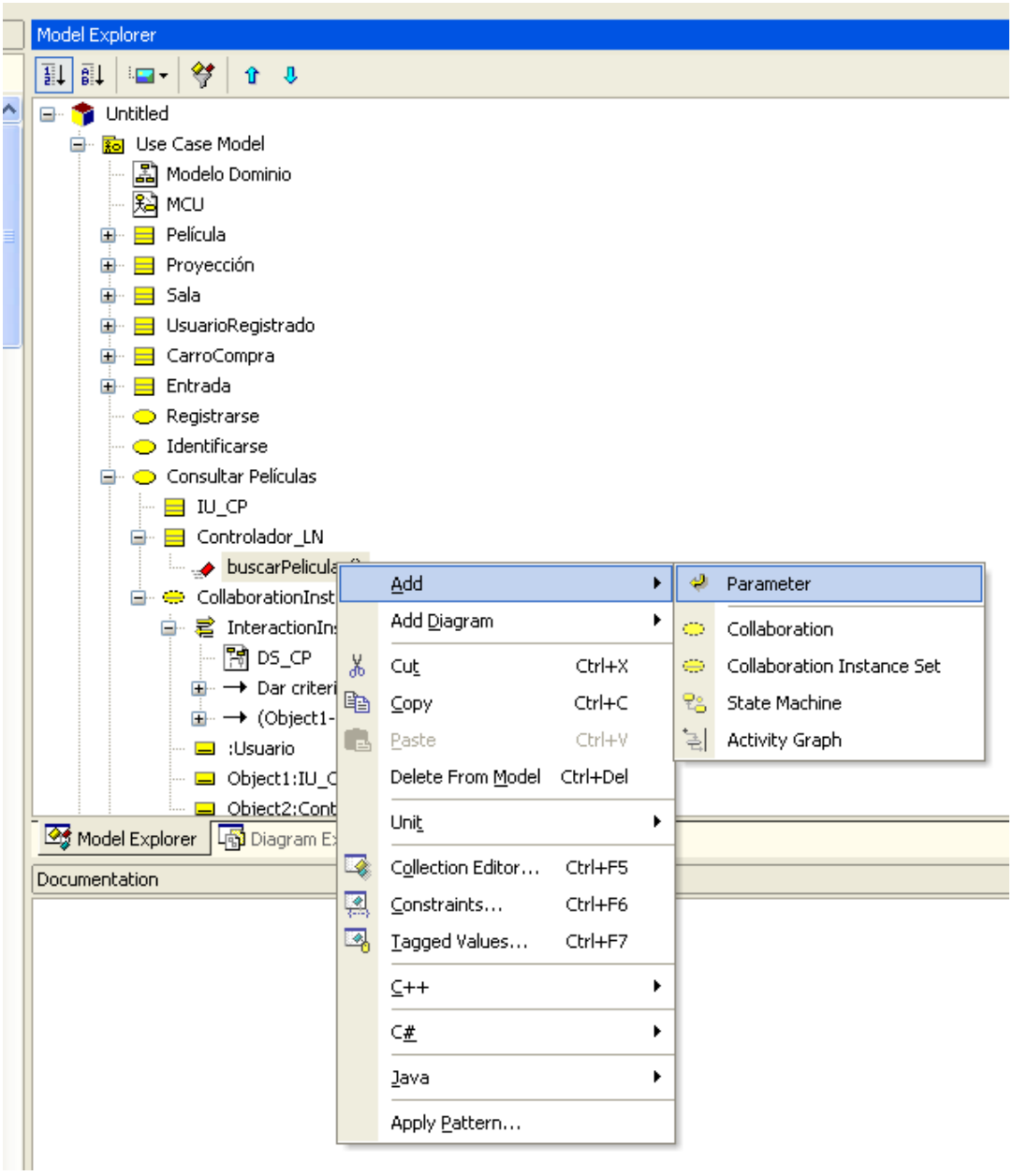
2.2 Definir los métodos. Crear un objeto de una nueva clase Controlador_LN (arrastrándolo desde “Object” de “Sequence” en “Toolbox”) y un nuevo método

entre el objeto de la clase IU_CP y el objeto de la clase Controlador_LN pulsando en el icono etiquetado con “Add Operation” que aparece a la derecha del objeto de Controlador_LN, equitado con el nombre buscarPeliculas.

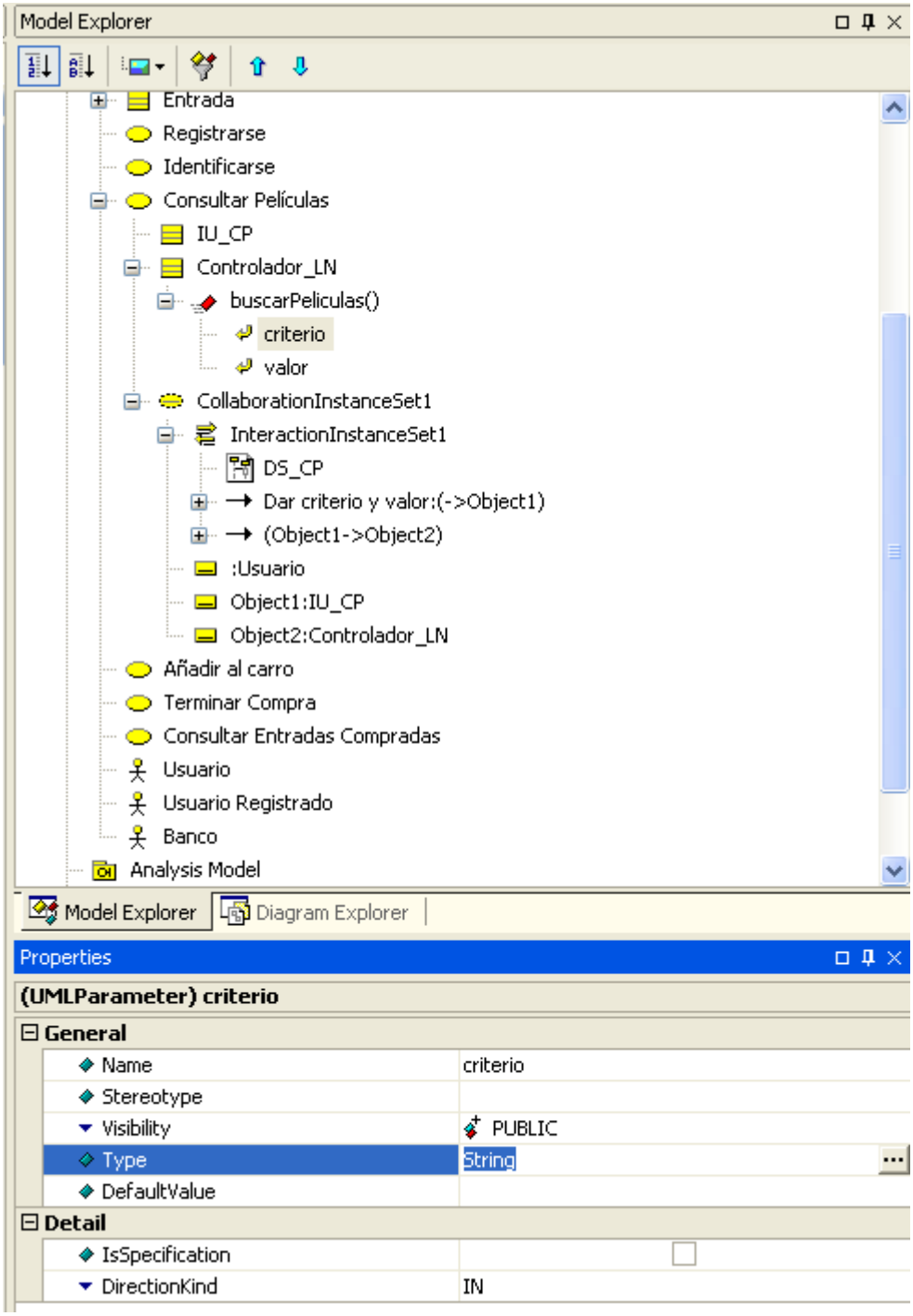


2.3 Definir los parámetros de los métodos. Para ello en el menú “Model Explorer”, posicionarse sobre el método de la clase, hacer click en “Add Parameter” y poner un nombre al parámetro. El tipo de datos se puede poner, seleccionando en “Model Explorer” el nombre del parámetro y escribiéndolo en el campo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

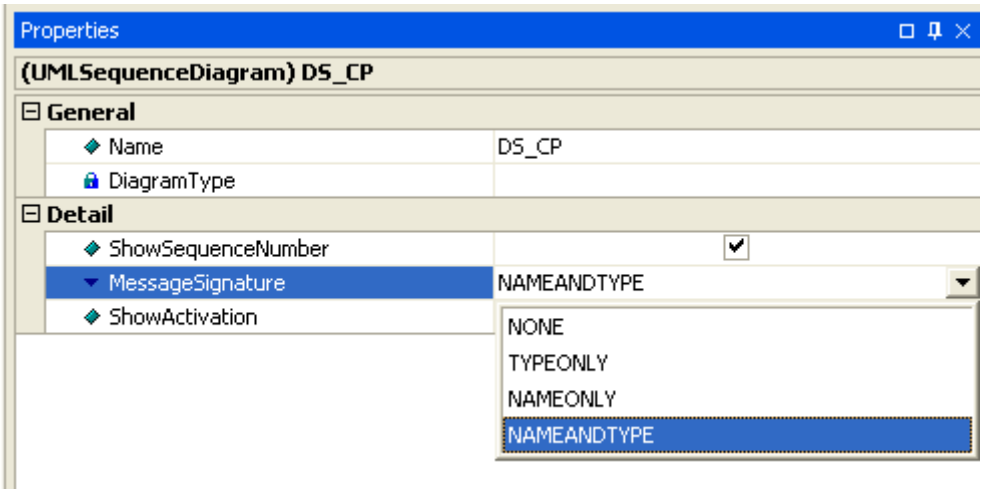
Para añadir un parámetro:



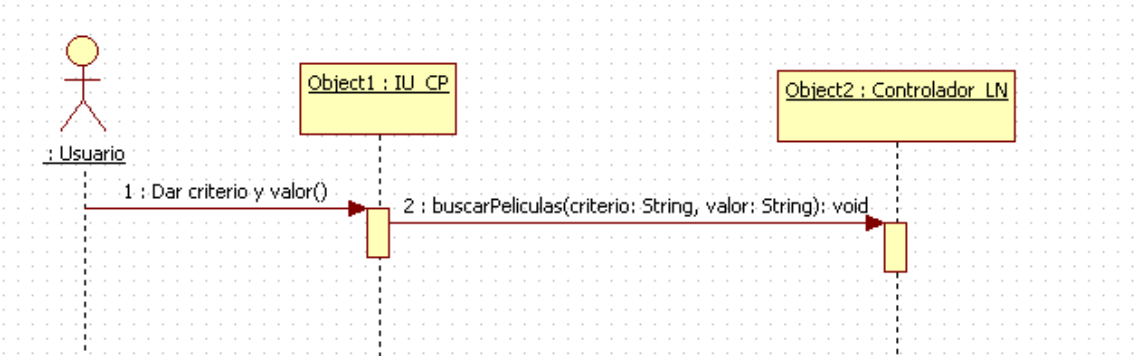
Para añadir el tipo del parámetro:



Para que los parámetros y tipos de los mismos sean visibles, hay que seleccionar el diagrama de secuencia DS_CP en el “Model Explorer”, para que aparezca su menú de propiedades (“Properties”). Y en el mismo, seleccionar “NAMEANDTYPE” de “MessageSignature”

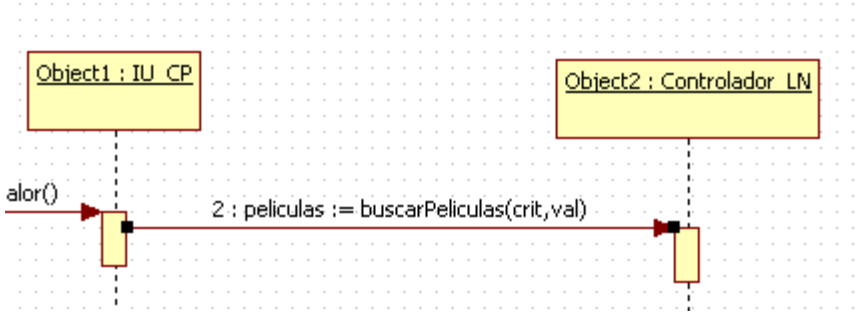


Para que aparezca la llamada al método así, con los parámetros y tipos formales:

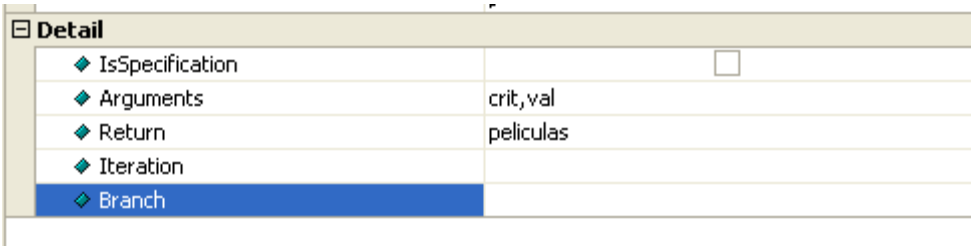


Sin embargo, se puede modificar lo que se muestra en esa llamada para añadir una variable donde obtener la respuesta, y los valores de los parámetros actuales, que no tienen por qué coincidir con los parámetros formales. Para ello, se selecciona el diagrama de secuencia en el “Model Explorer” y en los campos “Arguments” y “Return” del menú “Properties” se puede añadir lo que se desee.

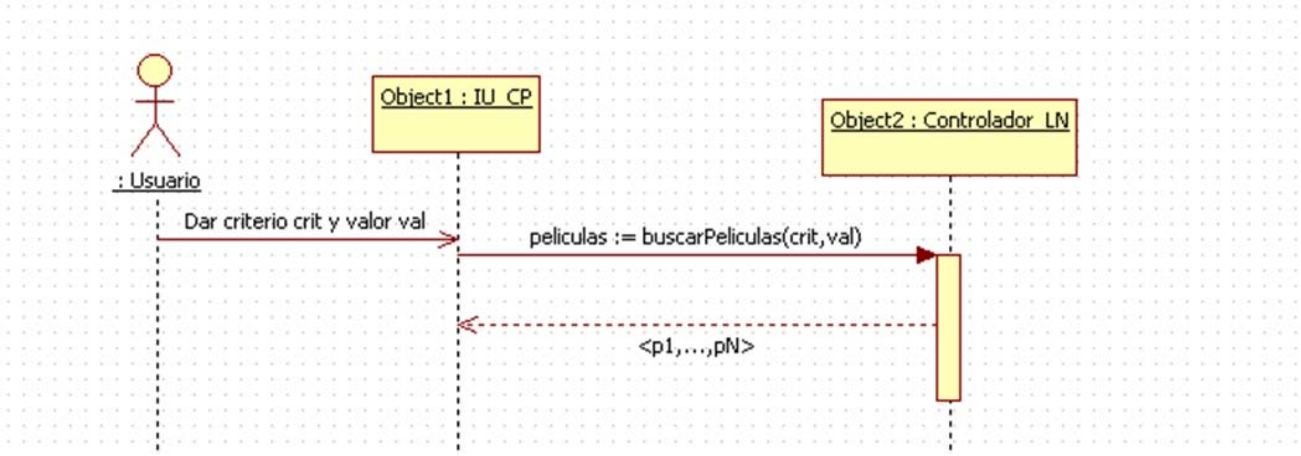
Esto es lo que se mostraría en el diagrama de secuencia:



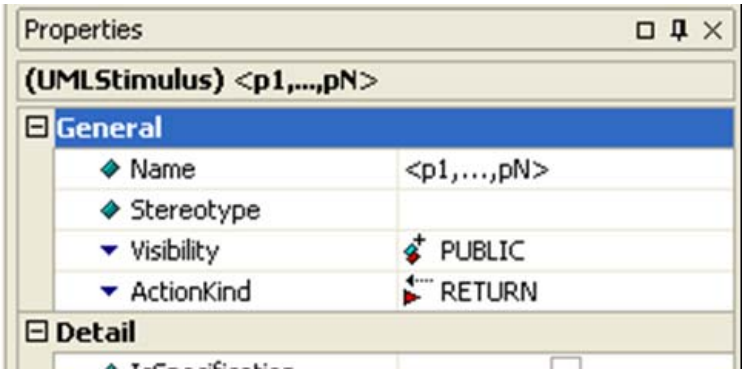
Si hubiéramos dados los siguientes valores (que no tienen sentido en este caso)



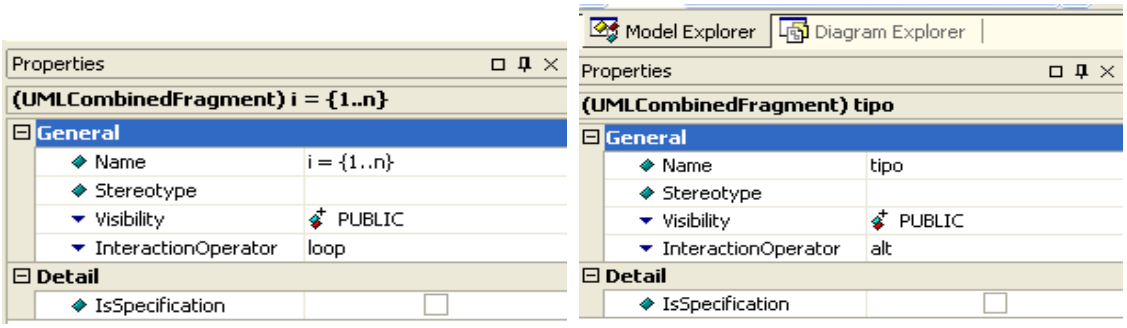
Aunque en ese caso, sería mejor que la interacción entre el actor y el objeto de IU_CP dijera que los parámetros crit y val son los valores criterio y valor dados por el usuario. Además es mejor quitar los paréntesis, ya que no es una llamada a un método. Eso se consigue seleccionando el "stimulus" y diciendo que es el tipo SEND.



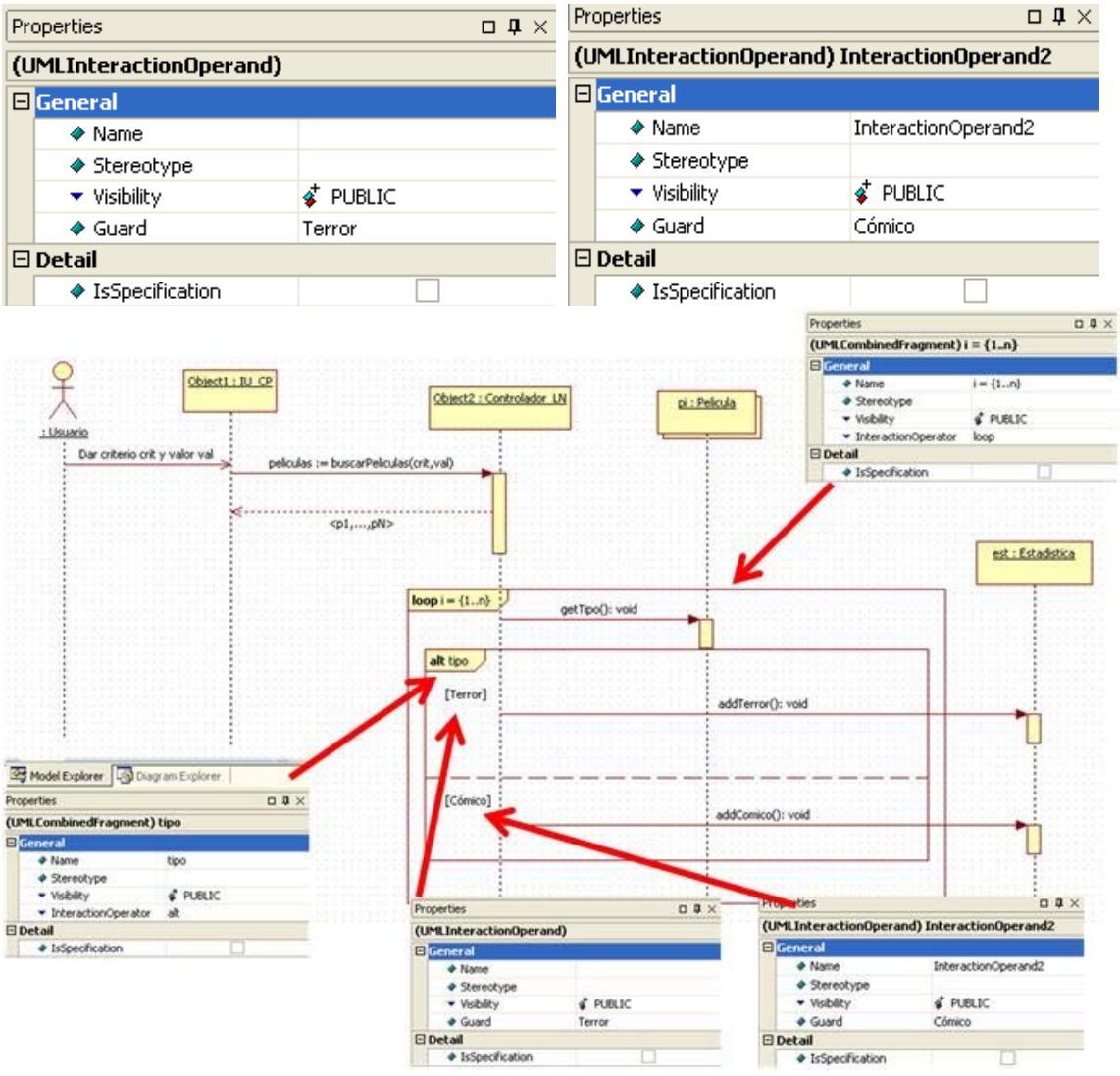
La interacción de tipo resultado es de tipo "RETURN", como puede verse a continuación:



A continuación añadimos un par de bloques (CombinedFragment) de tipo “loop” (para indicar una repetición) y otro de tipo “alt” (para indicar una condicional). El tipo “alt” o “loop” se selecciona en el atributo InteractionOperator, y la condición del bucle o de la condicional en el atributo Name.

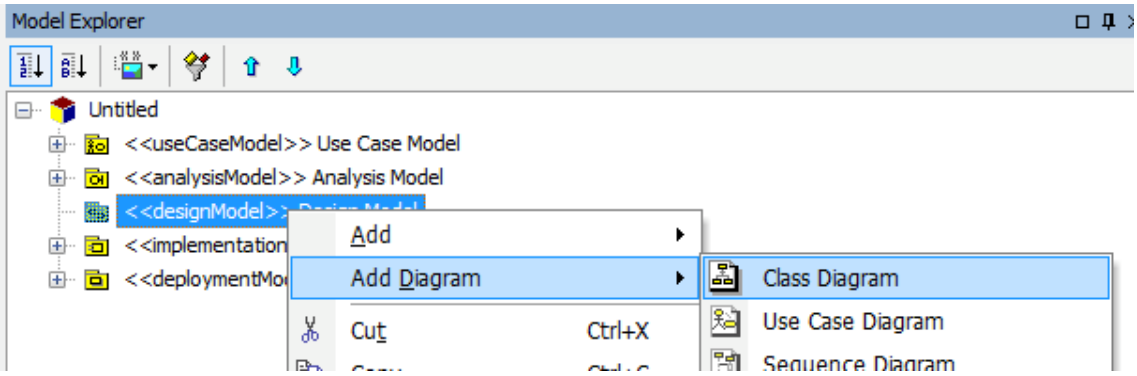


Dentro del bloque alt hemos añadido dos bloques del tipo InteractionOperand para indicar cada una de las alternativas (etiquetadas como Terror o Cómico en el atributo Guard)



3. 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", y arrastraremos las clases que deseemos que aparezcan. Además, se podrán añadir nuevos atributos y asociaciones.



4. Generación de código

La herramienta StarUML nos permite generar algo de código (los nombres de los atributos y las signaturas de los métodos) a partir de las clases definidas. Para ello:

1.- Model => Profiles => "Java Profile" => include, para incluir el perfil de Java, que indica cómo se traducen a Java las clases del diseño.

2.- Tool => Java => Generate code
A continuación hay que seleccionar las clases para las que queremos generar el código (aunque previamente debe indicarse el paquete o modelo en el que se encuentran, de las que deben colgar directamente). Por último se indicará el directorio en el que se generarán las clases Java.