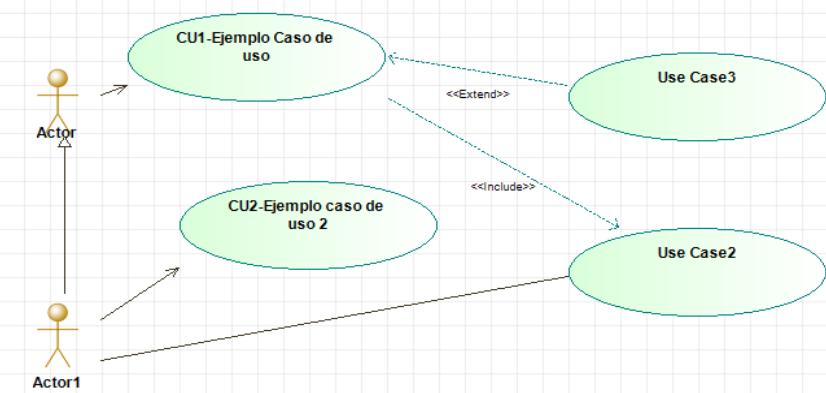


## Diagrama de caso de uso (Use case diagram)

Utilizar: Actor, Use case, Communication link, Inclusion relationship, Extension relationship y Generalization.

Para mostrar la flecha cuando sea necesario, en las relaciones actor-caso de uso: seleccionar la línea de la relación y usar la opción Symbol para marcar Show navigability.

Ejemplo de formato:



## Diagrama de actividad (Activity diagram)

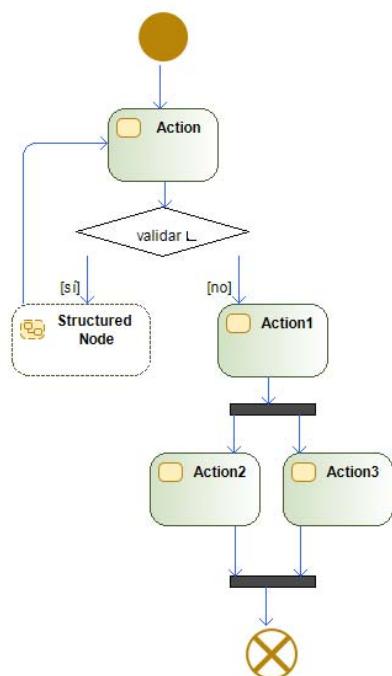
Crearlo desde el caso de uso para que quede vinculado a él.

El concepto 'actividad' visto en teoría se desarrolla en Modelio con Structured node.

Utilizar: Action, Structured node, Initial node, Activity final node, Decision-merge node, Fork/Join y Flow.

Los nodos de decisión deben mostrar la condición y poder adaptar su tamaño (en Configuration /Diagram themes / Modelio 3.x / Decision-merge node: Show label y Representation mode = Structured). También añadir etiquetas (guard) a las líneas de salida de nodos de decisión.

Ejemplo de formato:



## Diagrama de clases de diseño (Class diagram)

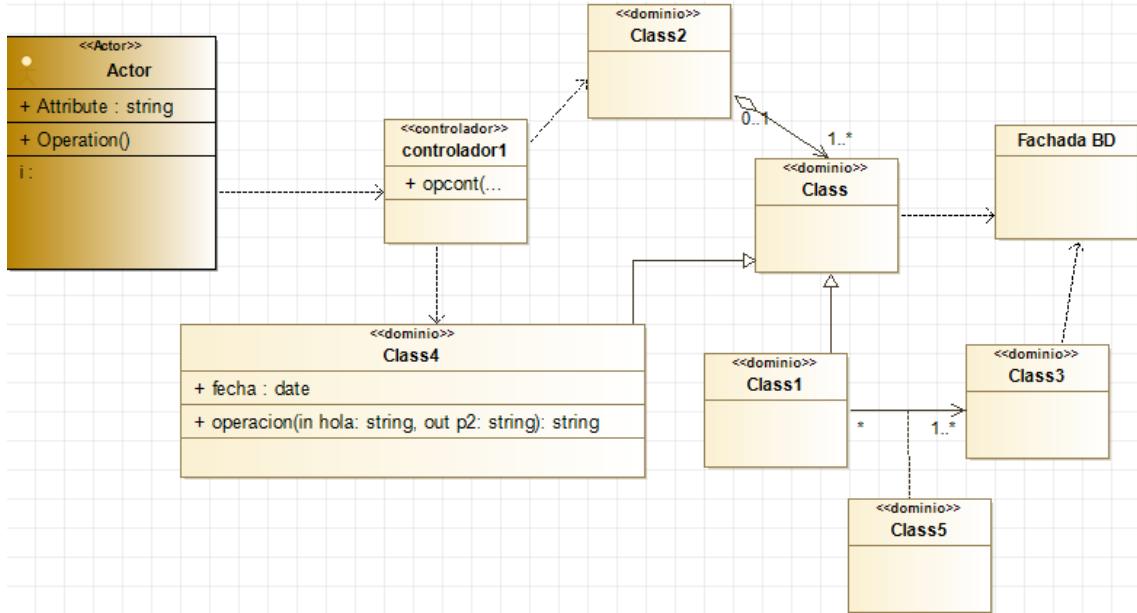
Utilizar: Class, Attribute, Operation, Class association, Association, Aggregation, Composition, Generalization y Usage (en Import Link) para las relaciones de dependencia.

Arrastar los actores del repositorio para las clases de interfaz y enSymbol en Representation model dar el valor Structured.

Añadir los estereotipos dominio y controlador: en Local module / Local profile / Create element / Stereotype.

Las clases deben mostrar el estereotipo dominio y controlador (en Configuration /Diagram themes / Modelio 3.x / Class / Header: Stereotypes mode = text + icon).

Ejemplo de formato:



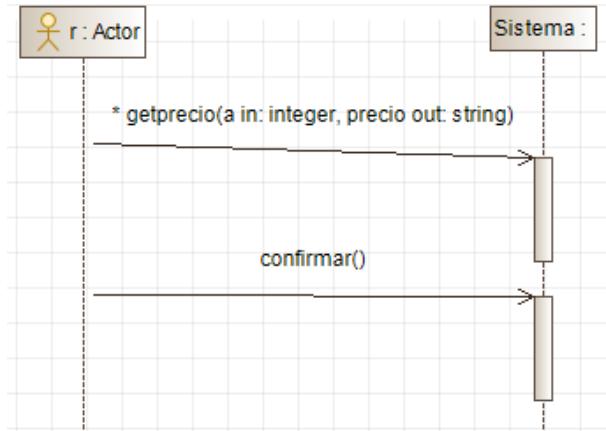
## Diagrama de secuencia del sistema (Sequence diagram)

Crearlo desde el caso de uso para que quede vinculado a él.

Utilizar: Lifeline y Asynchronous message.

Adaptación del formato de los mensajes visto en teoría a Modelio: el mensaje no llevará número de secuencia y no incluirá objeto de retorno, solo parámetros de entrada, de salida y de entrada/salida.

Ejemplo de formato:



## Diagrama de colaboración de diseño (Communication diagram)

Crearlo desde el caso de uso para que quede vinculado a él.

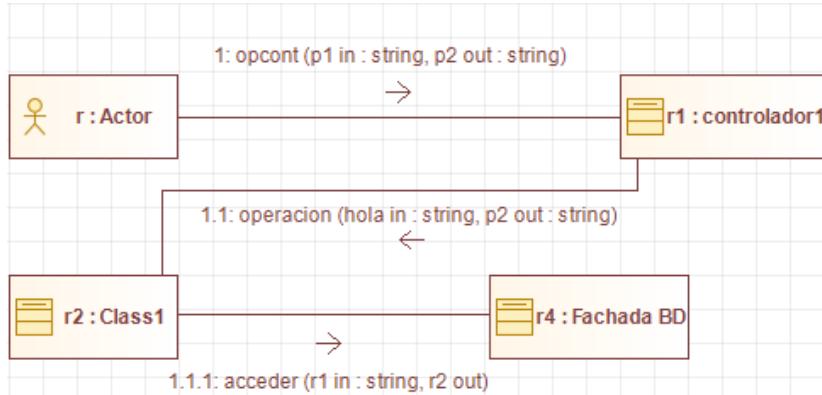
Utilizar: Link y Communication message para los mensajes, los objetos se arrastran del repositorio.

Adaptación del formato de los objetos visto en teoría a Modelio: No es posible utilizar multiobjeto, se usará un objeto simple en su lugar.

Adaptación del formato de los mensajes visto en teoría a Modelio: El formato del mensaje será sin objeto de retorno, solo con parámetros de entrada, de salida y de entrada/salida.

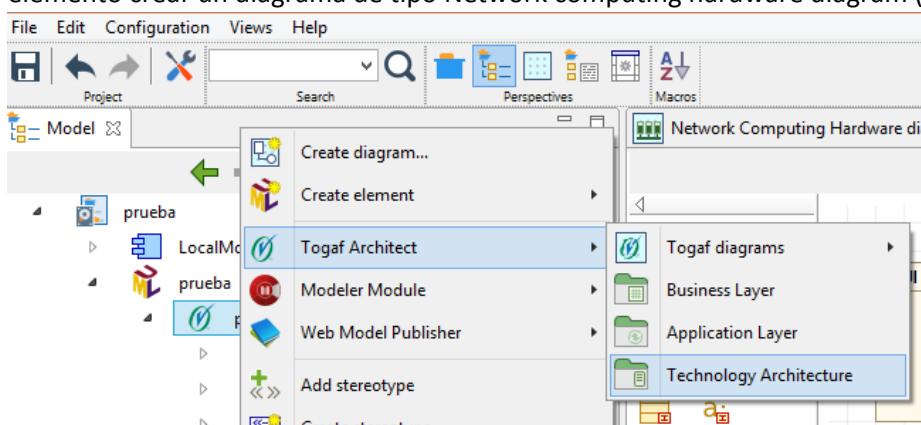
Una vez creado el link sin nombre, se asocia a él un communication message. En el communication message se añade la operación en Invoked (seleccionándola del desplegable que sale con Ctrl+espacio) y después se copia y se pega en Argument, borrando para dejar solo los parámetros. También se añade la numeración en Sequence. Si se quiere enviar mensajes entre dos clases en ambas direcciones hay que crear dos link entre ellas, cada uno con su correspondiente communication message, y, por claridad del dibujo, hacerlos coincidir.

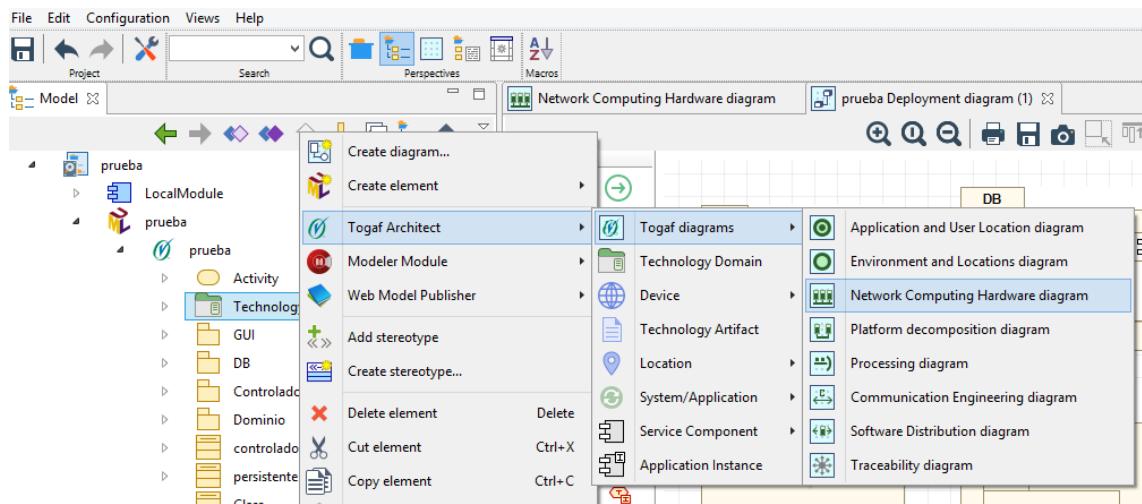
Ejemplo de formato:



## Diagrama de despliegue (Network computing hardware diagram)

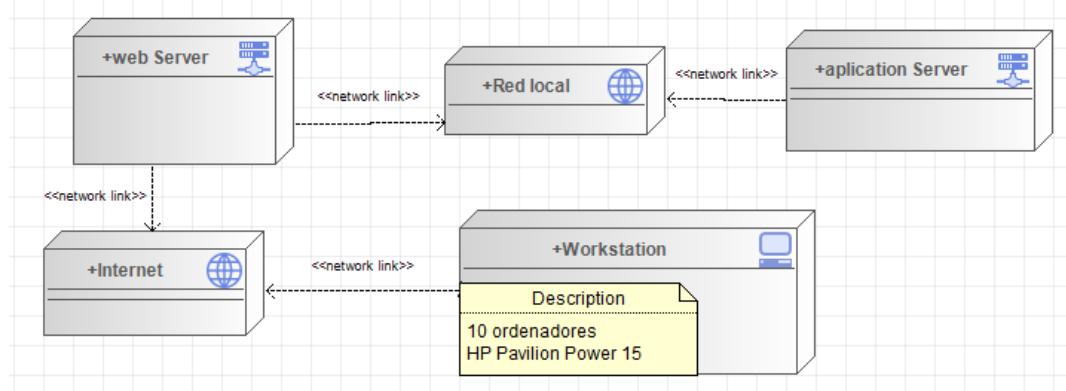
Es necesario instalar en Modelia el módulo TOGAF Architect (disponible en Aula Virtual) mediante Configuration / Modules catalog y después añadirlo al proyecto mediante Configuration / Modules. Despues añadir al proyecto Technology Architecture y desde ese elemento crear un diagrama de tipo Network computing hardware diagram (ver imágenes)





Utilizar: conexiones Network link y nodos necesarios (Network Node). Es posible modificar el nombre de los nodos, por ejemplo tener un nodo Internet y otro Red local, este último se ha creado como nodo de Internet y se le ha cambiado el nombre.

Ejemplo de formato:



### Diagrama de componentes (Deployment diagram)

Utilizar: paquetes que representen agrupaciones de clases o bases de datos, Public Component como fachada, artefactos para otros elementos como ficheros y conexiones de tipo use. Es posible mostrar dentro de un paquete, otros paquetes y su fachada.

Ejemplo de formato:

