



# **Análisis y Diseño de Sistemas II - Laboratorio Computación e Informática**

---



# ÍNDICE

Presentación	5
Red de contenidos	6

## **UNIDAD 1: Análisis Orientado a Objetos**

TEMA 1 : Arquitectura de Análisis	8
TEMA 2 : Análisis de Casos de Uso: realización de análisis de un caso de uso de mantenimiento	41
TEMA 3 : Análisis de Casos de Uso: realización de análisis de un caso de uso buscar	58
TEMA 4 : Análisis de Casos de Uso: realización de análisis de un caso de uso generar cita	60
TEMA 5 : Casos prácticos	64

## **UNIDAD 2: Modelo de Datos**

TEMA 1 : Modelo Conceptual	68
TEMA 2 : Modelo Lógico	76
TEMA 3 : Modelo Físico	84

## **UNIDAD 3: Diseño Orientado a Objetos**

TEMA 1 : Capas lógicas de la arquitectura: definición de capas, subsistemas y componentes	95
TEMA 2 : Diseño de Casos de Uso: realización de diseño de un caso de uso de mantenimiento con patrón arquitectónico MVC	113
TEMA 3 : Diseño de Casos de Uso: realización de diseño de un caso de uso de mantenimiento con patrón arquitectónico MVC y patrón de diseño DAO	121

TEMA 4 : Diseño de Casos de Uso: realización de diseño del caso de uso generar cita	134
TEMA 5 : Modelo de Diseño Componente y despliegue	139

## **ANEXOS**

ANEXO 1 : Cambio de <i>workspace</i>	145
ANEXO 2 : Importación de proyectos	148
ANEXO 3 : Publicación de modelos	151
ANEXO 2 : Ingeniería reversa de código fuente a clases de diseño	153
GLOSARIO	157

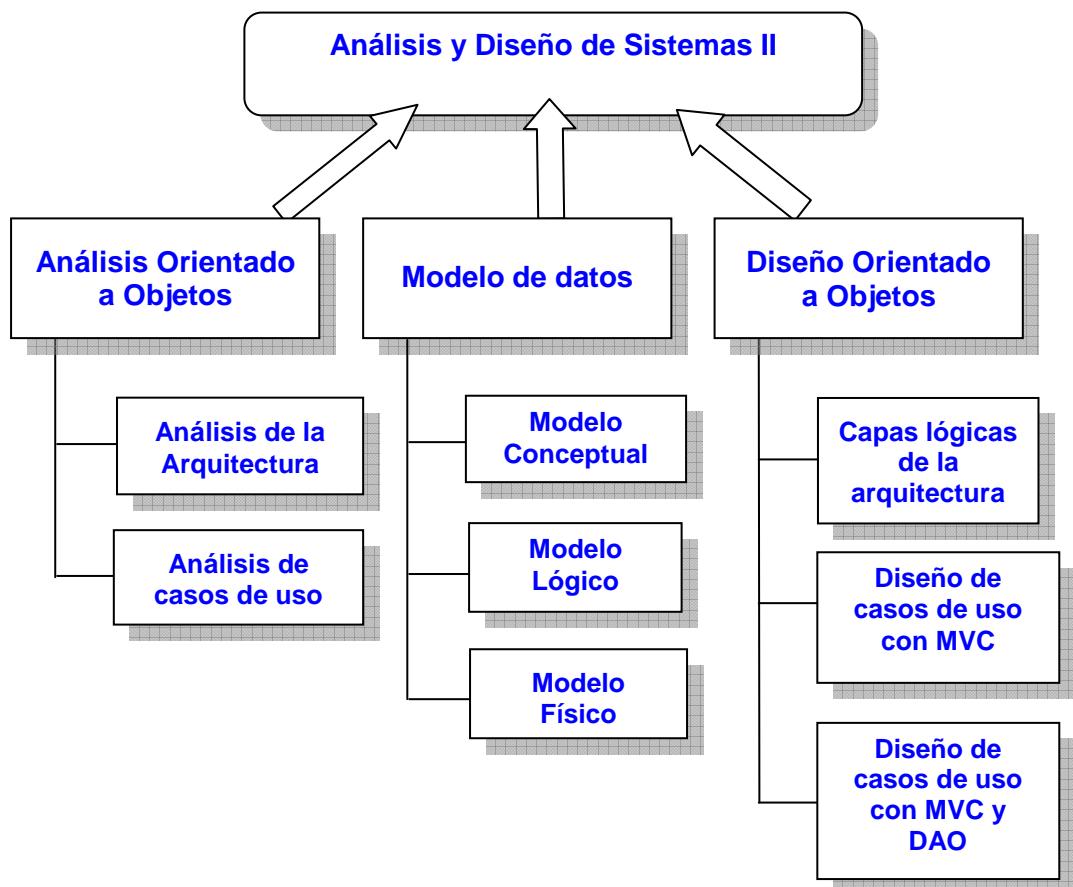
# PRESENTACIÓN

**Análisis y Diseño de Sistemas II** pertenece a la línea formativa y se dicta en la carrera de Computación e Informática. El curso imparte conocimientos relacionados con la disciplina de análisis y diseño, y el modelo de datos. Las herramientas que se utilizarán en las sesiones de laboratorio son las siguientes: IBM Rational Software Architect (RSA) e InfoSphere Data Architect.

El manual del curso ha sido diseñado bajo la modalidad de unidades de aprendizaje, las que se desarrollan durante semanas determinadas. En cada una de ellas, hallará los logros, que debe alcanzar al final de la unidad; además, el tema tratado, el cual será ampliamente desarrollado; y los contenidos que debe desarrollar, es decir, los subtemas. Por último, encontrará las actividades que deberá desarrollar en cada sesión, que le permitirán reforzar lo aprendido en la clase.

El curso es teórico - práctico: consiste en un taller de desarrollo de proyectos de software. En primer lugar, se describe el flujo de trabajo del análisis orientado a objetos. A continuación, se explica el modelo de datos. Por último, se presenta el flujo de trabajo del diseño orientado a objetos.

# RED DE CONTENIDOS





# ANÁLISIS ORIENTADO A OBJETOS

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la primera unidad, el alumno modula la arquitectura de análisis que da soporte a los procesos del negocio, diagrama la estructura y el comportamiento de sus funcionalidades mediante diagramas de clases y diagramas de comunicación respectivamente. Asimismo, crea el esquema conceptual de la base de datos. Los artefactos serán creados utilizando la herramienta CASE *IBM Rational Software Architect (RSA)*.

## TEMARIO

### Tema 1: Arquitectura de análisis

1. Presentación del caso de estudio
2. Creación del modelo de análisis.
3. Organización de casos de uso según análisis.
4. Arquitectura de Análisis.

### Tema 2: Análisis de casos de uso

1. Realizaciones de análisis de casos de uso
2. Diagrama de Clases
3. Diagrama de Comunicación del flujo básico y subflujos de un mantenimiento.

## ACTIVIDADES PROPUESTAS

1. Los alumnos crean el modelo de análisis de un caso propuesto contenido los siguientes diagramas: organización de casos de uso según análisis y arquitectura de análisis.
2. Los alumnos desarrollan las realizaciones de análisis de un caso de uso propuesto.

## 1. ARQUITECTURA DE ANÁLISIS

A continuación, se presenta el caso de estudio que desarrollaremos en este ciclo. En esta primera parte, con la información del caso, se pide elaborar los siguientes artefactos:

1. Modelo de casos de uso del Negocio
2. Modelo de análisis del negocio
3. Modelo de casos de uso
4. Modelo de análisis con los siguientes diagramas:
  - a. Organización de casos de uso según análisis
  - b. Arquitectura de análisis

### CASO DE ESTUDIO: “CLÍNICA EL ÁNGEL”

La “Clínica El Ángel” es una empresa dedicada a la atención médica tanto en emergencia como en atención por consultorios externos. Para ello, cuenta con un equipo médico especializado y excelentes equipos de tecnología de punta. Se está inaugurando una nueva sucursal en el Cono Norte de la ciudad y se espera captar el 30% del mercado de esta parte de la ciudad. Para lograrlo, se espera tener un control total de los consultorios y obtener un tiempo de atención menor en un 30% que la sucursal principal.

Se desea desarrollar un sistema que dé soporte a los principales procesos de la empresa y los requerimientos solicitados que se detallan a continuación:

El Departamento de Emergencia brinda servicios de atención médica a personas que en forma súbita o violenta presentan alteración de su salud, que pone en riesgo su vida o que puedan dejar una secuela permanente. Diariamente llegan pacientes a la clínica que son atendidos por una enfermera, que genera una Orden de Atención de Emergencia; todas las personas que llegan son atendidas por tratarse de una emergencia y se les apertura una historia clínica si es que no tienen una historia clínica. Para generar una Orden de Atención, la enfermera debe especificar los datos del paciente y realizar una búsqueda para asignar el médico que se encuentra disponible, luego imprime la Orden de Atención. Después de terminada la atención, la jefa de enfermeras registrará en un estado de cuenta por pagar todos los gastos de la Orden de Atención, realizando la consulta previa por Orden de Atención. Internamente, se genera una deuda por pagar cuyo concepto será gastos de emergencia. El paciente deberá ir a caja a cancelar el estado de cuenta y solicitará su comprobante de pago correspondiente. El cajero de emergencia generará el comprobante de pago, realizando la búsqueda por deuda por pagar por paciente filtrando el concepto. El paciente puede realizar el pago con cualquier tarjeta de crédito o débito. Adicionalmente, la jefa de enfermeras deberá contar con una opción de consulta detallada de órdenes de atención; para ello, previamente, consultará las órdenes de atención por algún criterio de búsqueda.

El servicio de Consulta Externa permite la atención médica especializada en los horarios de lunes a sábado entre las 8:00 a.m. y las 7:00 p.m. Cuando un paciente llega al área de consultorios y desea atenderse, es recibido por una anfitriona quien le entregará un ticket de atención y le derivará al pool de recepcionista para que saque su cita. La recepcionista llamará al paciente según

su ticket; luego, generará la cita, verificando previamente si el paciente ya cuenta con una historia clínica, pero, si no tiene, le apertura una historia clínica. Para generar la cita, la recepcionista debe especificar los datos del paciente, especialidad y médico que atenderá al paciente; para ello, consulta los datos del paciente de la historia clínica y asigna el médico que el paciente solicita realizando una búsqueda previa del horario del médico; por último, completa los datos para registrar la cita. Internamente, se generará una deuda por pagar cuyo concepto será gastos de cita. La recepcionista entregará la cita al paciente e informa que deberá acercarse a caja para cancelar la cita y solicitar su comprobante de pago. El cajero de consultorios generará el comprobante de pago, realizando la búsqueda por deuda por pagar por paciente filtrando el concepto. No obstante, el paciente podrá pagar con cualquier tarjeta de crédito o débito.

Posteriormente, el paciente entregará la cita a la enfermera del piso, ella entregará la cita al médico, quién lo llamará para atenderlo. Después de examinarlo, el médico registrará en la historia clínica el resultado de la cita. Obteniendo previamente el dato de la cita. Al final, el médico le entrega su receta al paciente quien se acercará a farmacia a recoger sus medicinas. Asimismo, el técnico de farmacia solicitará la receta y registrará una Orden de Entrega de Medicamentos, obteniendo previamente el dato de la cita; posteriormente, imprime la Orden e internamente se generará una deuda por pagar, cuyo concepto será por gastos de medicina y, luego, le informa al paciente que cancele en caja. El cajero de farmacia generará el comprobante de pago, realizando la búsqueda por deuda por pagar por paciente filtrando el concepto. El paciente puede pagar con cualquier tarjeta de crédito o débito.

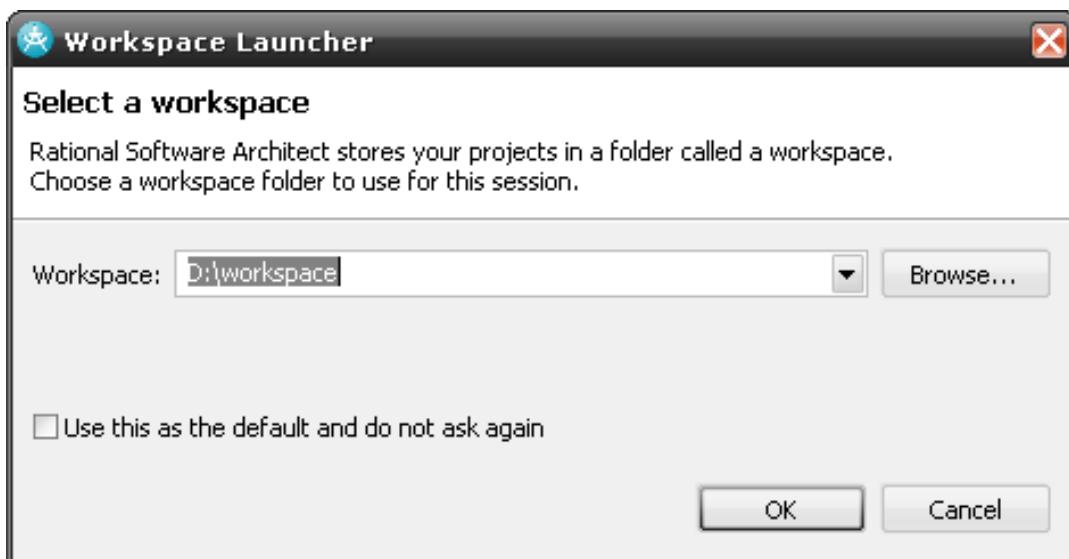
Adicionalmente, se está solicitando que el nuevo sistema pueda...

1. Permitir al administrador la creación de las series de comprobantes de pago.
2. Permitir al administrador el mantenimiento de cajeros y la asignación de los comprobantes.
3. Permitir al cajero la apertura de caja.
4. Permitir al cajero que pueda generar el cuadre de caja
5. Permitir al Administrador el cierre de la caja
6. Permitir al administrador la anulación de los comprobantes

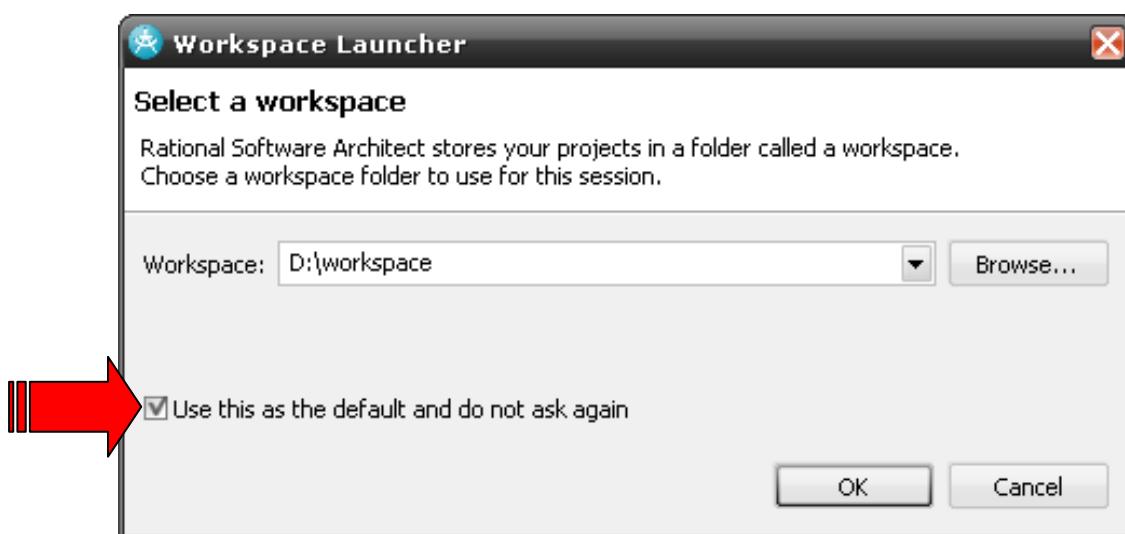
## Creación del workspace

Para empezar a trabajar con IBM RSA, se debe definir una carpeta como workspace (espacio de trabajo), la cual contendrá el proyecto que se creará en el entorno de la herramienta. A continuación, se indican los pasos:

1. Al cargar el IBM RSA se muestra la siguiente ventana. Con el botón **Browse** se ubica la ruta del workspace.



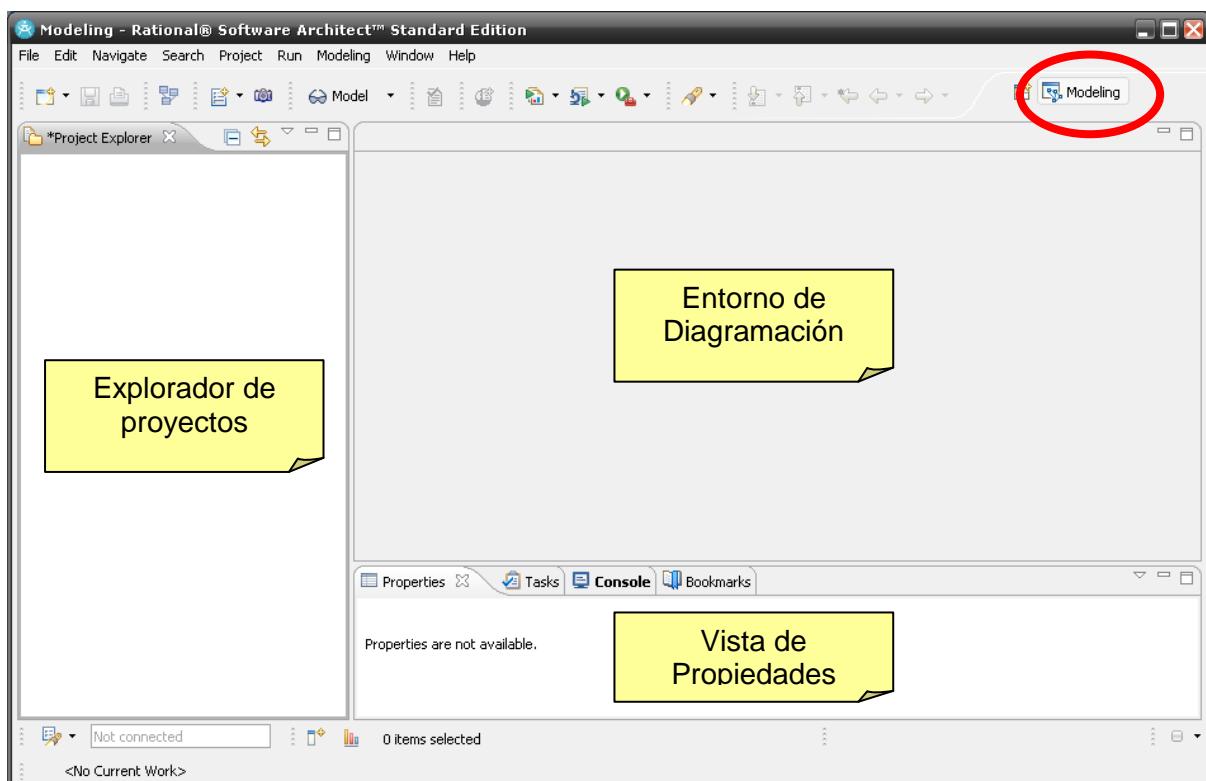
2. Ahora, active la opción de la parte inferior para que la siguiente vez no pida especificar un workspace. Luego, de clic en **OK**.



3. A continuación, se presentará una página de bienvenida, la cual se mostrará solo si se define por primera vez el workspace. Para trabajar en el entorno se cierra esta página.

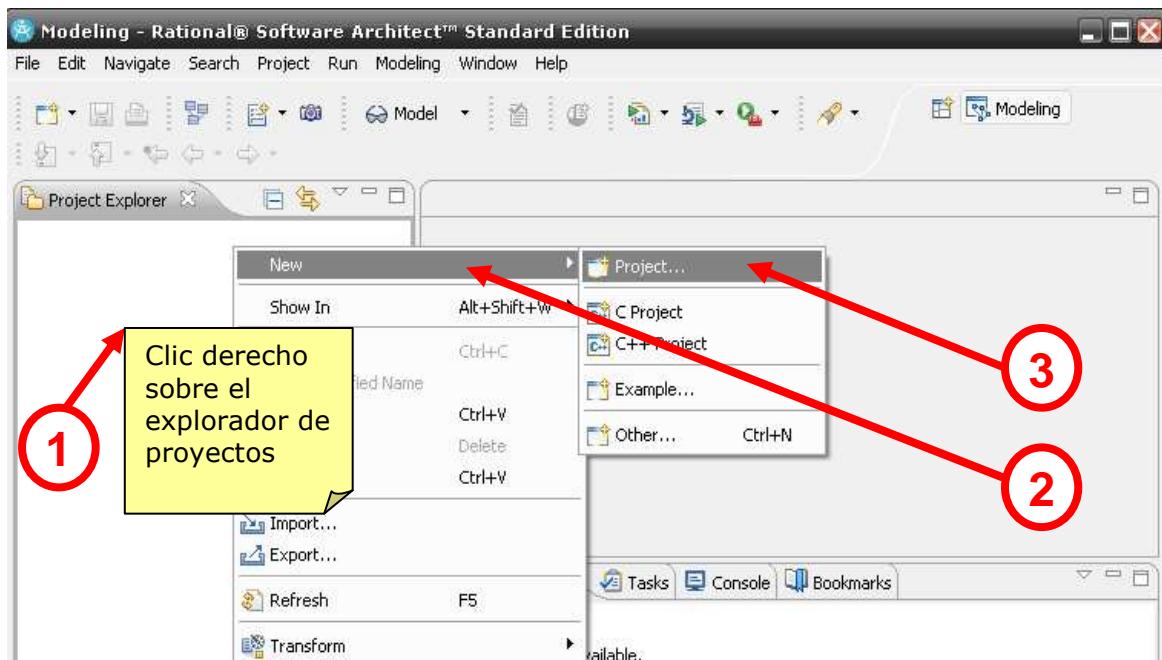


4. Luego, se visualizará la perspectiva **Modeling**, con la cual podrá crear proyectos que contendrán modelos con UML.

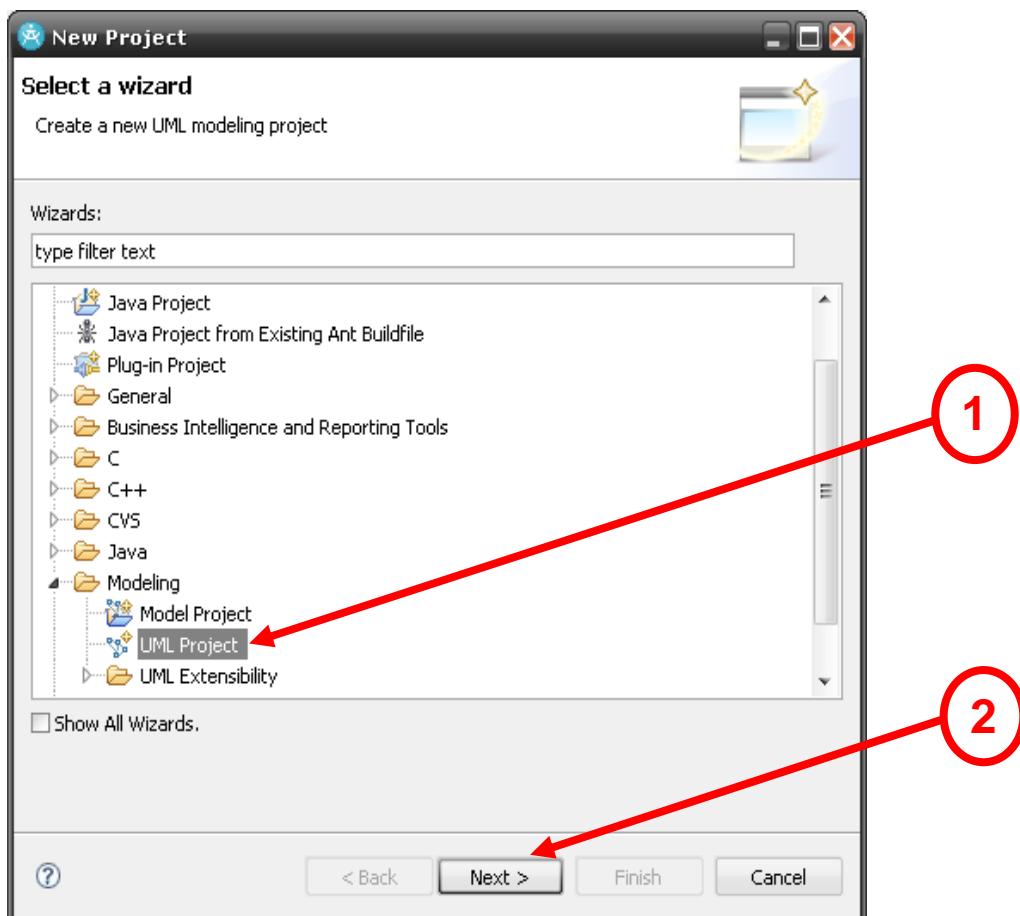


## Creación del proyecto

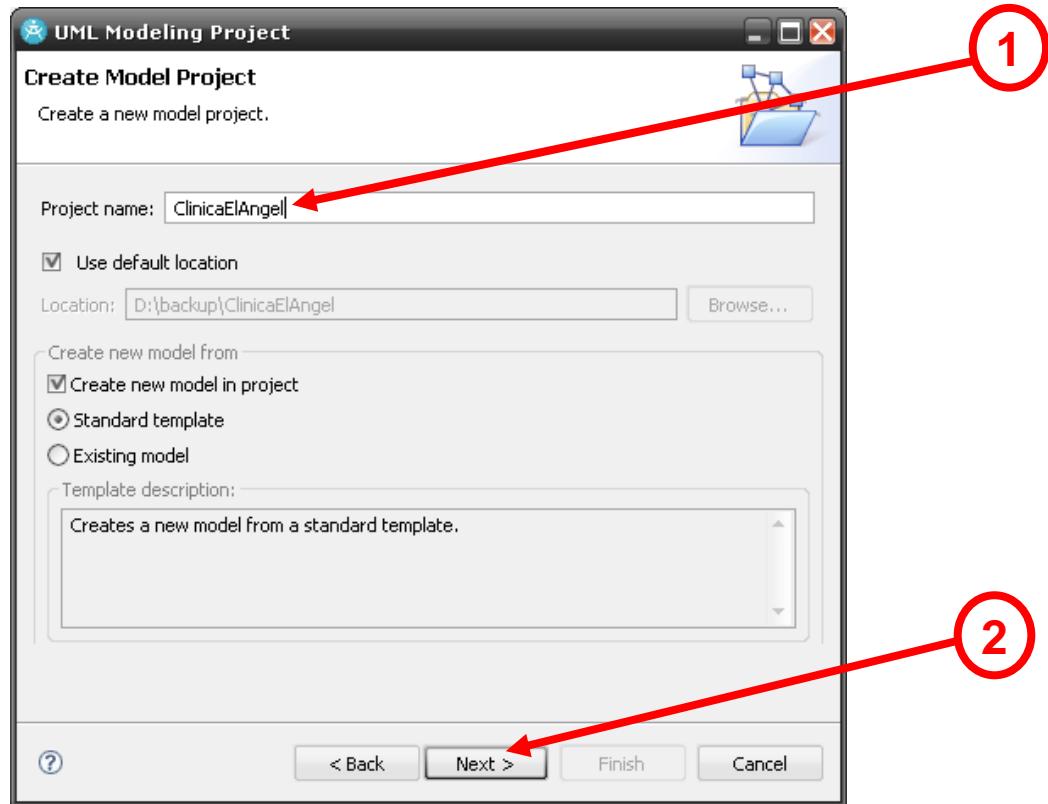
1. Seleccione crear nuevo proyecto.



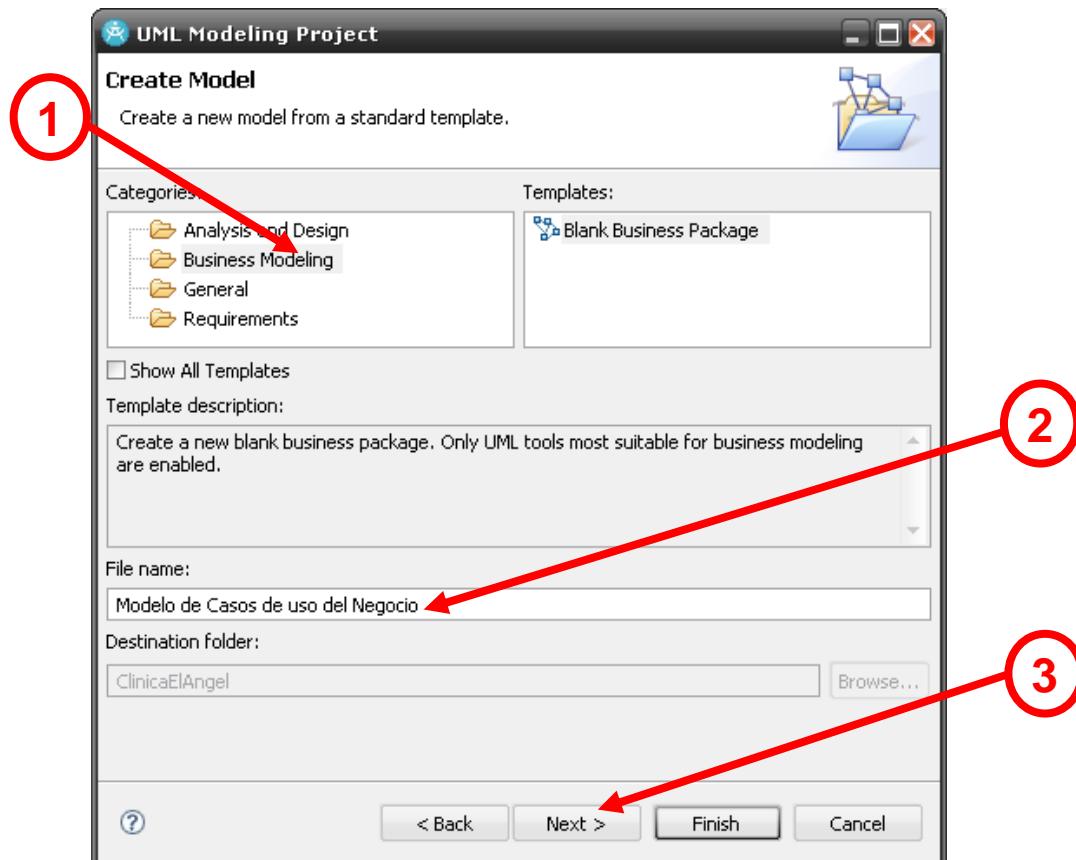
2. Seleccione tipo de proyecto (Model Project o UML Project).



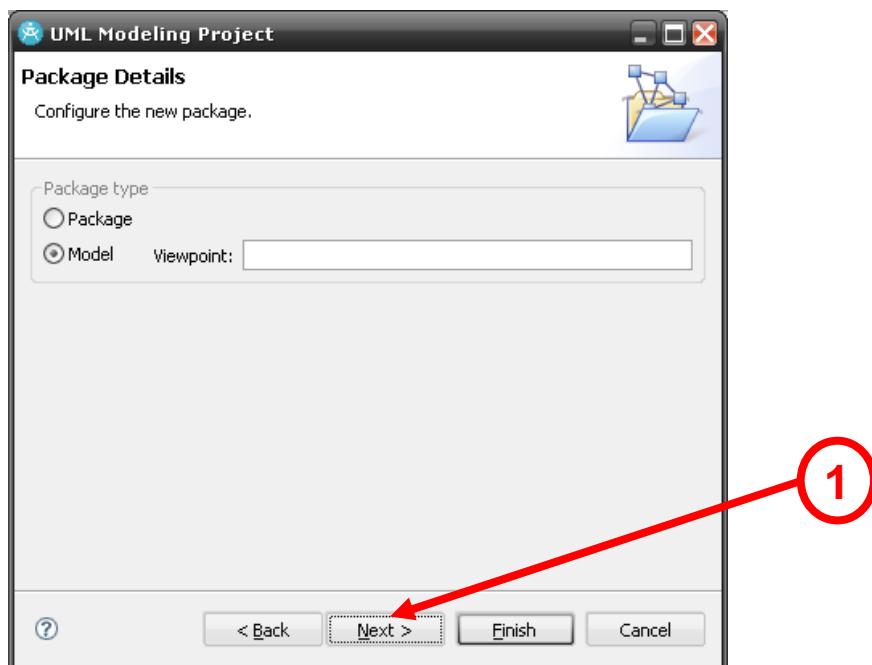
3. A continuación, edite un nombre para el proyecto “ClínicaElAngel”.



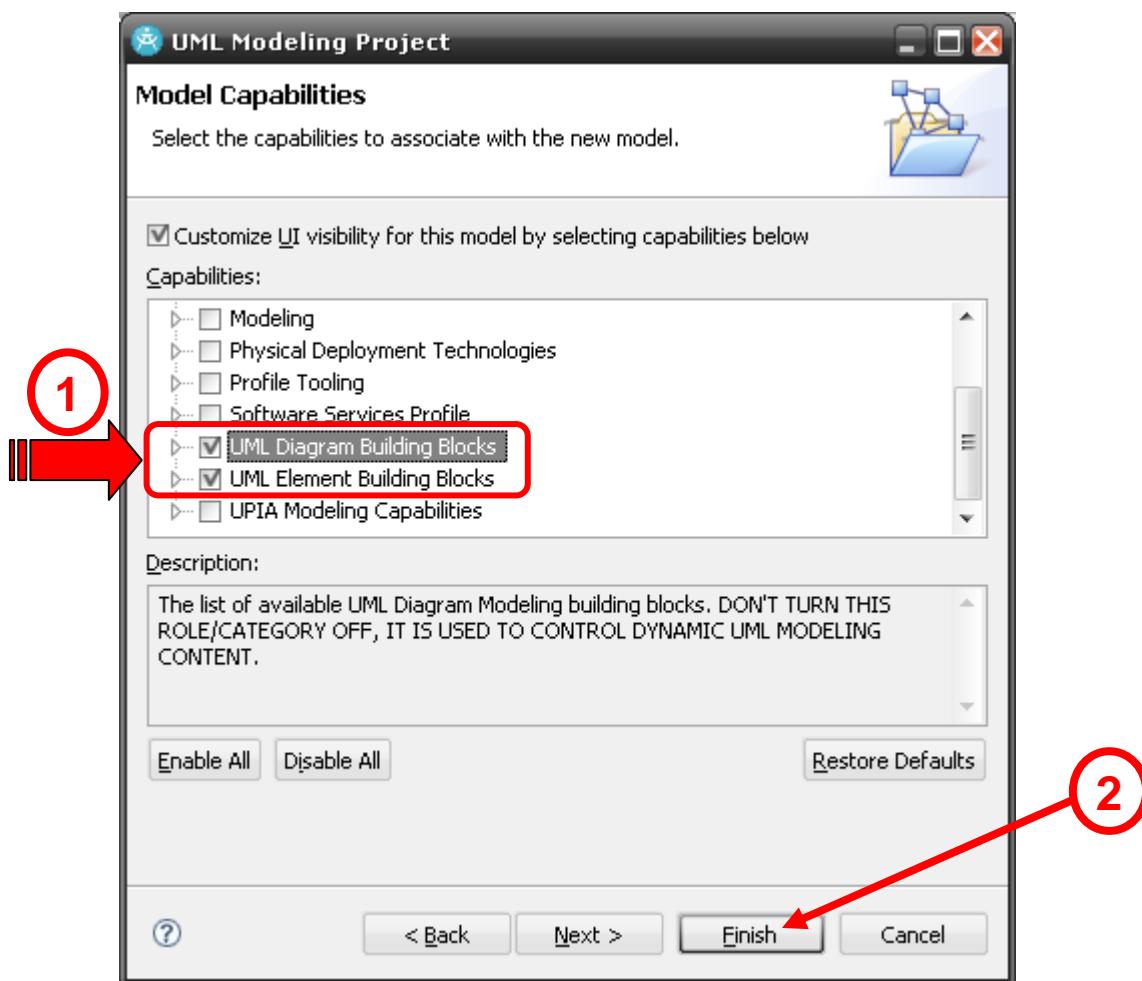
4. Luego, seleccione la categoría de modelado de negocio e ingrese el nombre del modelo a crear.



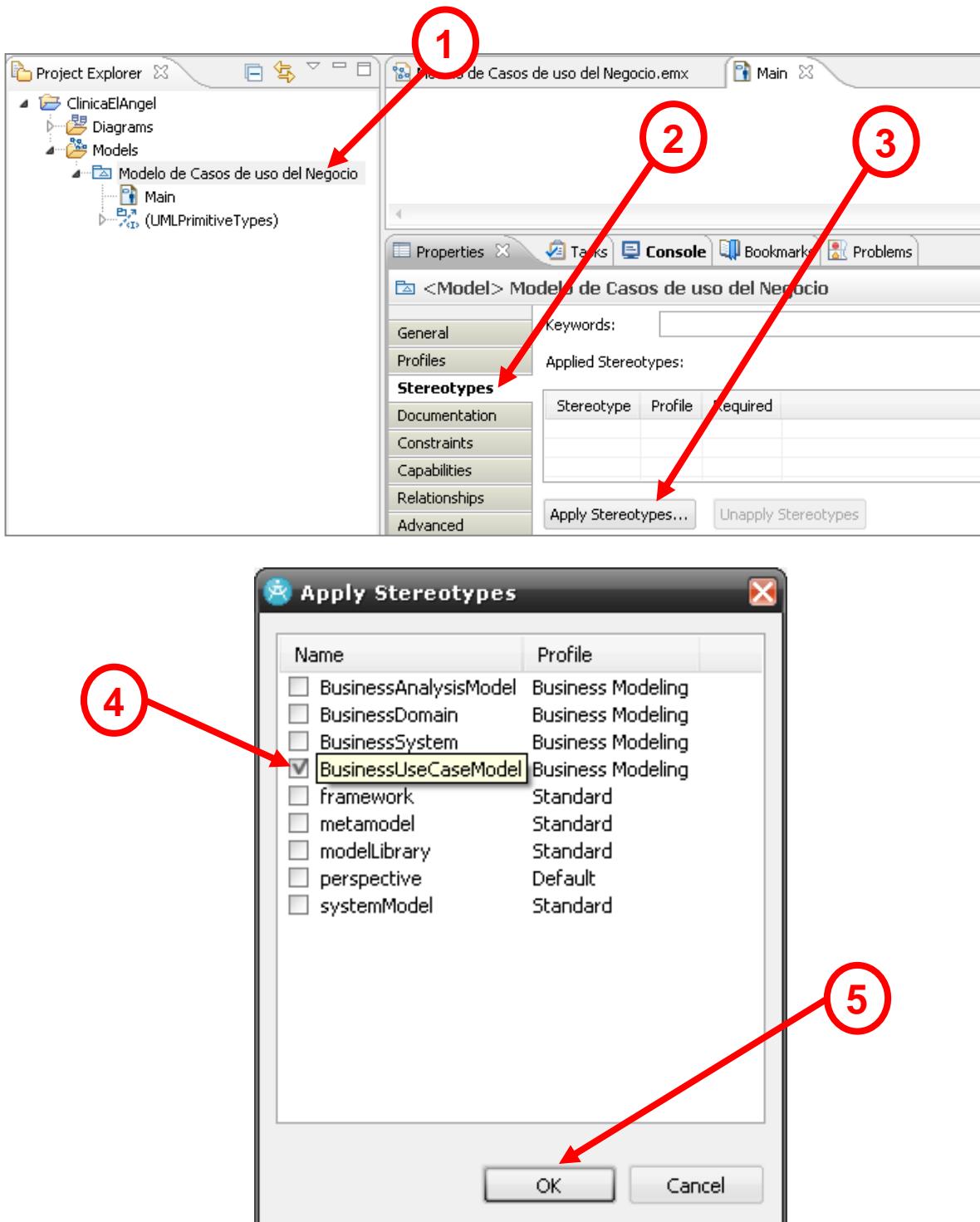
5. Seleccione **Next**.



6. Ahora, active las capacidades de UML (diagramas y elementos).

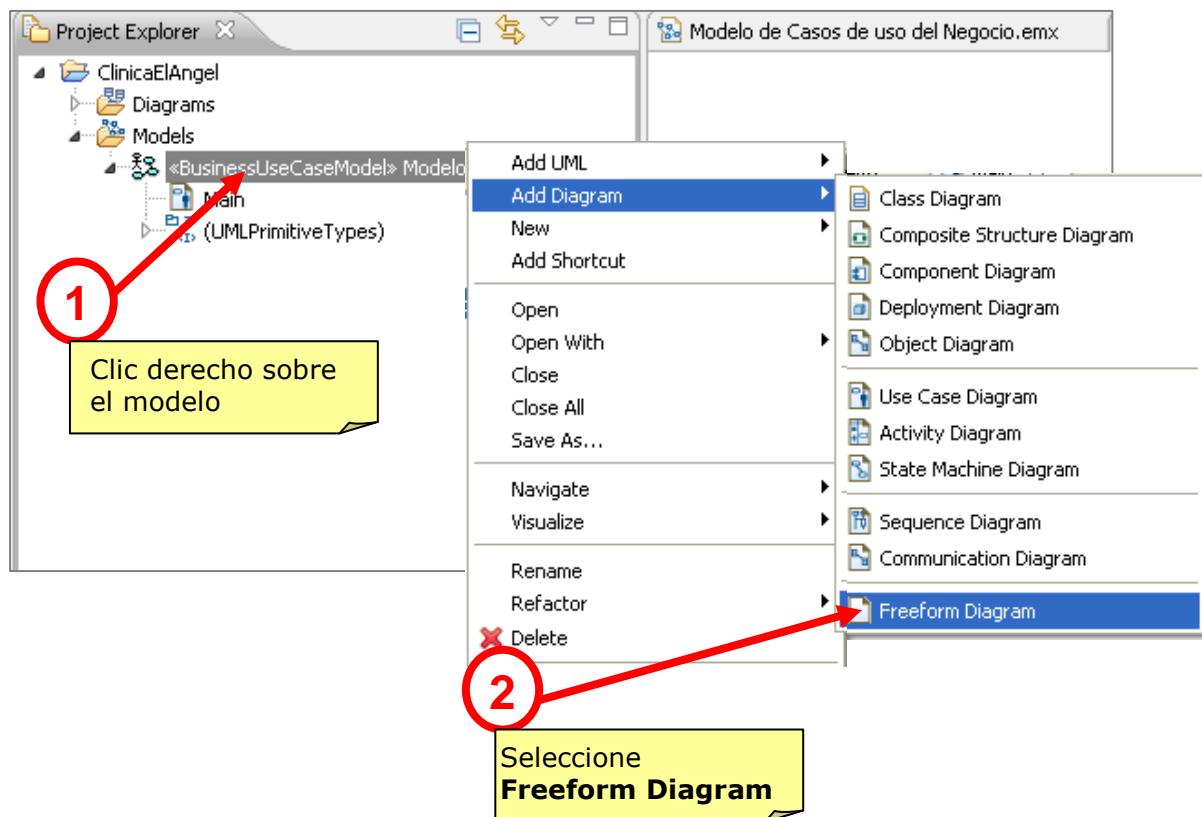


7. Aplique estereotipo del Modelo de casos de uso del negocio a *Business Use Case Model*.

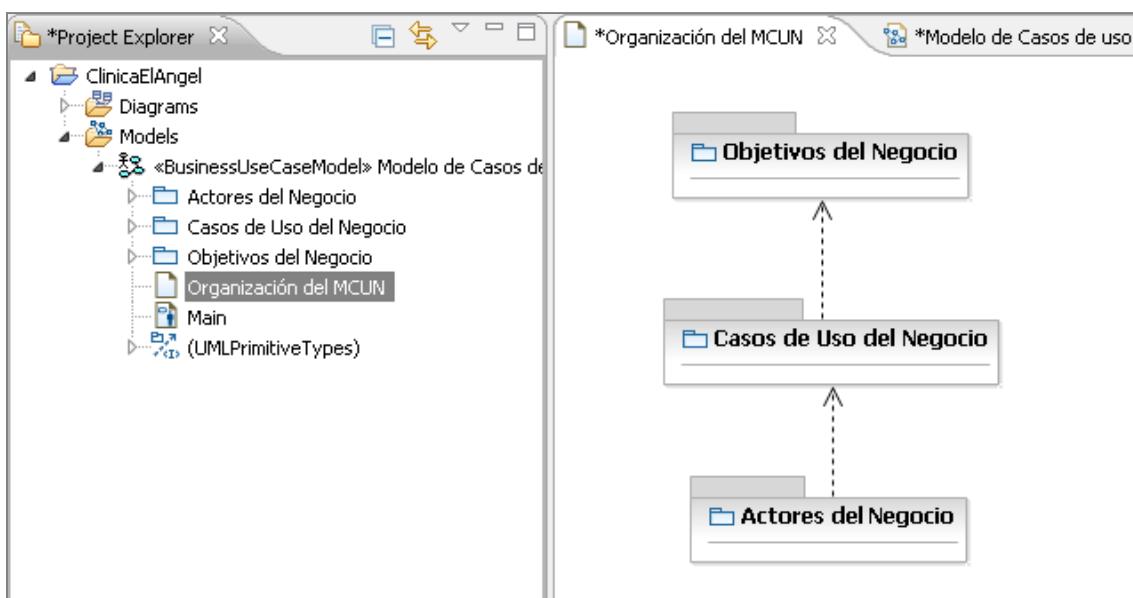


## Completar el Modelo de casos de uso del negocio

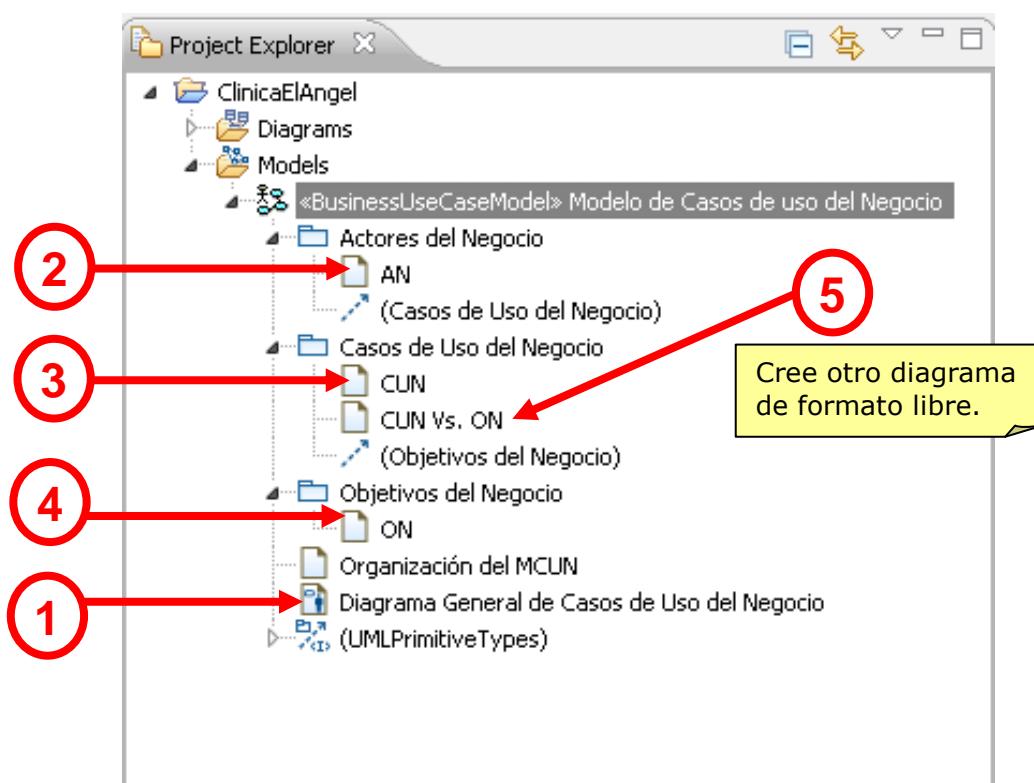
1. Agregue un diagrama de formato libre con el nombre “Organización del MCUN”.



2. Sobre el diagrama “Organización del MCUN” agregue los paquetes del modelo y sus dependencias.

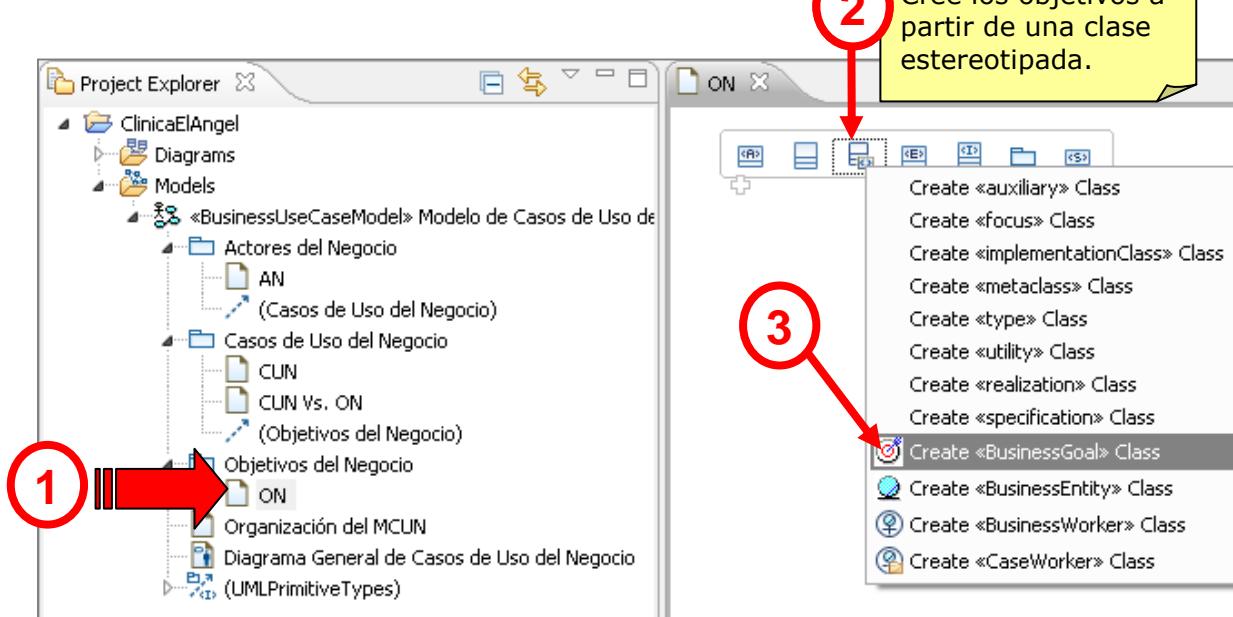


3. A continuación, renombre el diagrama *Main* a “Diagrama General de Casos de Uso del Negocio” y los diagramas de cada paquete. Luego, agregue otro diagrama de formato libre con el nombre “CUN Vs. ON” para representar los Objetivos del Negocio versus Casos de uso del Negocio. Debe quedar así:



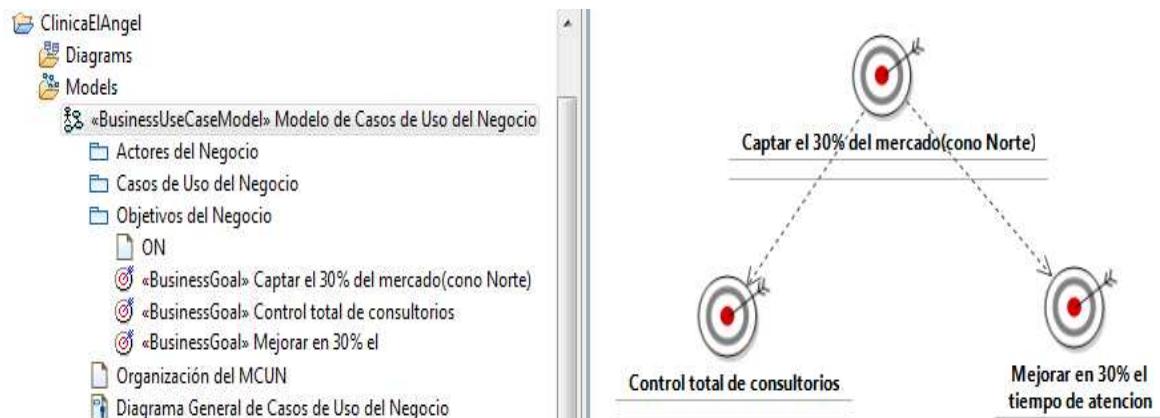
4. Ahora empiece a crear los elementos para el Modelo de casos de uso del negocio. Primero, identifique los objetivos del negocio.

**¿Cuáles son los objetivos del negocio?**



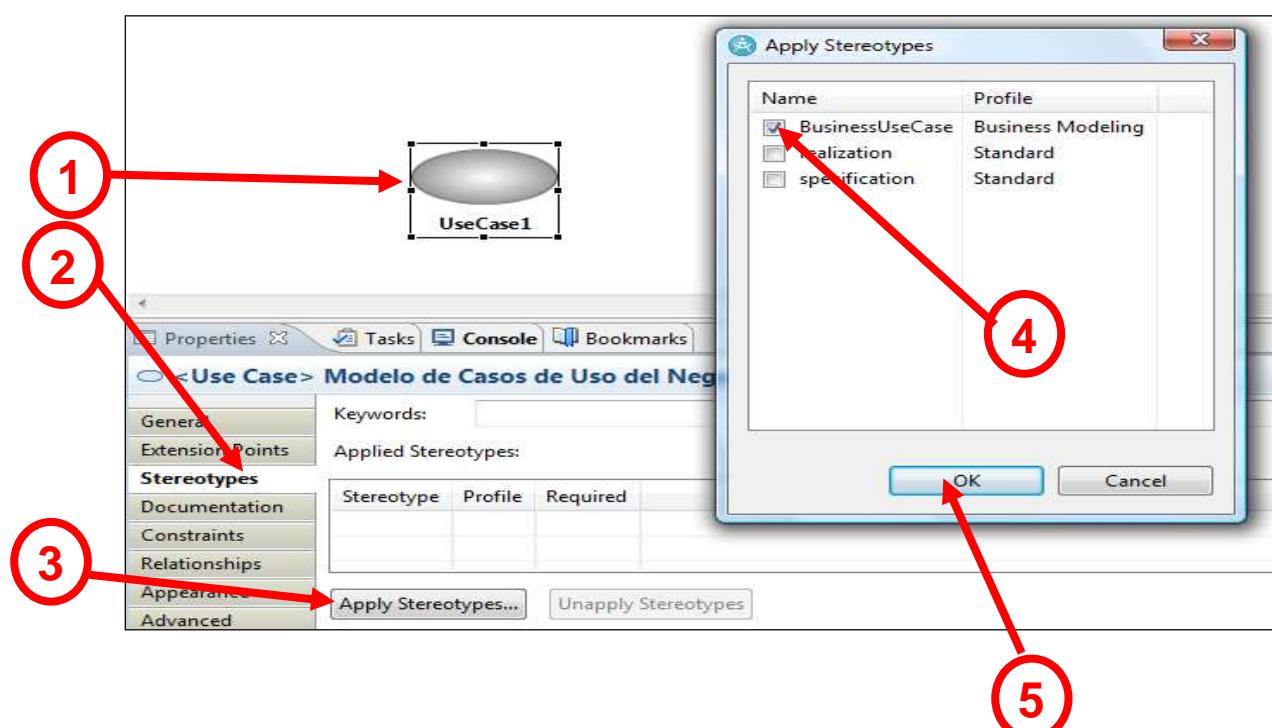
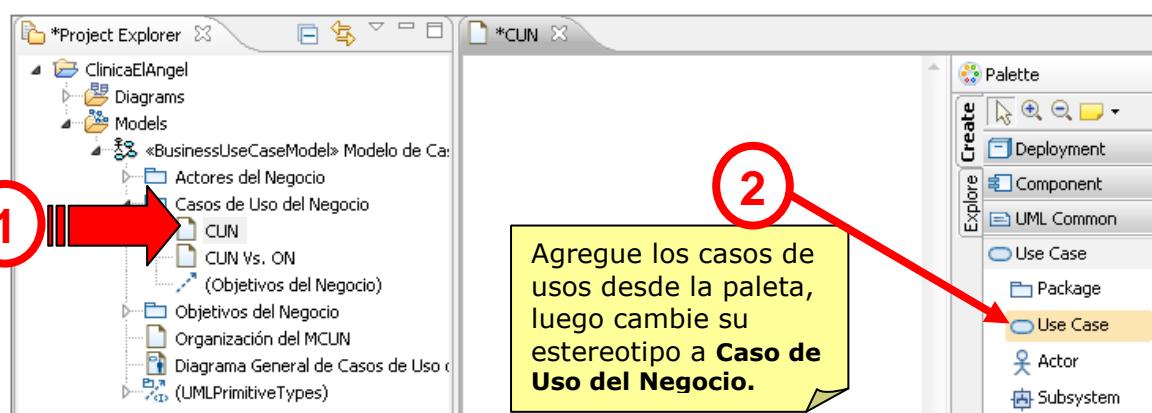
Los objetivos serían tres:

- 1.- Captar el 30% del mercado (Cono Norte) que es un Objetivo general.
- 2.- Control total de consultorios que es un Objetivo específico.
- 3.- Mejorar en 30% el tiempo de atención que es un Objetivo específico.



5. Aquí represente los casos de uso del negocio.

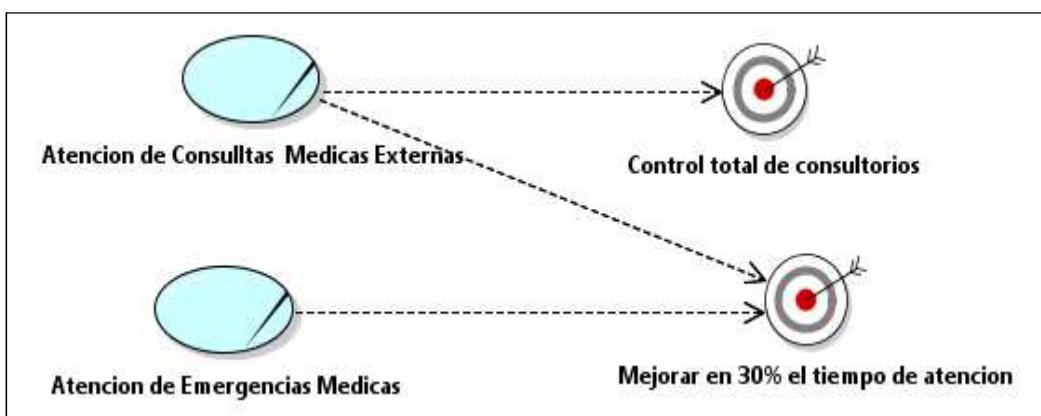
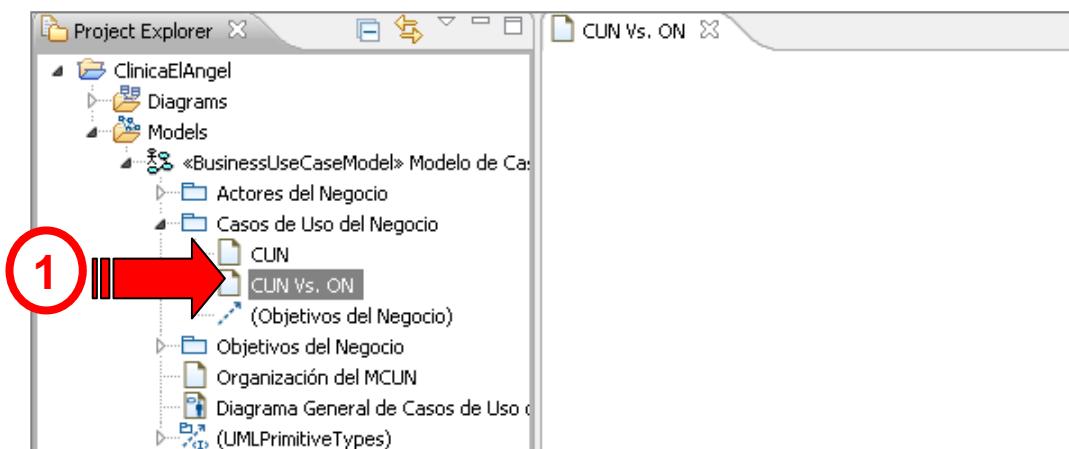
**¿Qué casos de uso del negocio se identificaron?**





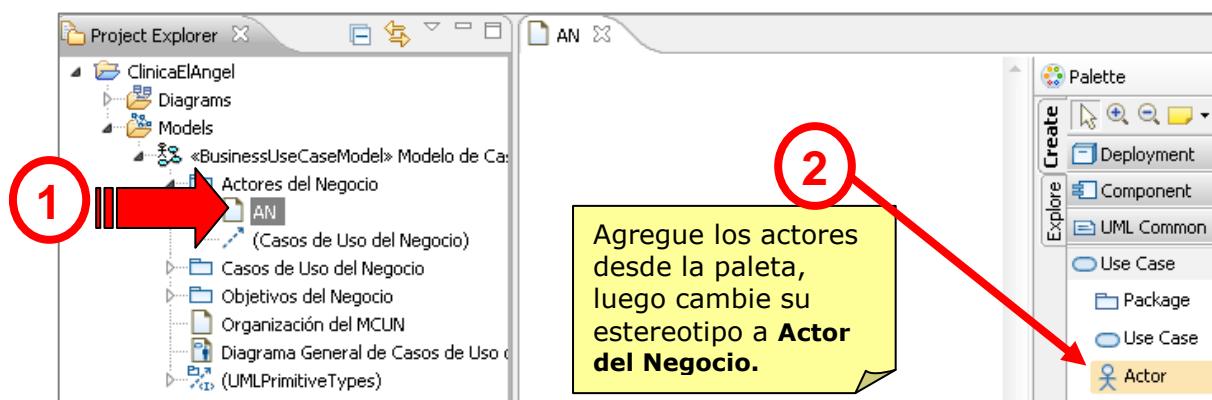
6. Sobre este diagrama arrastre los objetivos y casos de uso del negocio.

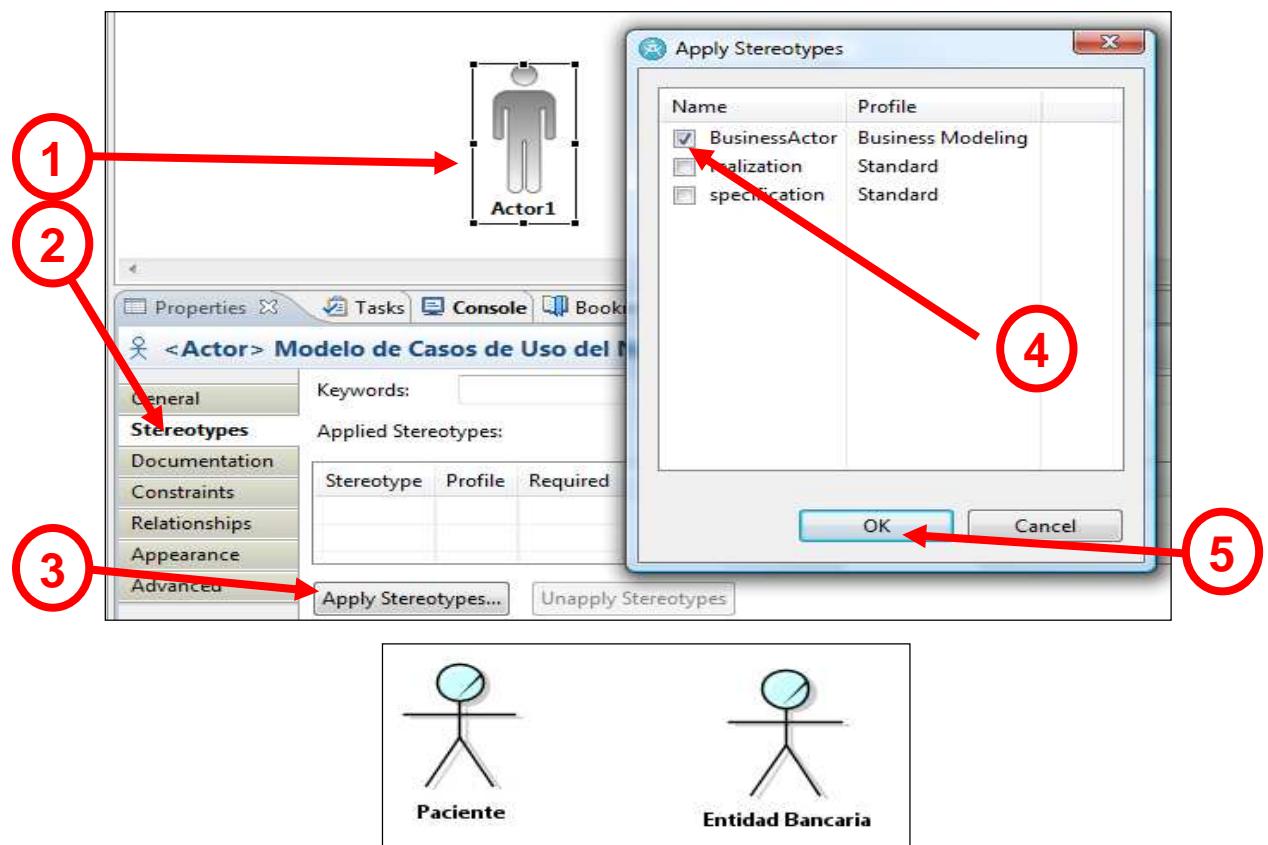
**¿Cuáles son los objetivos que debe alcanzar cada caso de uso de negocio identificado?**



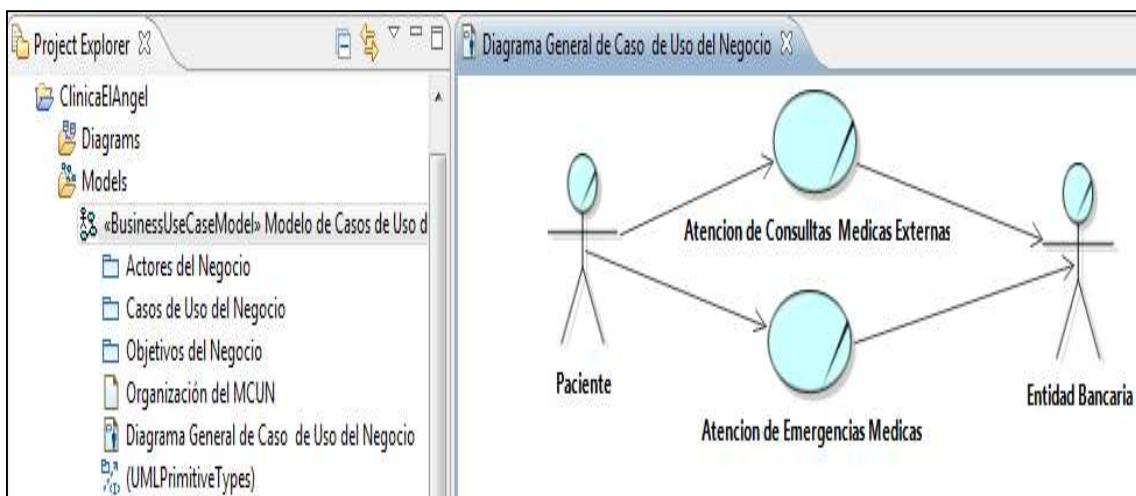
7. Aquí represente los actores del negocio que inician o participan en un caso de uso del negocio.

**¿Qué roles externos a los proceso de negocio se identificaron?**



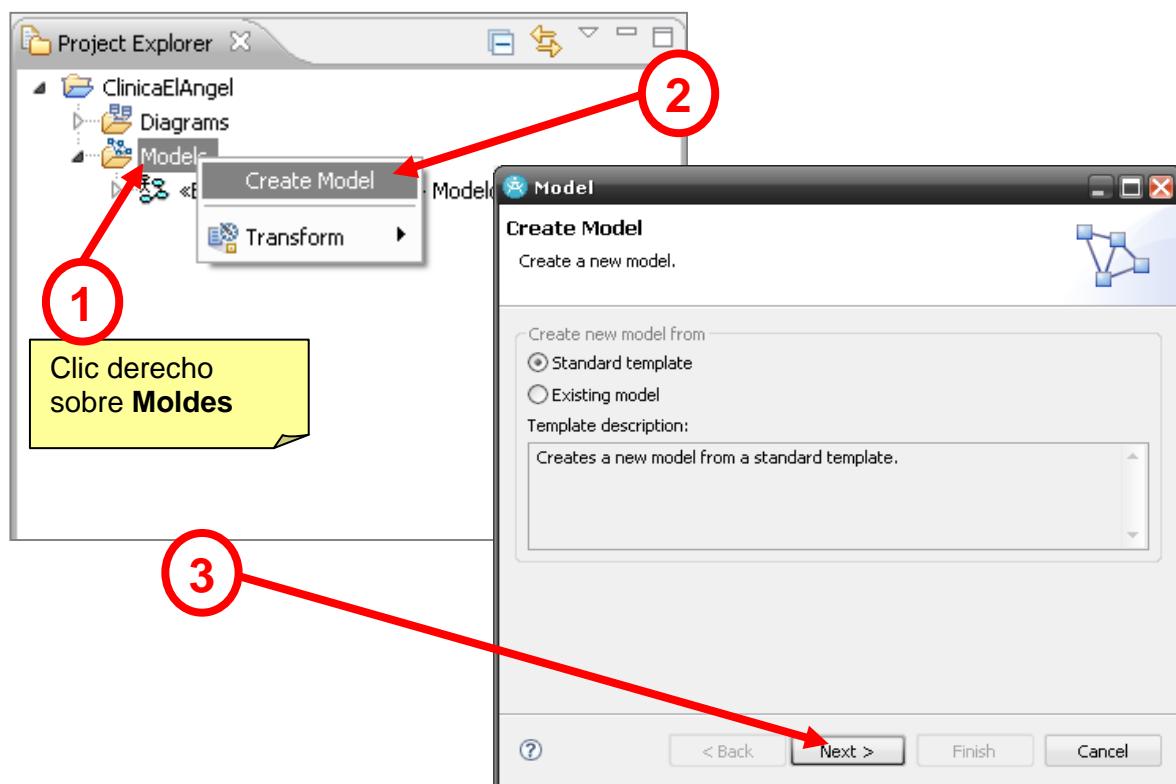


8. Por último, sobre el “Diagrama general de Casos de Uso del Negocio” arrastre los casos de uso y actores del negocio identificados en los pasos anteriores y agregue sus interrelaciones.

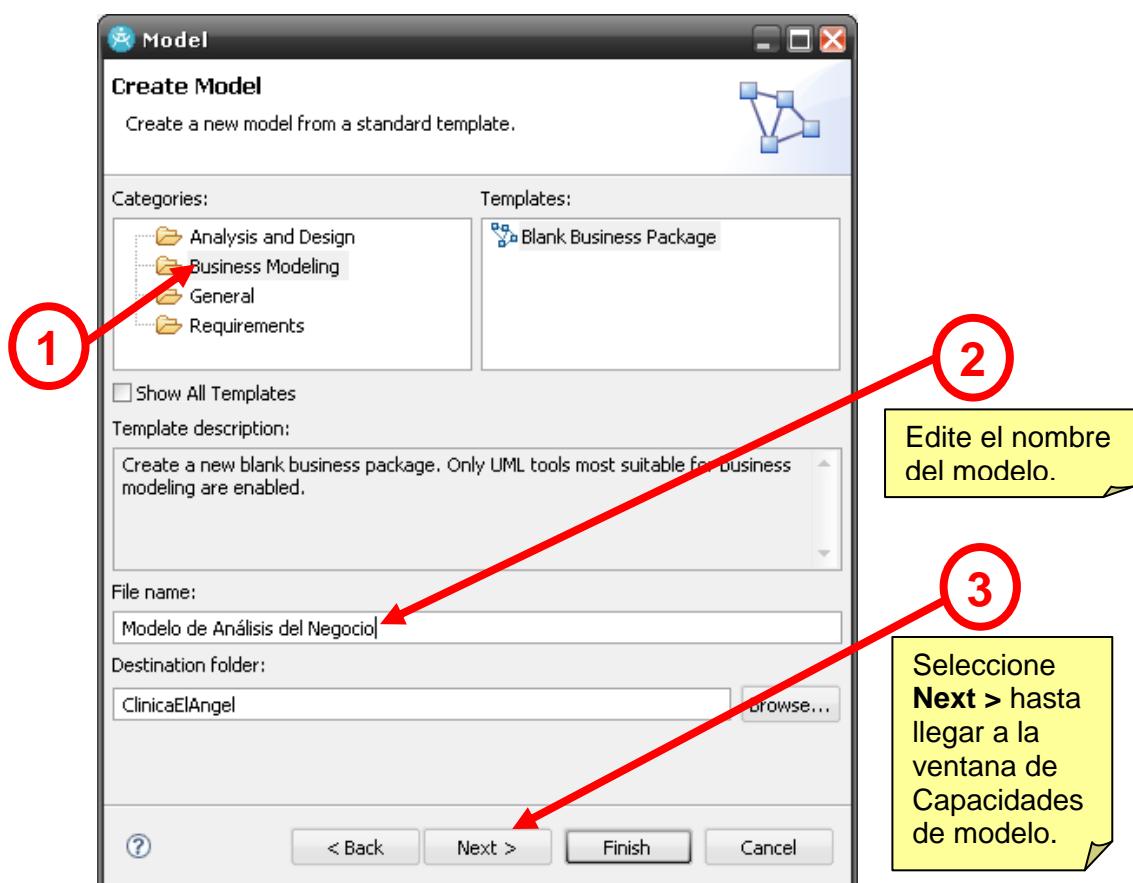


## Creación del Modelo de análisis del negocio

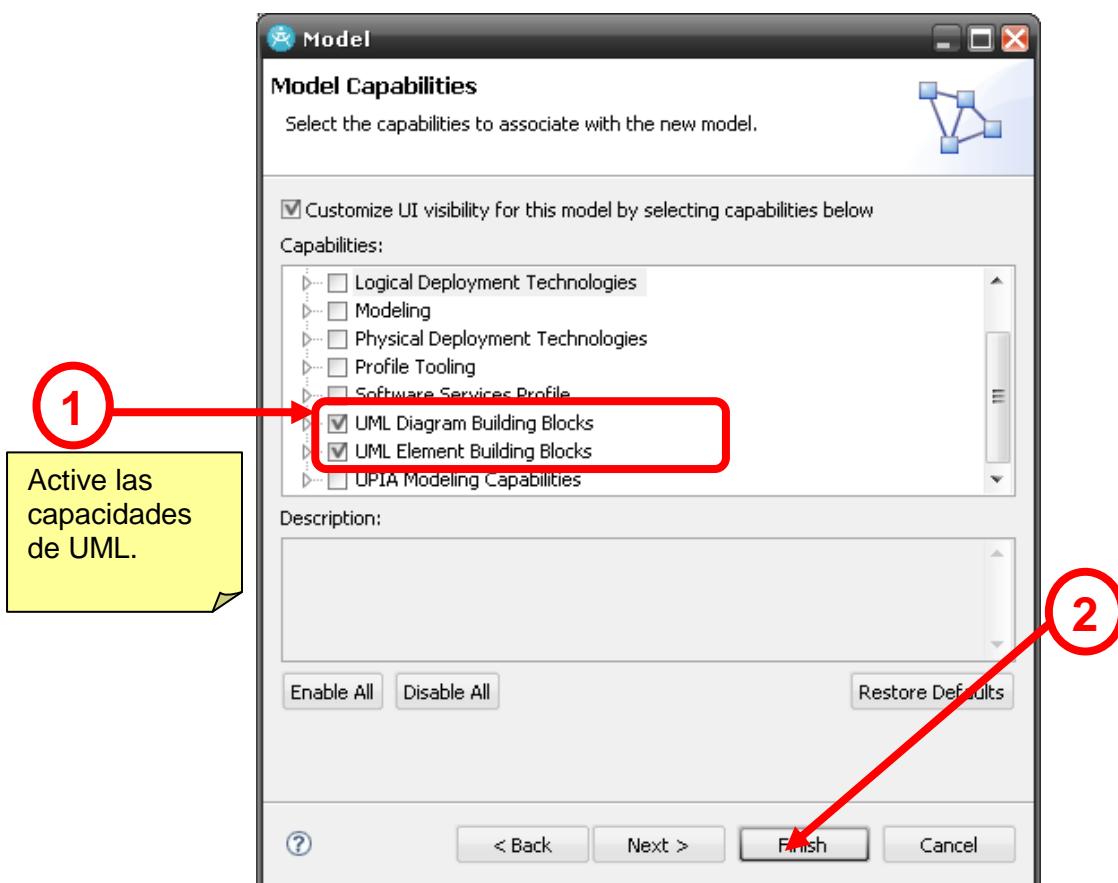
1. Seleccione crear modelo a partir del folder **Models**.



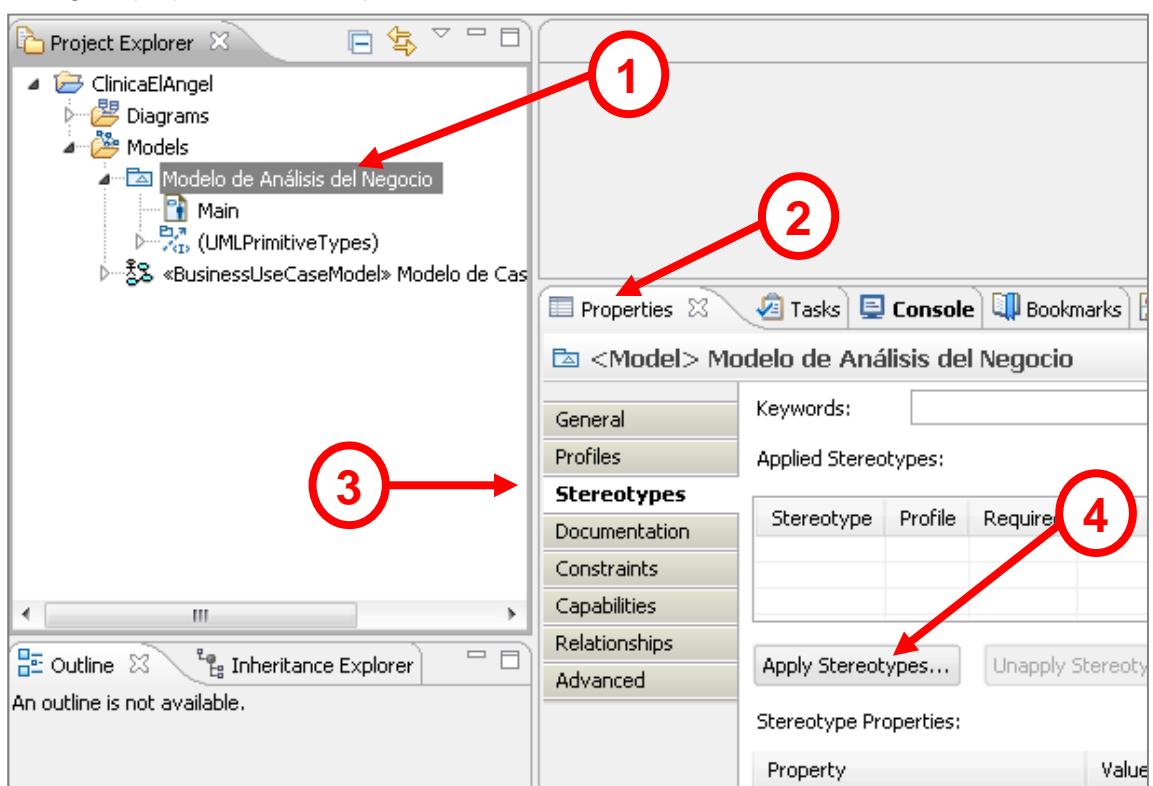
2. Seleccione la categoría ***Business Modeling***.

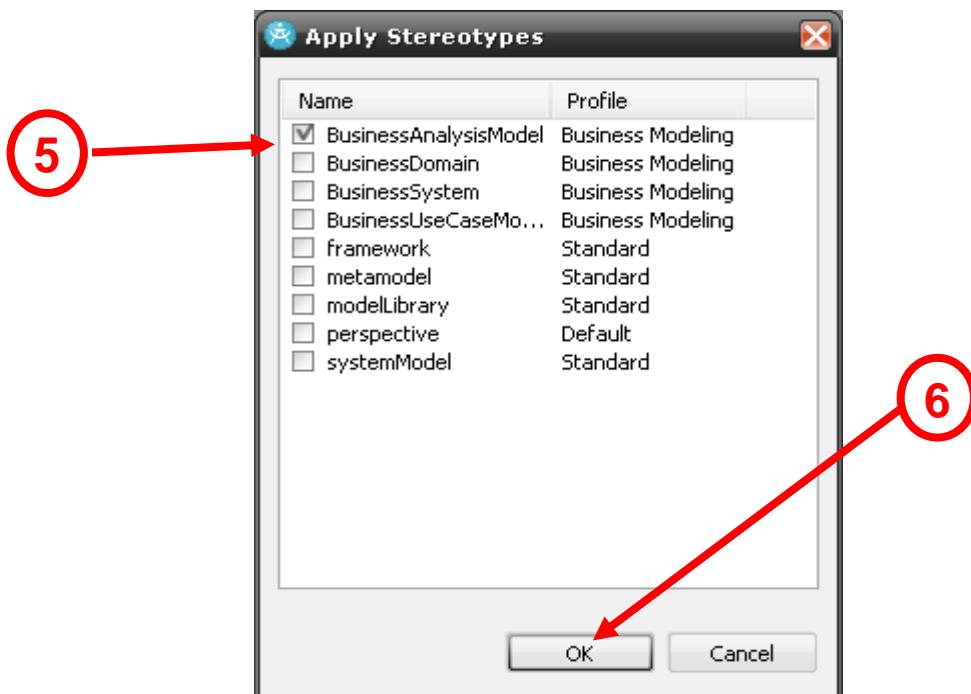


3. Luego, en la ventana de **Capacidades**, active las casillas relacionadas a UML.

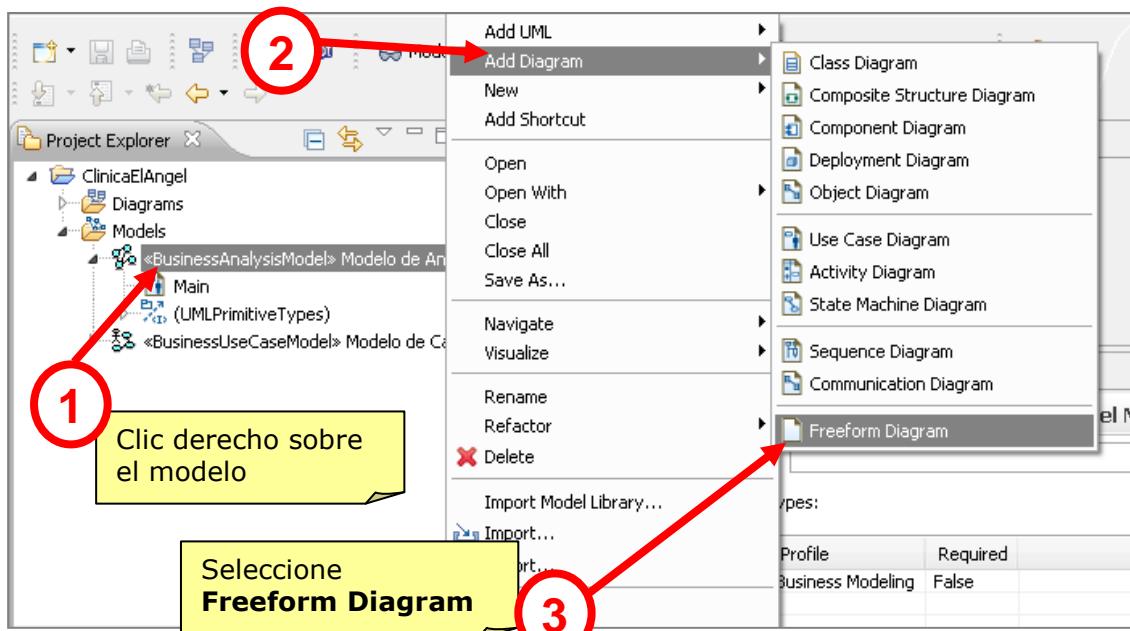


4. Luego, aplique el estereotipo *Bussiness Analysis Model*.

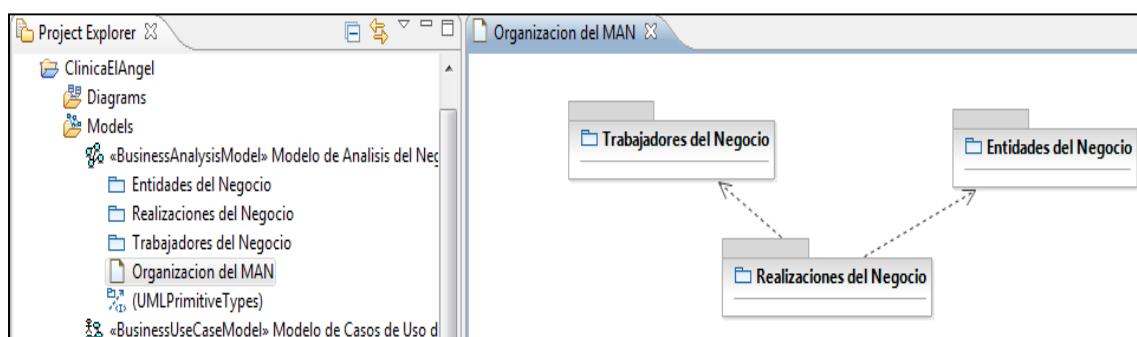




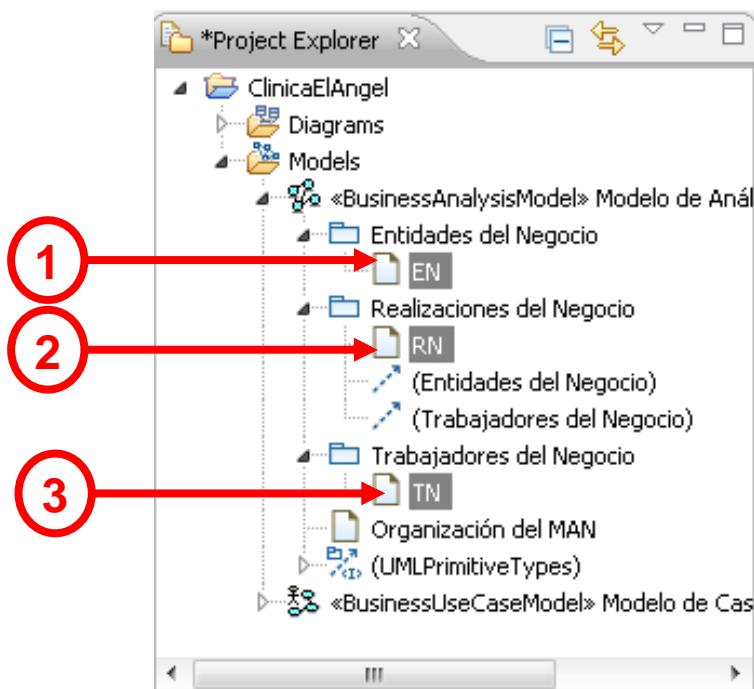
5. A continuación, agregue un diagrama de formato libre con el nombre “Organización del MAN”



6. Elimine el diagrama *Main* y, sobre el nuevo diagrama, agregue los paquetes del modelo y sus dependencias.

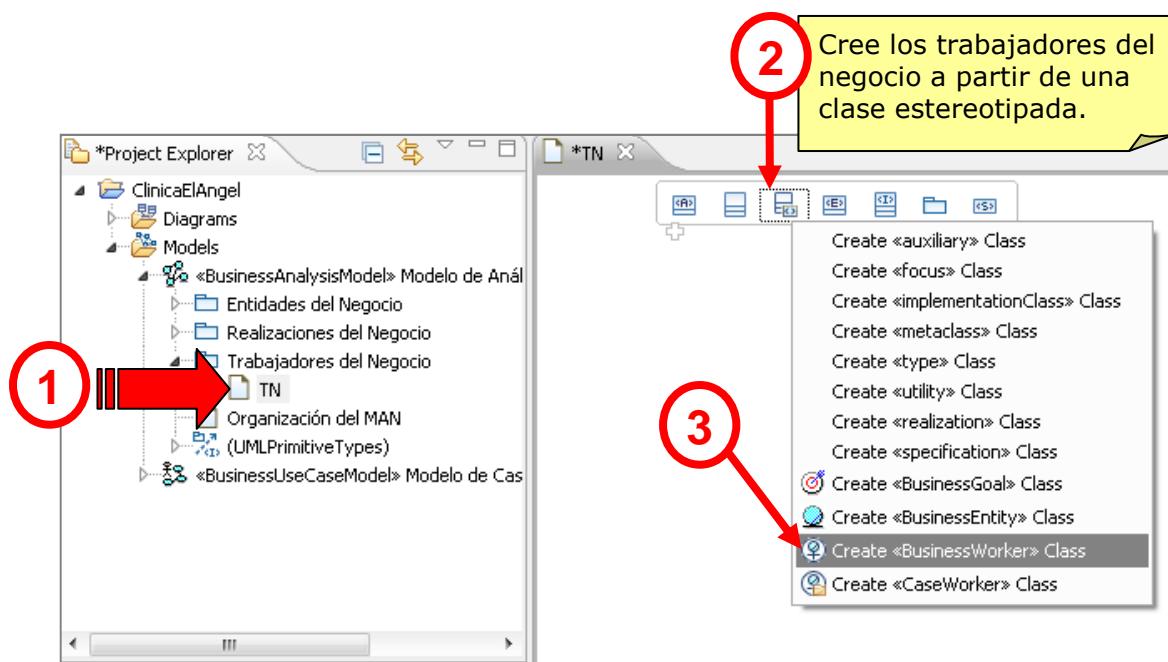


7. Luego, renombre los diagramas de cada paquete. Debe quedar así:



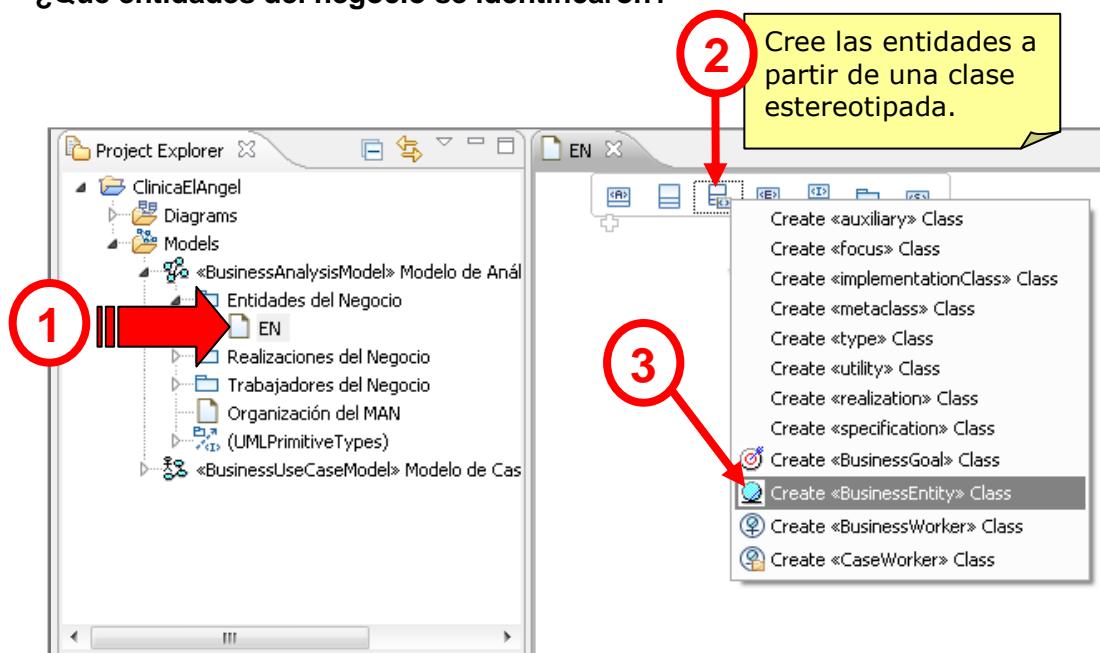
8. Ahora, empiece a crear los elementos identificados para el Modelo de análisis del negocio. En este diagrama, represente los trabajadores del negocio.

**¿Qué roles internos para los procesos del negocio se identificaron?**

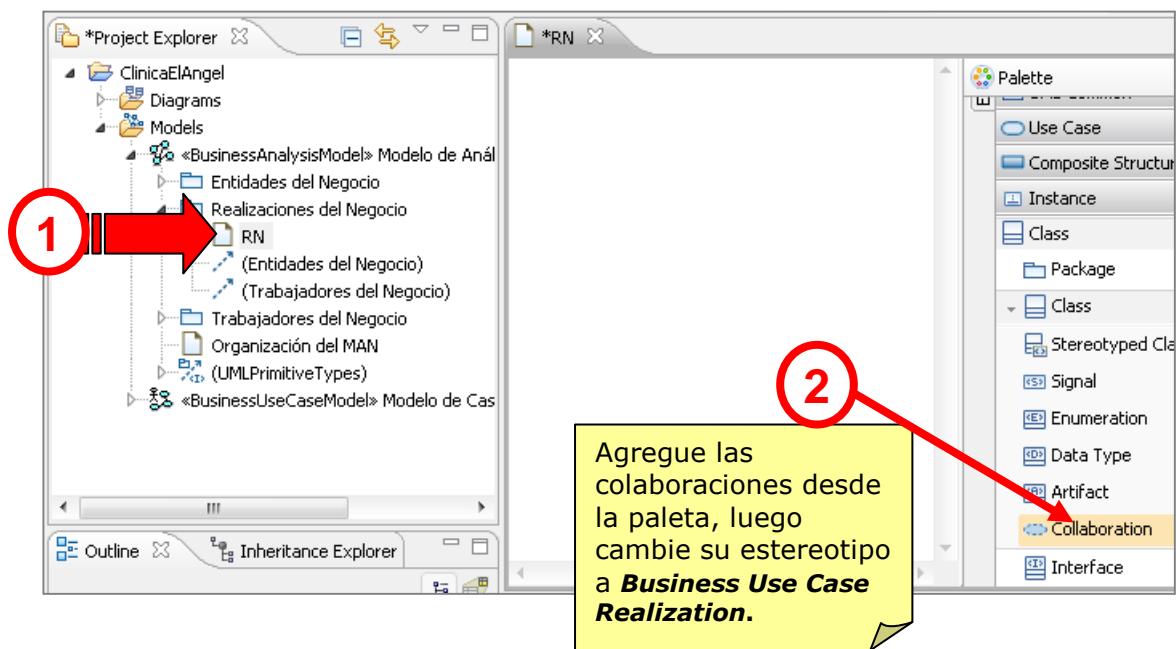


9. Aquí represente las entidades del negocio.

¿Qué entidades del negocio se identificaron?

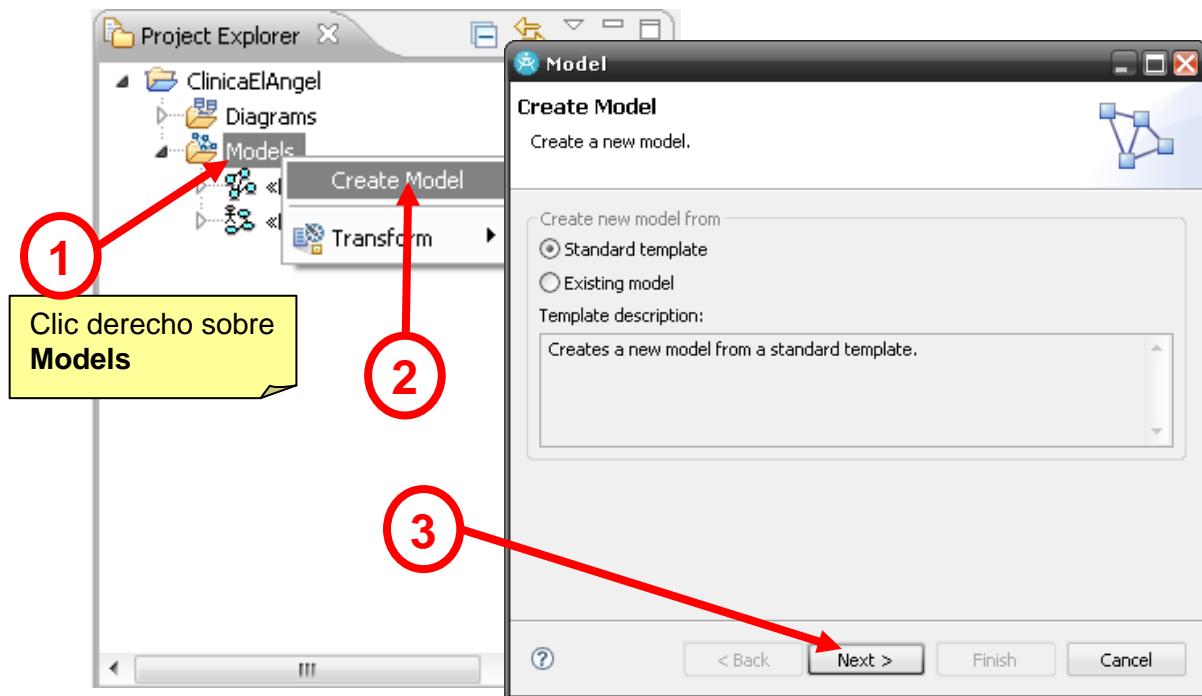


10. A continuación, cree las realizaciones de negocio. Por cada realización, desarrolle el diagrama de clases y de actividades.

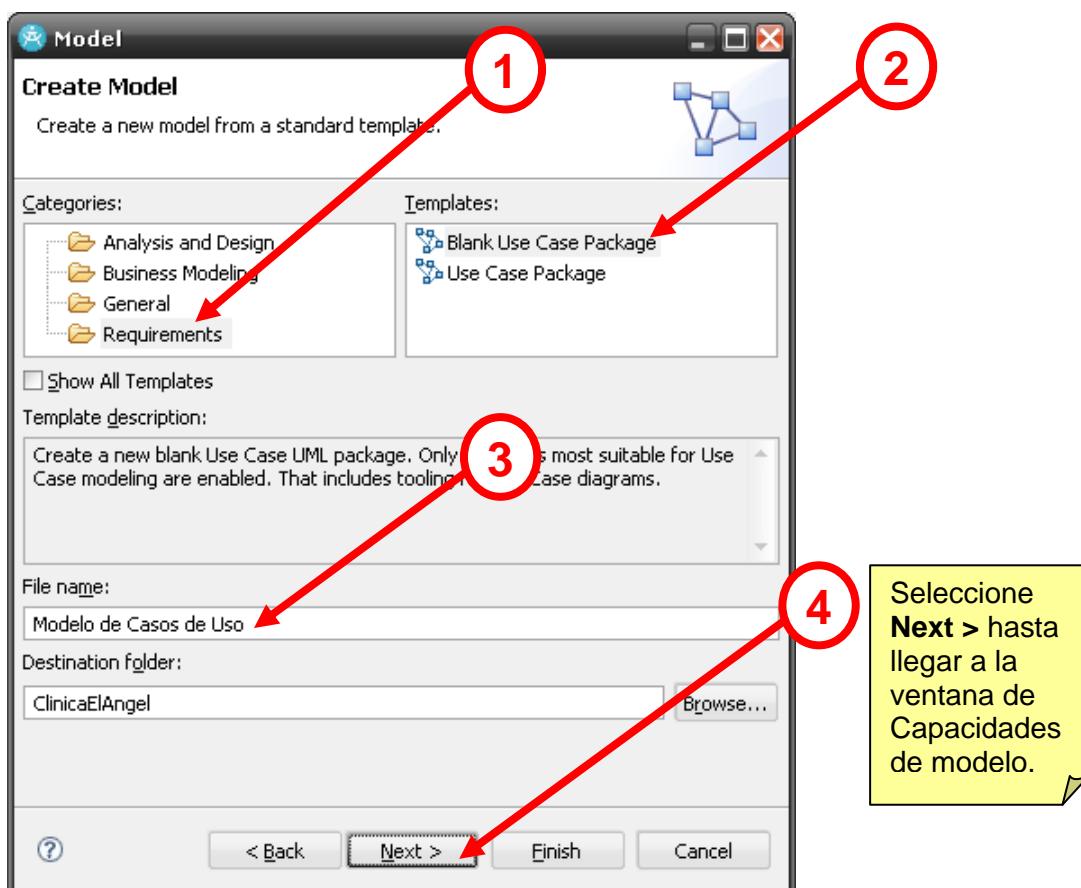


## Creación del Modelo de casos de uso

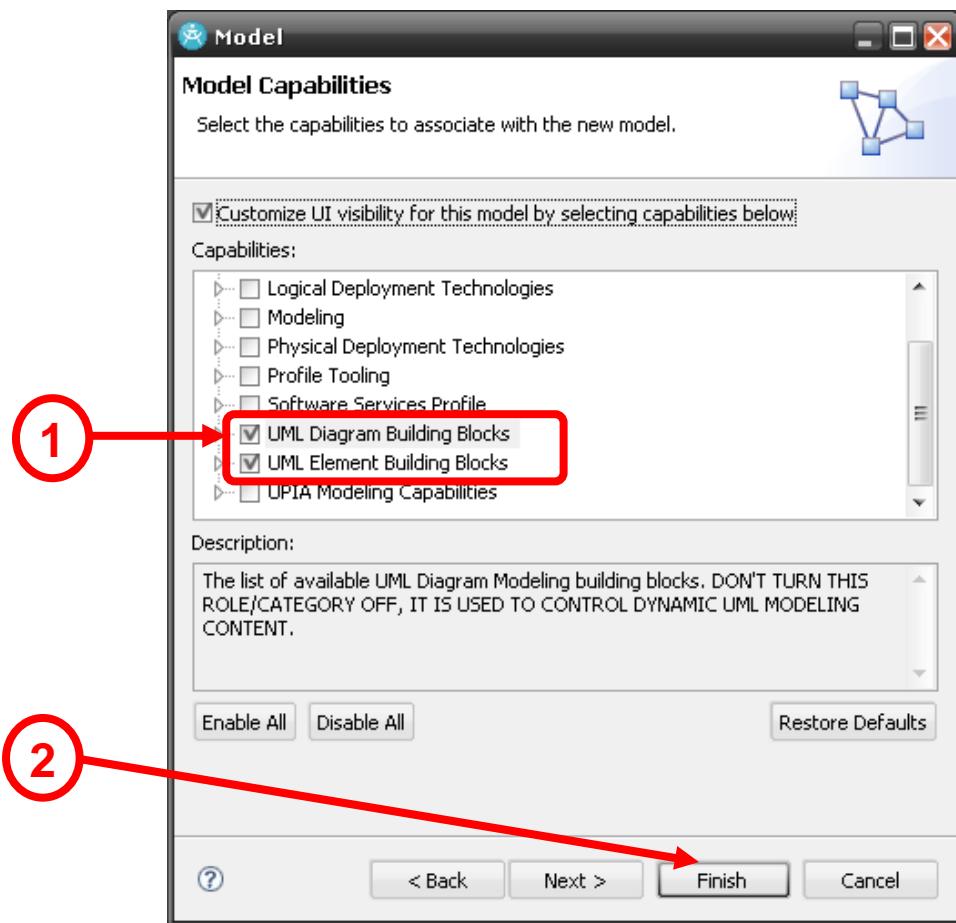
1. Seleccione crear modelo a partir del folder **Models**.



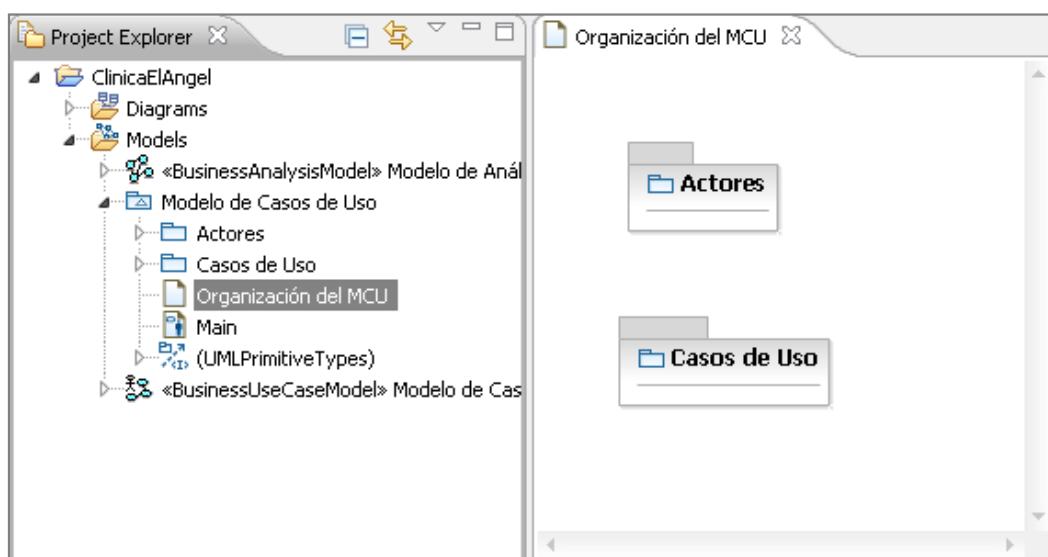
2. Para crear un Modelo de casos de uso en el cual se representa los requisitos funcionales del sistema, seleccione categoría **Requirements**.



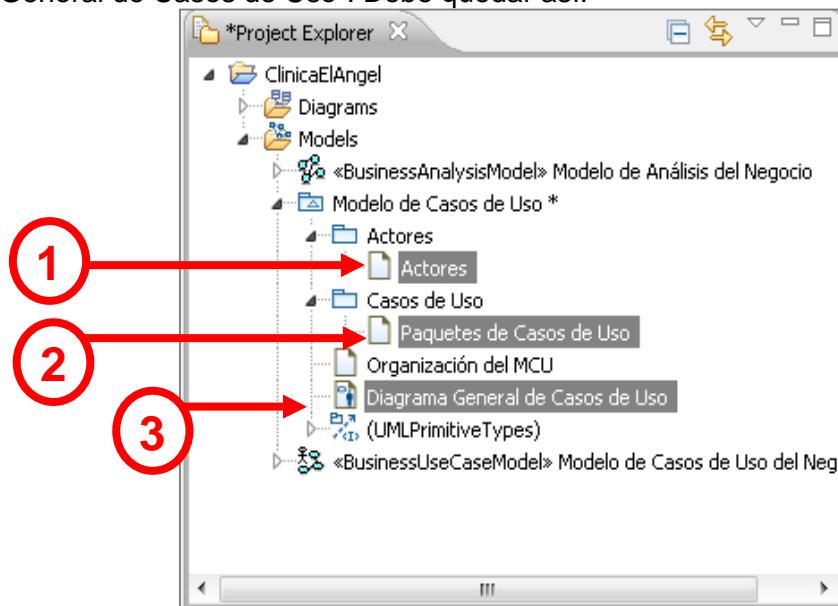
3. En la ventana de capacidades de modelo, active las dos opciones relacionadas a UML.



4. Agregue un diagrama de formato libre de nombre “Organización del MCU” y sobre él, agregue los paquetes del modelo.

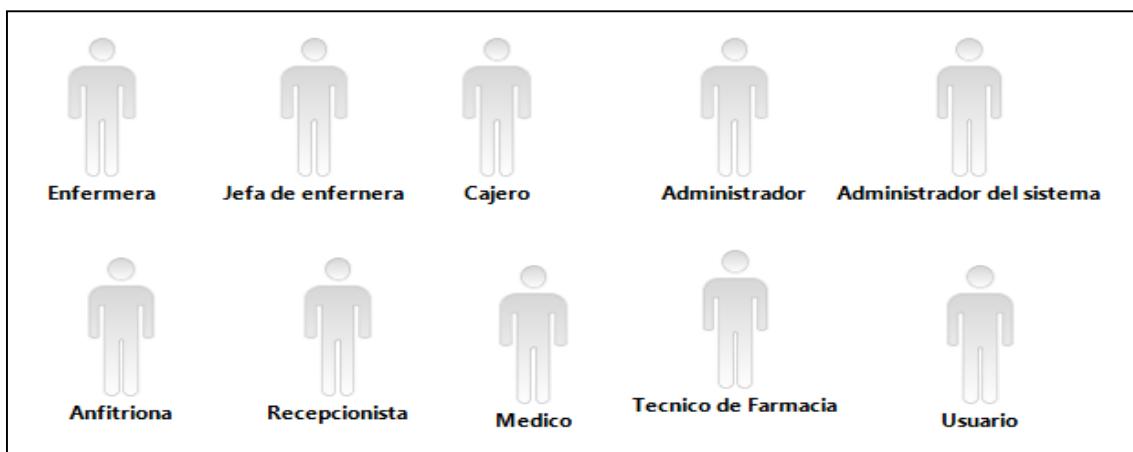
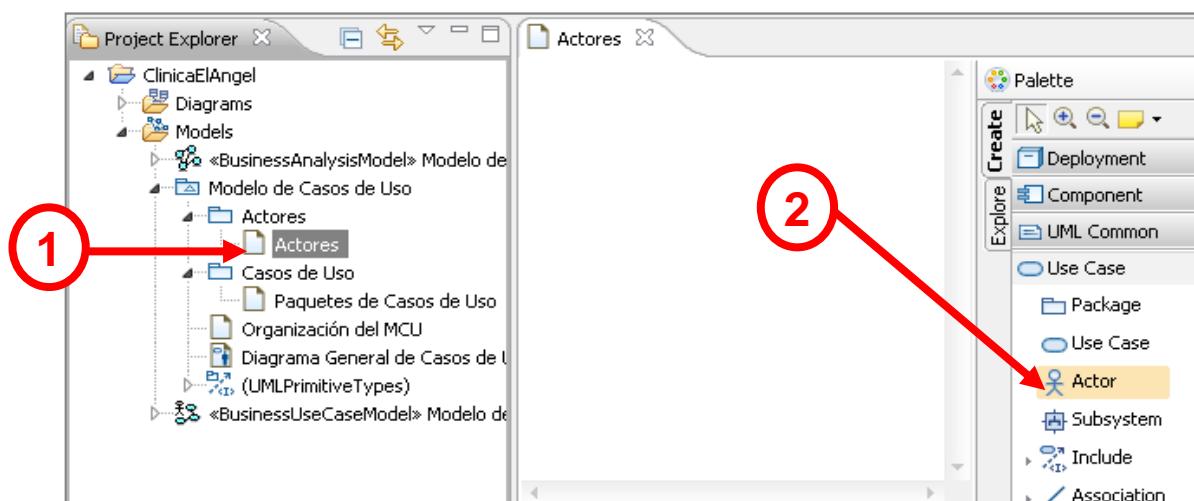


5. Luego, renombre los diagramas de cada paquete y el diagrama *Main* a “Diagrama General de Casos de Uso”. Debe quedar así:



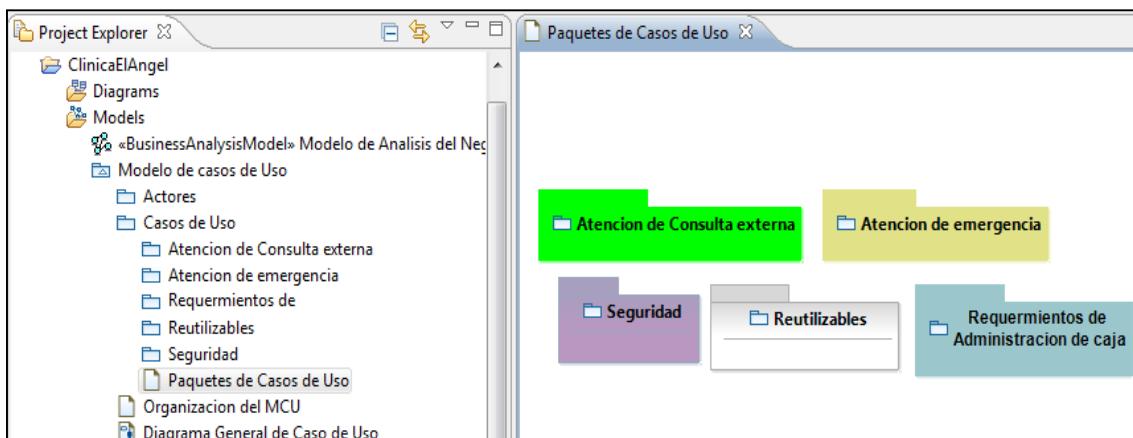
6. Ahora represente los roles (humanos o software) que interactuarán con el sistema.

**¿Cuáles son los actores identificados para el sistema?**



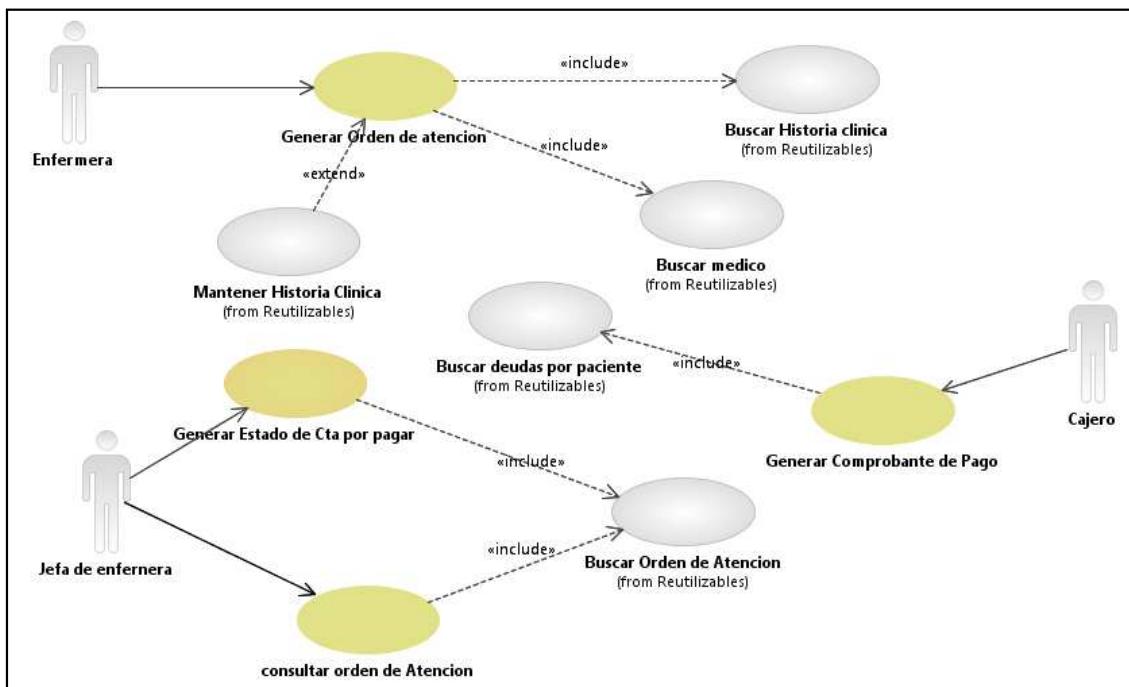
7. A continuación, crearemos los paquetes de casos de Uso. Primero los del Negocio, luego los Requerimientos solicitados (mantenimiento, Consultas y Reportes). Por último, el de seguridad. Se crearán los siguientes paquetes:

- Atención de Consulta externa (**Procesos de negocio**)
- Atención de emergencia(**Procesos de negocio**)
- Reutilizables
- Requerimientos de administración de caja
- Seguridad

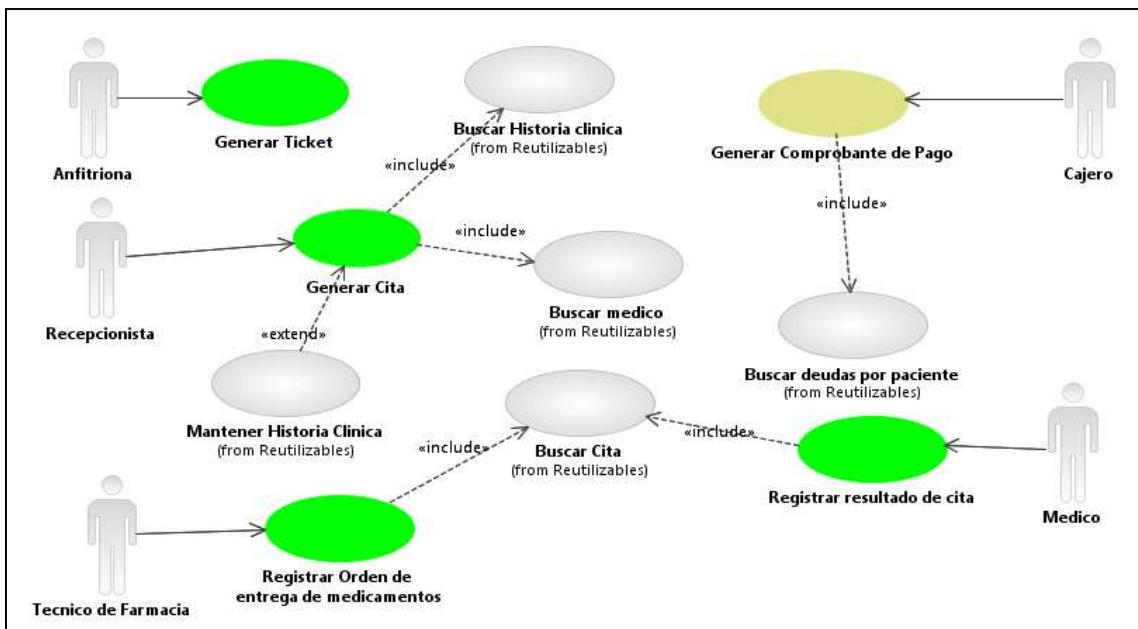


8. En cada paquete, desarrolle el diagrama de casos de uso, con los casos correspondientes

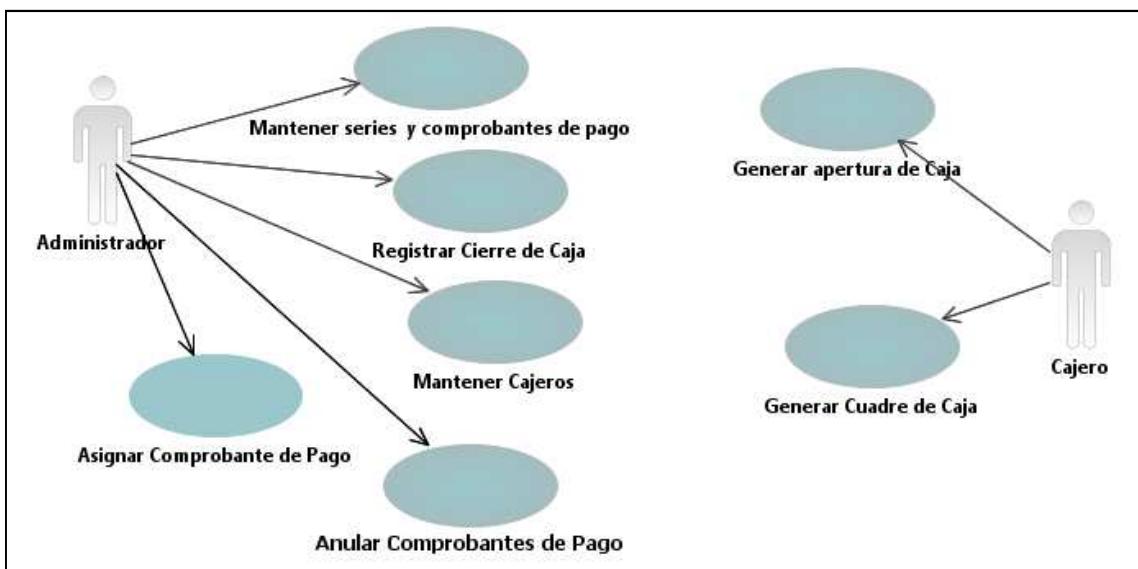
#### Paquete de Atención de emergencia



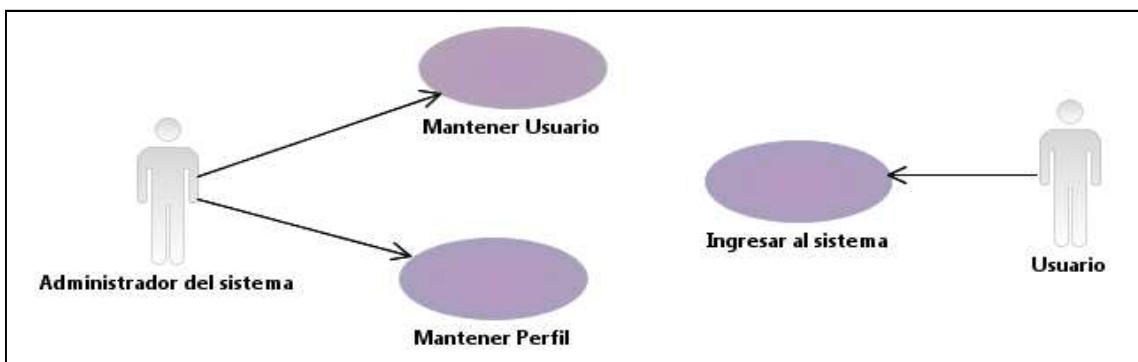
### Paquete de Atención de Consulta externa



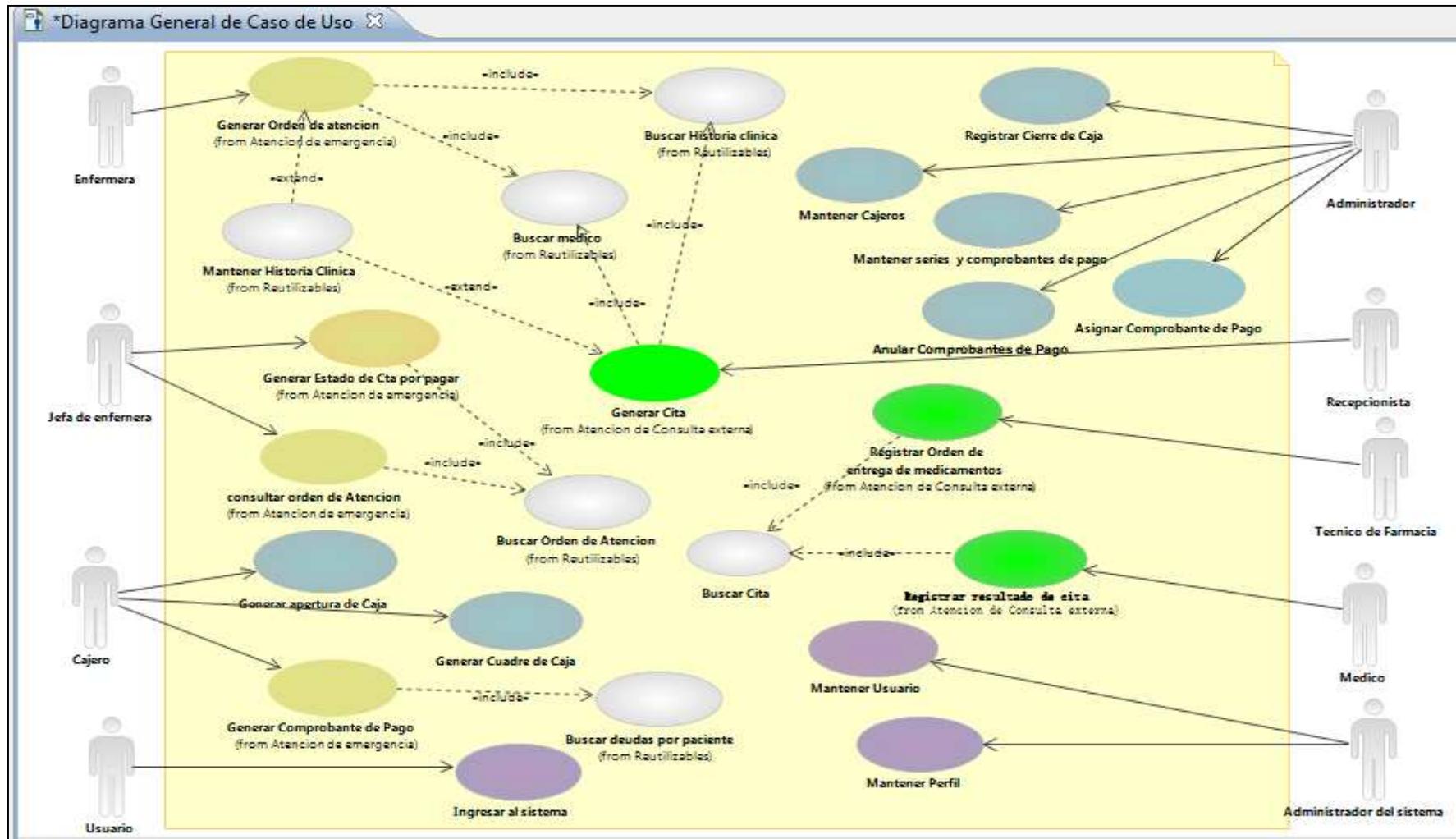
### Paquete de Caso de uso Requerimientos Administración de caja



### Paquete de seguridad

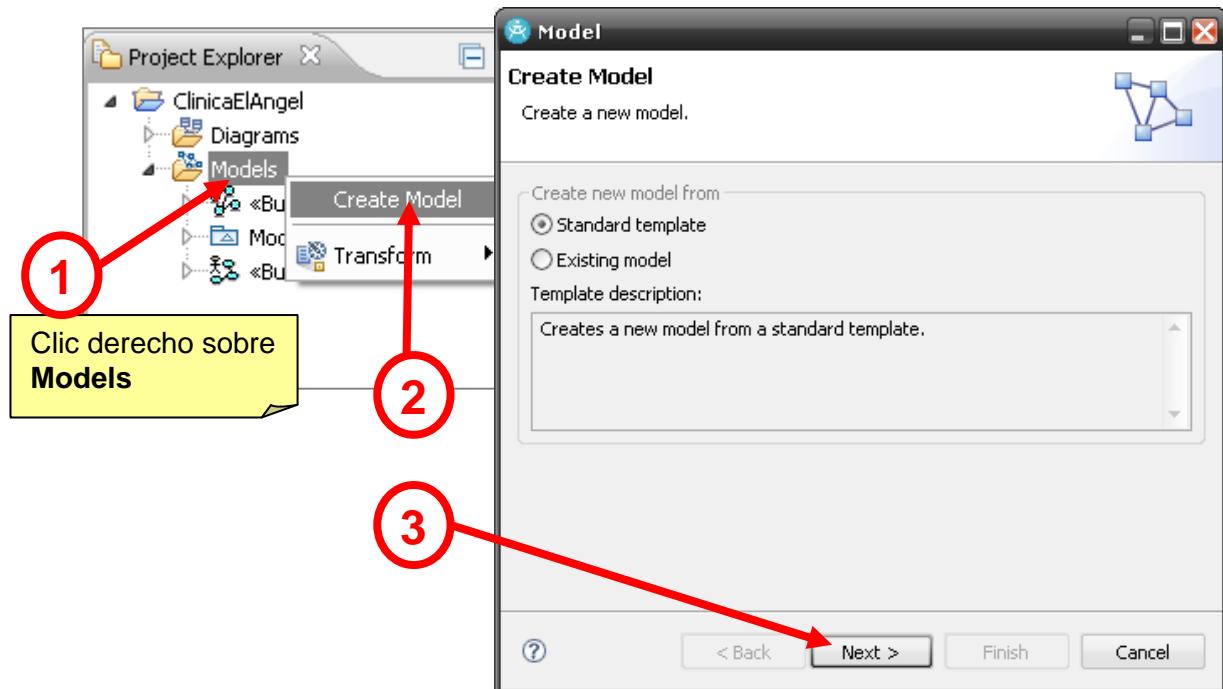


9. Por último, realice el Diagrama General de Casos de Uso

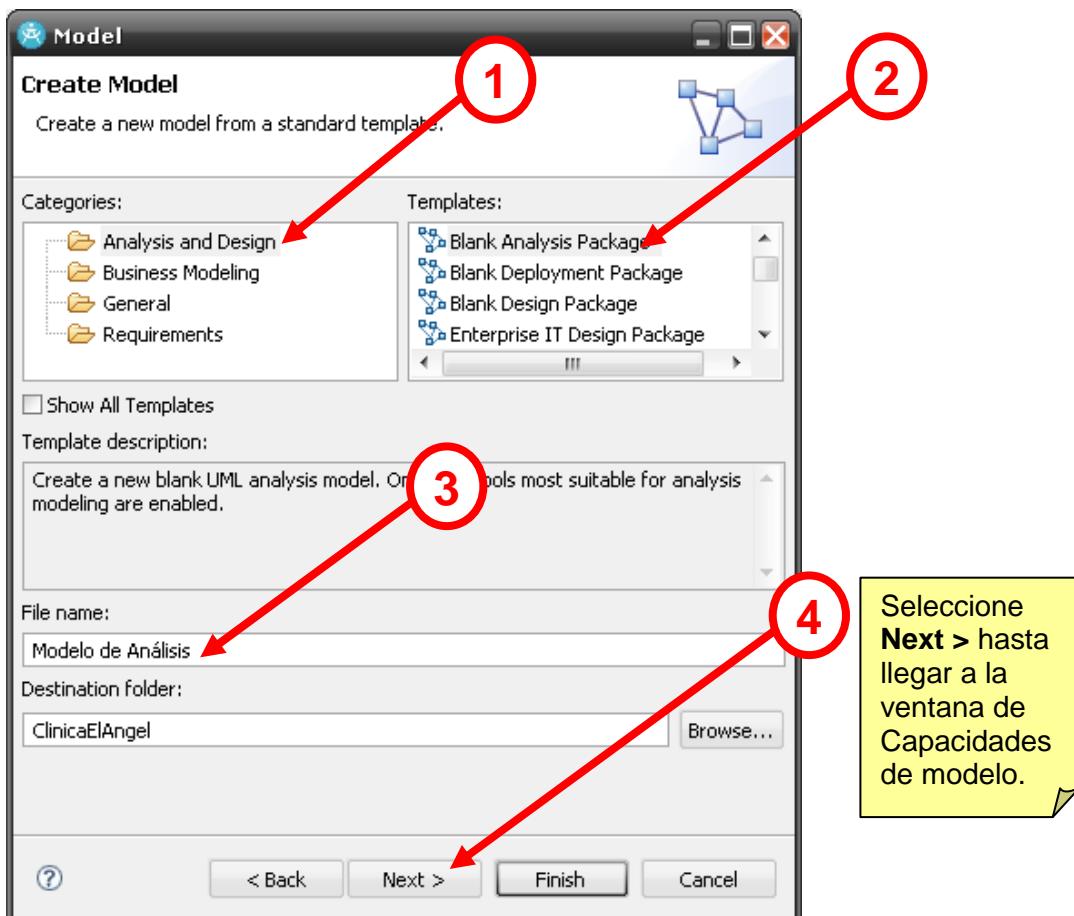


## Creación del Modelo de análisis

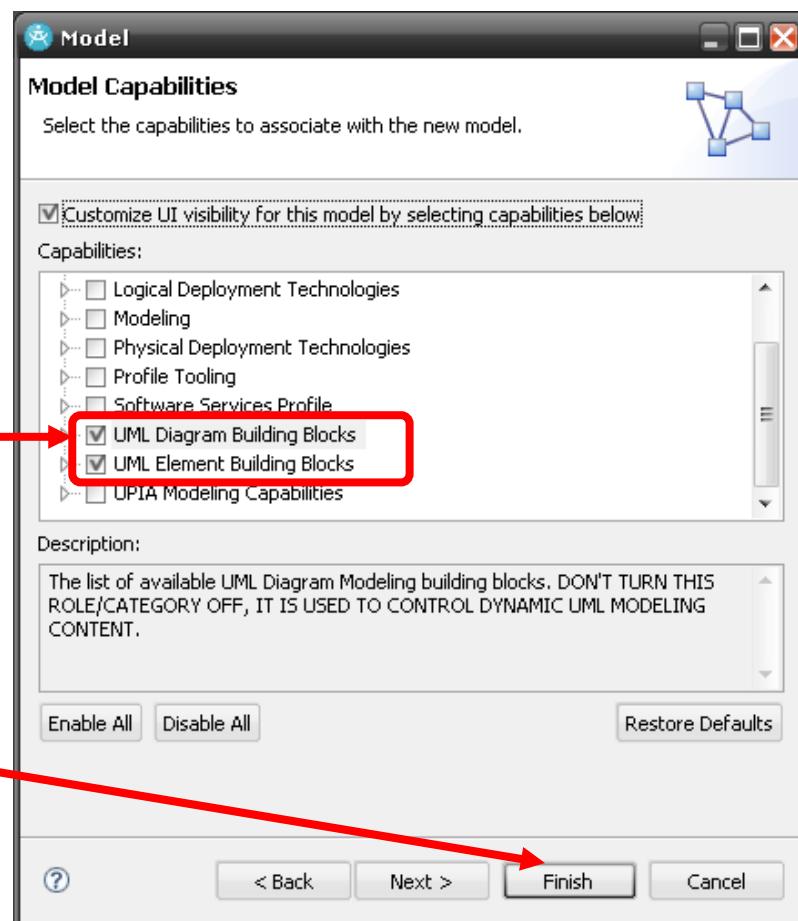
1. Seleccione crear modelo a partir del folder **Models**.



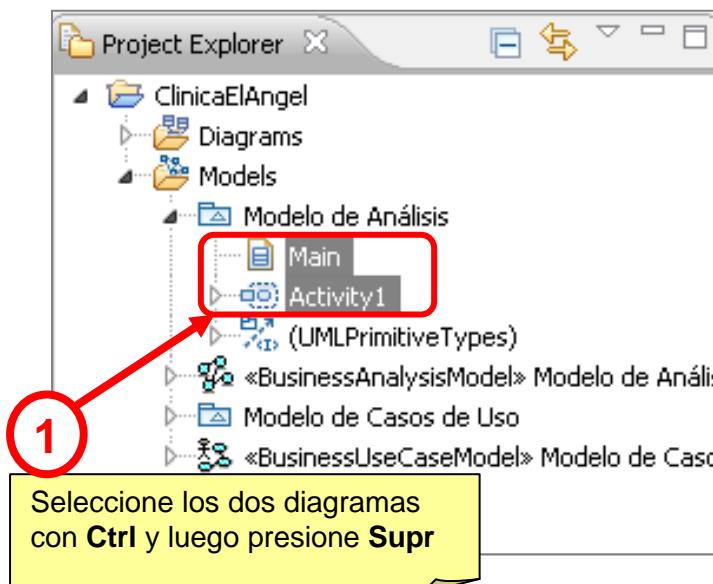
2. Para crear un Modelo de análisis en el cual se creará la arquitectura de análisis, seleccione categoría **Analysis and Design**.



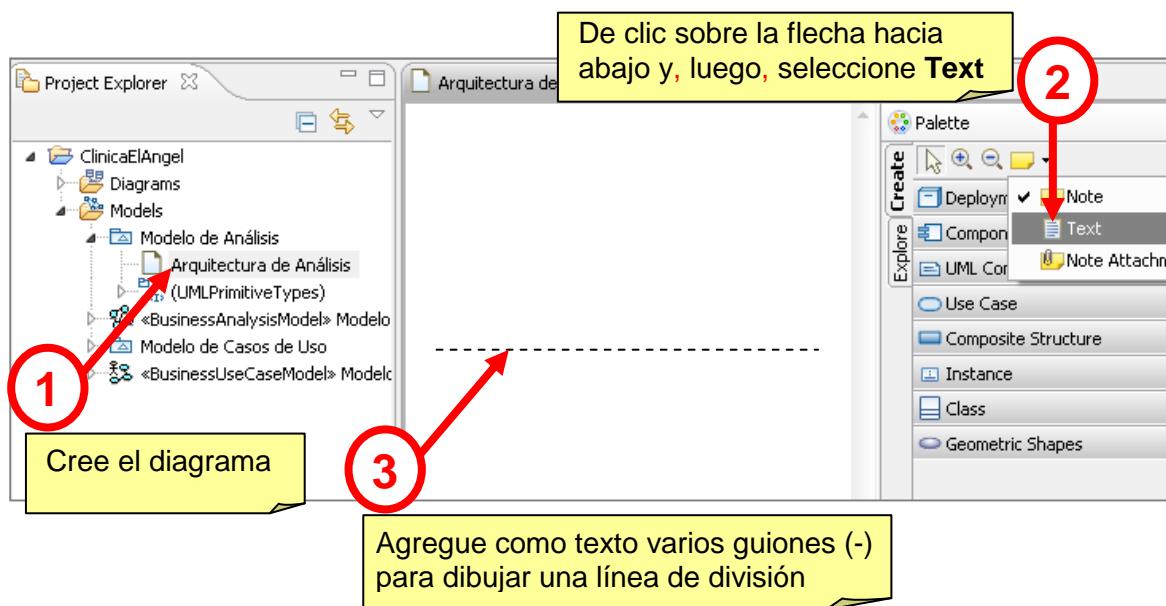
3. En la ventana de capacidades de modelo active las dos opciones relacionadas a UML.



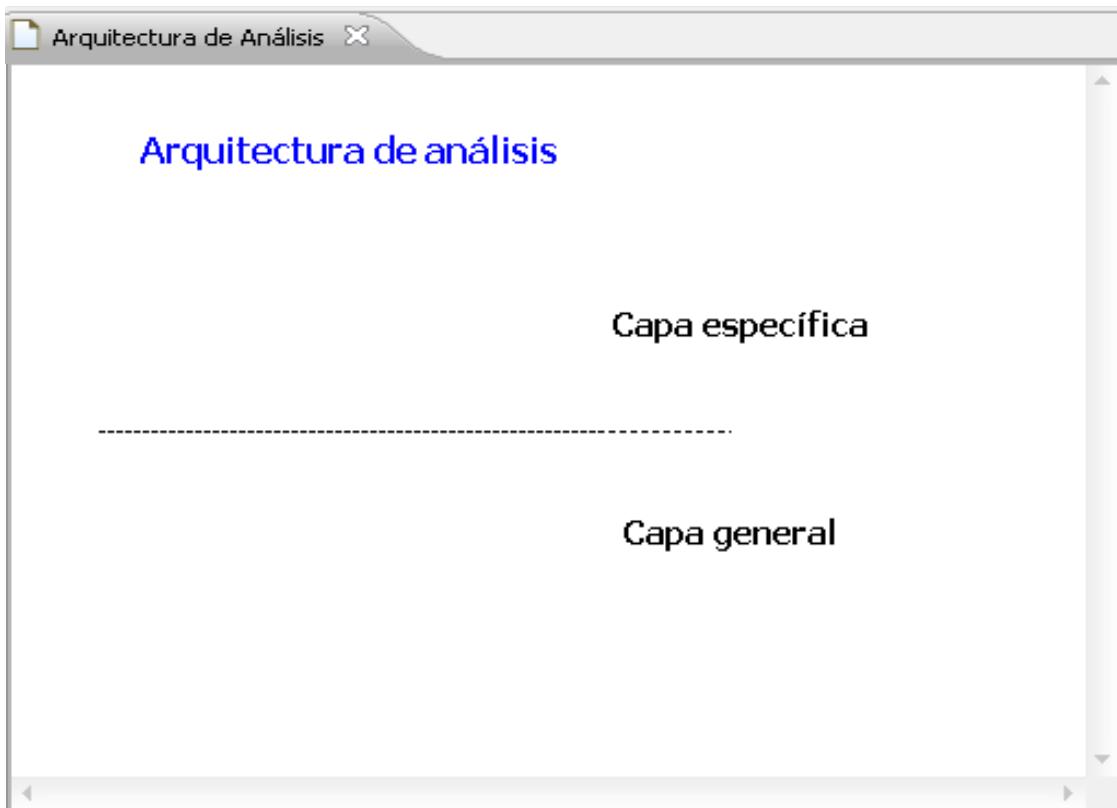
4. Elimine los dos diagramas que se crearon junto con el modelo.



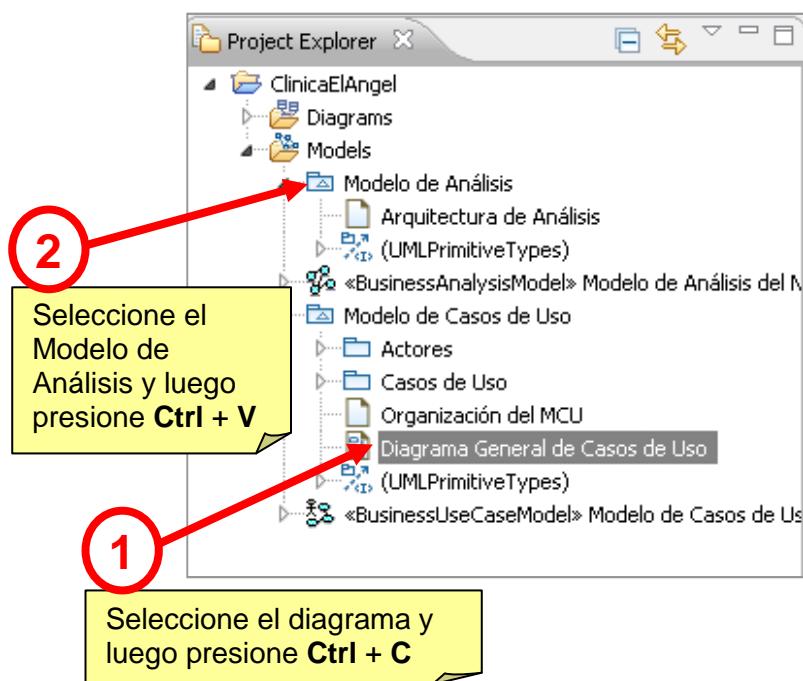
5. Agregue un diagrama de formato libre de nombre “Arquitectura de análisis” y sobre él, agregue una línea de separación para las capas general y específica.



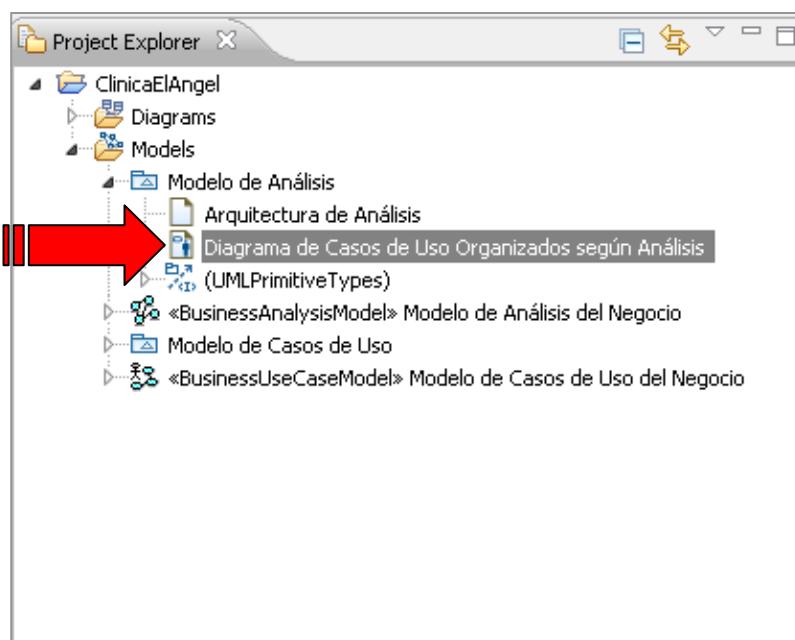
6. Ahora, agregue los títulos para cada capa con el elemento de texto.



7. Para realizar la arquitectura de análisis debemos, primero, analizar los casos de uso obtenidos en el modelo anterior. Así se procede a copiar el Diagrama general de casos de uso del modelo, de casos de uso al modelo de análisis.



8. A continuación, renombre el diagrama a “Diagrama de Casos de Uso Organizados según Análisis”. Luego, cambie a un solo color los casos de uso del diagrama.



9. Para realizar la organización de los casos de uso según análisis, debe utilizar algunos de los siguientes criterios.

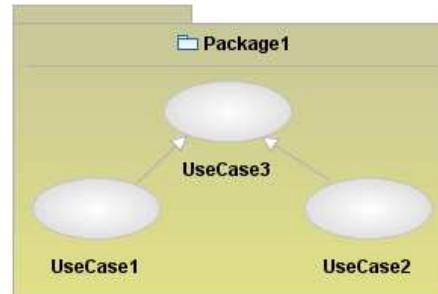
Entre las asignaciones adecuadas de casos de uso a un paquete en concreto se tiene lo siguiente:

1. Los casos de uso requeridos para dar soporte a un determinado proceso de negocio.
2. Los casos de uso requeridos para dar soporte a un determinado actor del sistema.

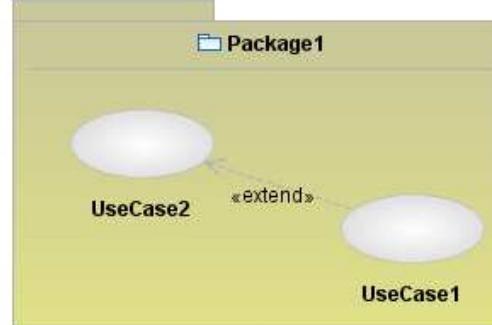
Para identificar los paquetes se basa en lo siguiente:

1. Tener un diagrama de casos de uso con los roles bien definidos.
2. Los casos de uso que estén bajo la responsabilidad de un actor deben tener contenidos estrechamente relacionados.

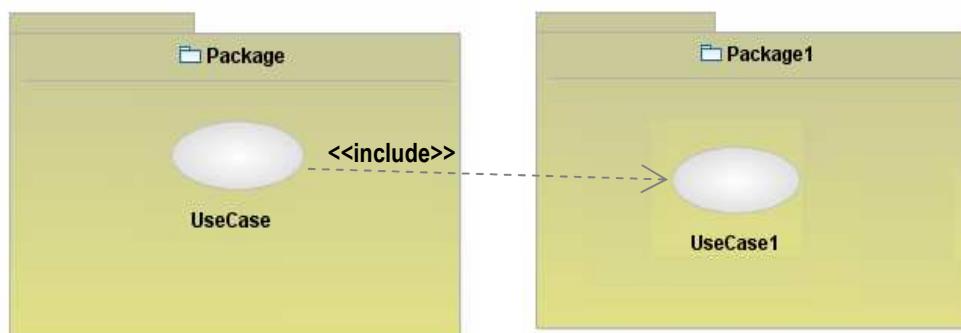
3. Los casos de uso que están relacionados mediante relaciones de generalización deben pertenecer al mismo paquete.

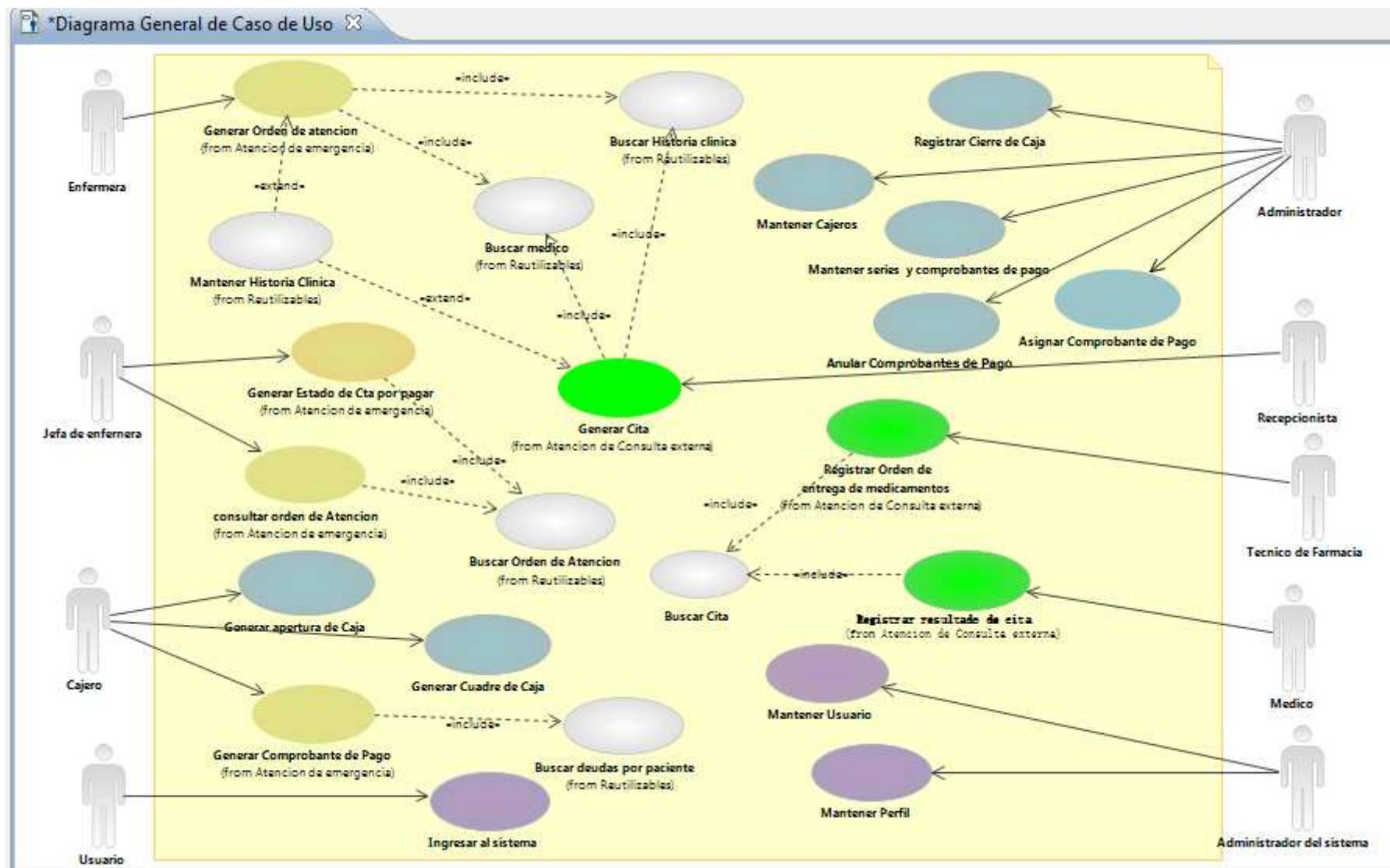


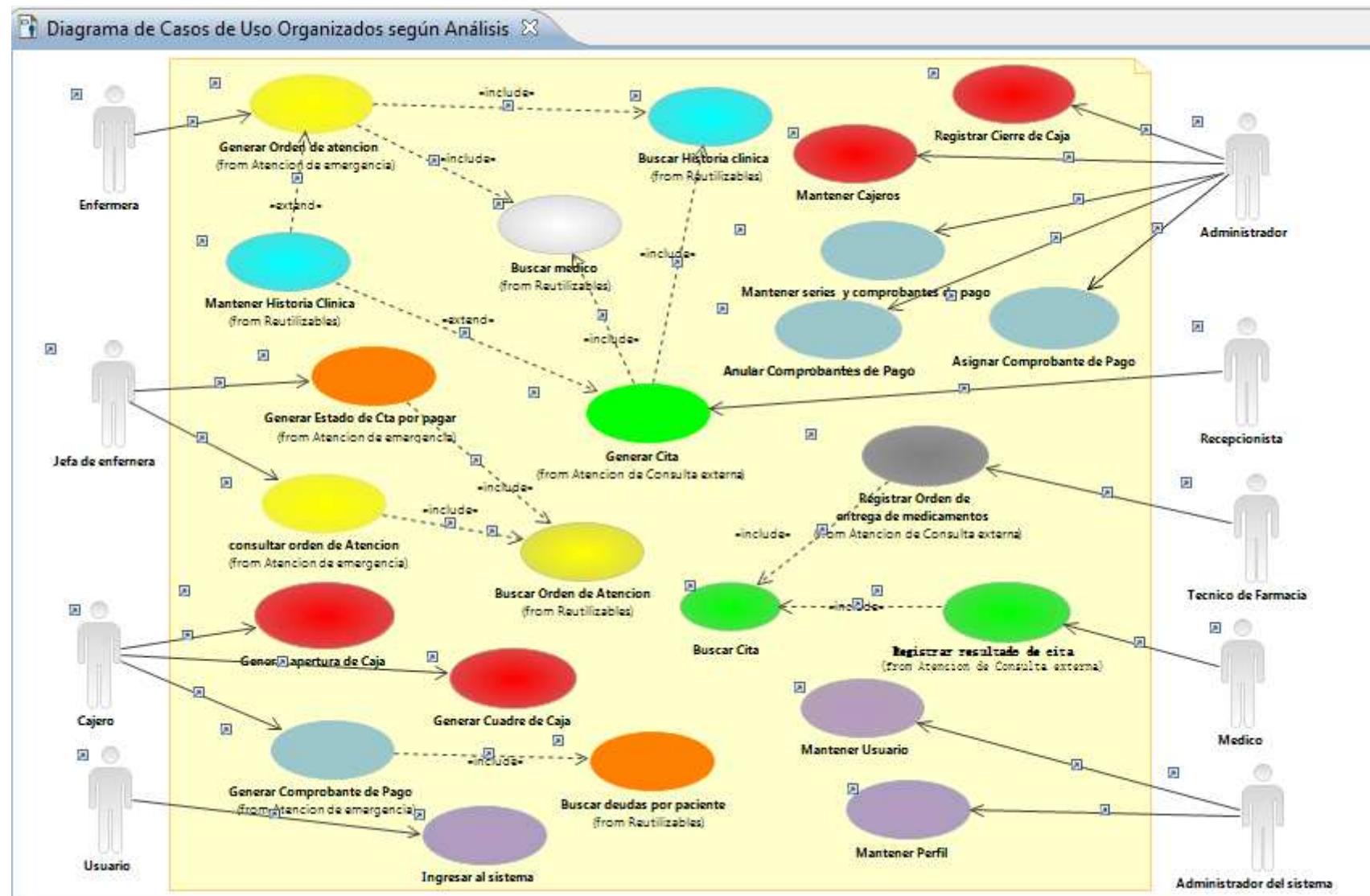
4. Los casos de uso relacionados mediante relaciones de extensión y solo se extienden a partir de un caso de uso base deben pertenecer al mismo paquete del caso de uso base.



5. Los casos de uso incluidos tienden a generar su propio paquete la mayor parte de veces. Si los casos de uso base, que incluyen al caso de uso, son funcionalidades con distintos contenidos; entonces, se debe crear un paquete para el caso de uso incluido.





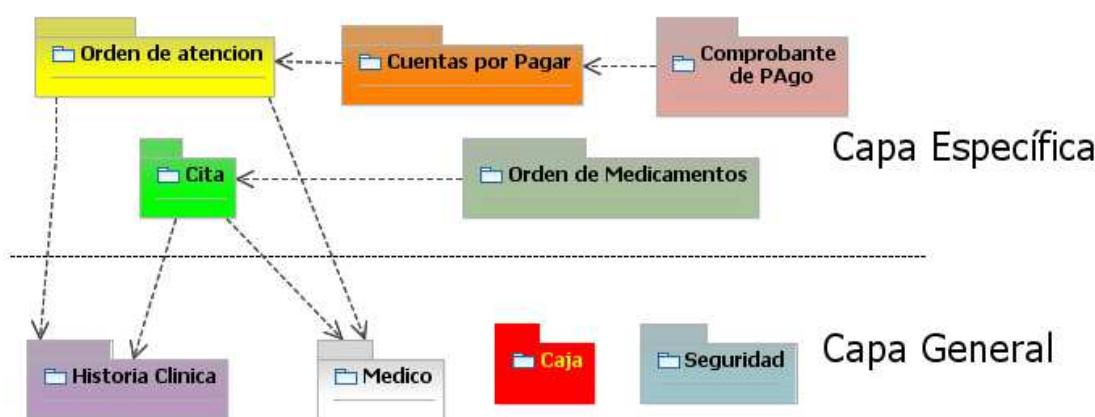


10. Una vez agrupado, por color, la nueva organización de casos de uso, complete el diagrama “Arquitectura de análisis” agregando los paquetes de análisis y sus dependencias.

**NOTA:**

Para identificar las dependencias entre paquetes, revise el diagrama de casos de uso según análisis, esto con el fin de ubicar las relaciones que existen entre los casos de uso. Las dependencias se crean a partir de los paquetes de análisis que contienen los casos de uso base.

### Arquitectura de Análisis (Capa de la Aplicación)



### ACTIVIDAD PROPUESTA

- Realice el Diagrama general de casos de uso organizados según análisis y la Arquitectura de análisis del siguiente caso. Traerlo para la próxima clase.

La empresa SECURITY S.A. tiene en su equipo comercial la difícil tarea de obtener contratos de servicios de Seguridad. Esta tarea se inicia cuando un cliente se pone en contacto con la empresa; por ende, es atendido por un vendedor quien le genera un presupuesto, verificando previamente si el cliente se encuentra registrado. Si no se encuentra lo registra. Cuando el cliente está de acuerdo con el presupuesto, el Supervisor de Ventas registra un contrato, realizando una búsqueda de Presupuestos. Todos los contratos son entregados al Gerente General para su evaluación quién registra el resultado de la evaluación, previamente realiza una búsqueda de contratos.

El Gerente General entrega copias de los contratos al Departamento de Cobranza, la secretaría de cobranza emite los comprobante de pagos (facturas), previamente realiza una búsqueda de contratos. Cada viernes la secretaría asigna un cobrador a cada comprobante emitido, realizando una búsqueda de comprobantes. Al final del día, registrará el pago de los comprobantes, realizando una búsqueda de comprobantes para realizar la cancelación del comprobante. La secretaría de cobranza puede adicionalmente actualizar los datos de los clientes como teléfono, correo, dirección, etc.

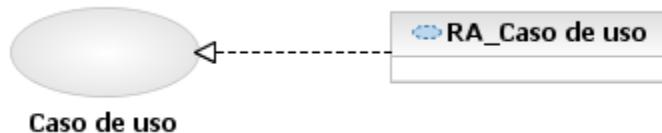
# Resumen

- ॥ Para el caso propuesto “Clínica EL Ángel”, se crearon los siguientes modelos:
  - Modelo de casos de uso del negocio
  - Modelo de análisis del negocio
  - Modelo de casos de uso
  - Modelo de análisis
- ॥ El objetivo principal del Modelo de análisis es identificar los módulos del sistema, mientras que del Modelo de casos de uso es identificar las funcionalidades del sistema.
- ॥ Entre las asignaciones adecuadas de casos de uso a un paquete de análisis en concreto se tiene los siguientes criterios:
  1. Los casos de uso requeridos para dar soporte a un determinado proceso de negocio.
  2. Los casos de uso requeridos para dar soporte a un determinado actor del sistema.
- ॥ Se debe considerar las siguientes bases para identificar los paquetes de análisis:
  1. Tener un diagrama de casos de uso con los roles bien definidos.
  2. Los casos de uso que estén bajo la responsabilidad de un actor deben tener contenidos estrechamente relacionados. Los casos de uso que están relacionados mediante relaciones de generalización deben pertenecer al mismo paquete.
  3. Los casos de uso relacionados mediante relaciones de extensión y solo se extienden a partir de un caso de uso base deben pertenecer al mismo paquete del caso de uso base.
  4. Los casos de uso incluidos tienden a generar su propio paquete la mayor parte de veces. Si los casos de uso base que incluyen al caso de uso son funcionalidades con distintos contenidos, entonces, se debe crear un paquete para el caso de uso incluido.
- ॥ Para identificar las dependencias entre paquetes, se debe revisar el diagrama de casos de uso organizados según análisis. Las dependencias se crean a partir de los paquetes de análisis que contienen los casos de uso base.

## 2. REALIZACIONES DE ANÁLISIS DE CASOS DE USO

Representa la vista interna de un caso de uso. Incluye diagramas de clases y diagramas de interacción (comunicación o secuencia).

La razón para separar las realizaciones de casos de uso de los casos de uso es la administración independiente de estos artefactos. Para cada caso de uso, en el modelo de casos de uso, existe una realización de caso de uso en el modelo de análisis. La relación entre ambos, en UML, se realiza con *REALIZATION*. Así:



Las realizaciones de los casos de uso exige la identificación de clases de análisis, es decir, las clases de entidad, interfaz y control. Los estereotipos de estas clases se presentan a continuación:

CLASE DE ANÁLISIS	DESCRIPCIÓN
	Representa almacenamiento permanente de información.
	Representa interacciones con los actores (usuarios y otros sistemas externos).
	Representa el control de interacción entre clases.

### 1.1. Diagramas de clases de análisis

Podemos utilizar las siguientes normas generales para identificar las clases de análisis:

1. Identificar clases de entidad mediante el estudio en detalle de la especificación del caso de uso para considerar la información que éste utiliza y manipula.
2. Identificar una clase de interfaz por cada ventana con el cual interactúa un actor humano.
3. Identificar una clase de interfaz para cada actor que sea un sistema externo. Esta clase representará la interfaz de comunicación con el sistema externo.
4. Identificar una clase control responsable del tratamiento del control y de la coordinación de la realización del caso de uso. Asimismo, se debe considerar crear una clase control por cada paquete de análisis. Por lo tanto, todos los casos de uso de un mismo paquete compartirán la misma clase control.

Conforme identifique las clases, las debe ir reubicando en sus paquetes de origen. Para crear los diagramas de clases debe considerar la siguiente relación:

**ACTOR - CLASE INTERFAZ - CLASE CONTROL - CLASE ENTIDAD**

De lo anterior se afirma que....:

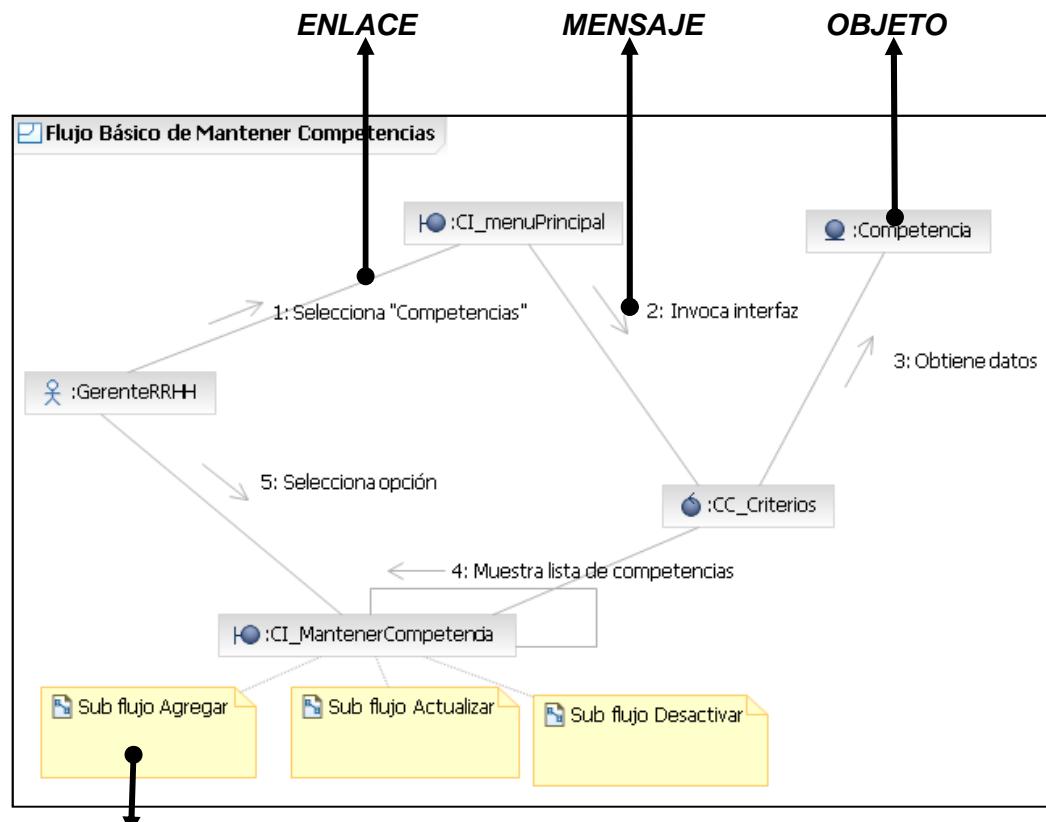
- Un actor solamente se puede comunicar con una clase interfaz
- Una clase interfaz únicamente se puede comunicar con una clase control
- Una clase control es el nexo entre una clase interfaz y una clase entidad. Por lo tanto, una clase control se puede comunicar con una clase interfaz y una entidad.
- Una clase interfaz no se puede comunicar con una clase entidad

## 1.2. Diagramas de comunicación

El diagrama de comunicación es un tipo de diagrama de interacción; en esta etapa, no se usa diagramas de secuencia, porque no es importante la cronología de las interacciones.

Un diagrama de comunicación muestra la colaboración dinámica entre los objetos, es decir, describe el comportamiento de un caso de uso mostrando explícitamente las relaciones de los objetos participantes.

Los elementos que se utilizan en un diagrama de comunicación son los siguientes:



**NOTAS QUE HACEN REFERENCIA A OTROS DIAGRAMAS DE COMUNICACIÓN**

## CASO DE ESTUDIO

Siguiendo con el caso de la sesión anterior, utilizaremos la especificación de un caso de uso de mantenimiento para crear su realización de análisis.

### ESPECIFICACIÓN DE CASO DE USO: Mantener Cajero

#### 1. Descripción

El caso de uso permite mantener actualizado el registro de los cajeros de la clínica. De acuerdo a su necesidad, el Administrador de la Clínica puede agregar, actualizar y desactivar un cajero.

#### 2. Actor(es)

Administrador.

#### 3. Flujo de Eventos

##### 3.1. Flujo Básico

1. El caso de uso se inicia cuando el Administrador selecciona la opción "Cajeros" en la interfaz del menú principal.
2. El sistema muestra la interfaz "MANTENER CAJERO" con la lista de cajeros con los campos: código, nombres, apellido paterno, apellido materno, teléfono, correo, dirección, fecha de registro, fecha de actualización y estado. Además, muestra las opciones: **Agregar Cajero**, **Actualizar Cajero** y **Desactivar Cajero**.
3. Si el Administrador elige un cajero
  - a. Si elige "Actualizar" ver el Subflujo Actualizar Cajero.
  - b. Si elige "Desactivar" ver el Subflujo Desactivar Cajero.
4. Si el Administrador NO elige un cajero
  - a. Si elige "Agregar" ver el Subflujo Agregar Cajero.
5. El Administrador selecciona "Salir" y el caso de uso finaliza.

##### 3.2. Subflujos

###### 3.2.1. Agregar Cajero

1. El sistema muestra la interfaz CAJERO con los siguientes campos: código (solo lectura), nombres, apellido paterno, apellido materno, teléfono, correo, dirección, fecha de registro (solo lectura) y fecha de actualización (solo lectura). Además, muestra las opciones: **Aceptar** y **Cancelar**.
2. El Administrador ingresa los datos del Cajero.
3. El Administrador selecciona la opción Aceptar.
4. El sistema valida los datos ingresados.
5. El sistema genera un nuevo código de cajero y obtiene la fecha del sistema para la fecha de registro y la fecha de actualización
6. El sistema graba un nuevo registro de cajero y muestra el MSG "Cajero creado con código Nro. 999999".
7. El Administrador cierra la interfaz CAJERO y regresa a la interfaz MANTENER CAJERO con la lista de cajeros actualizada y el subflujo finaliza.

###### 3.2.2. Actualizar Cajero

1. El sistema muestra los datos del cajero seleccionada en la interfaz CAJERO: código (solo lectura), nombres, apellido paterno, apellido materno, teléfono, correo, dirección, fecha de registro (solo lectura) y

fecha de actualización (solo lectura). Además muestra las opciones: **Aceptar y Cancelar.**

2. El Administrador actualiza los datos del cajero.
3. El Administrador selecciona la opción Aceptar.
4. El sistema valida los datos ingresados del cajero.
5. El sistema obtiene la fecha del sistema para la fecha de actualización, actualiza el registro de cajero y muestra el MSG "Cajero actualizado satisfactoriamente".
6. El Administrador cierra la interfaz CAJERO y regresa a la interfaz MANTENER CAJERO con la lista de cajeros actualizada y el subflujo finaliza.

### **3.2.3. Desactivar Cajero**

1. El sistema muestra el MSG: "¿Está seguro que desea desactivar el(los) cajero(s) seleccionado(s)?".
2. El Administrador selecciona la opción YES para confirmar la desactivación.
3. El sistema actualiza el registro del(los) cajero(s) en estado "Desactivado".
4. El sistema muestra la interfaz MANTENER CAJERO con la lista de cajeros actualizada y termina el subflujo.

## **3.3. Flujos Alternativos**

### **1. Datos del Cajero Inválidos**

Si los datos ingresados son nulos o inválidos, tanto en los subfusos Agregar como en Actualizar Cajero, el sistema muestra el MSG: "Se han encontrado datos inválidos" y los subfusos continúan en el paso 2.

### **2. Cajero ya existe**

Si el sistema detecta que el cajero ya existe en el paso 4 del subfuso Agregar Cajero, muestra el MSG: "Cajero ya existe" y el subfuso finaliza.

### **3. No confirma Desactivación**

Si el Administrador selecciona NO en el paso 2 del subfuso Desactivar Cajero, finaliza el subfuso.

## **4. Precondiciones**

1. El Administrador está identificado en el sistema.
2. Lista disponible de Cajeros.

## **5. Poscondiciones**

1. En el sistema quedará registrado el nuevo Cajero.
2. En el sistema quedará actualizado el registro del Cajero.
3. En el sistema quedará desactivado el Cajero.

## **6. Puntos de Extensión**

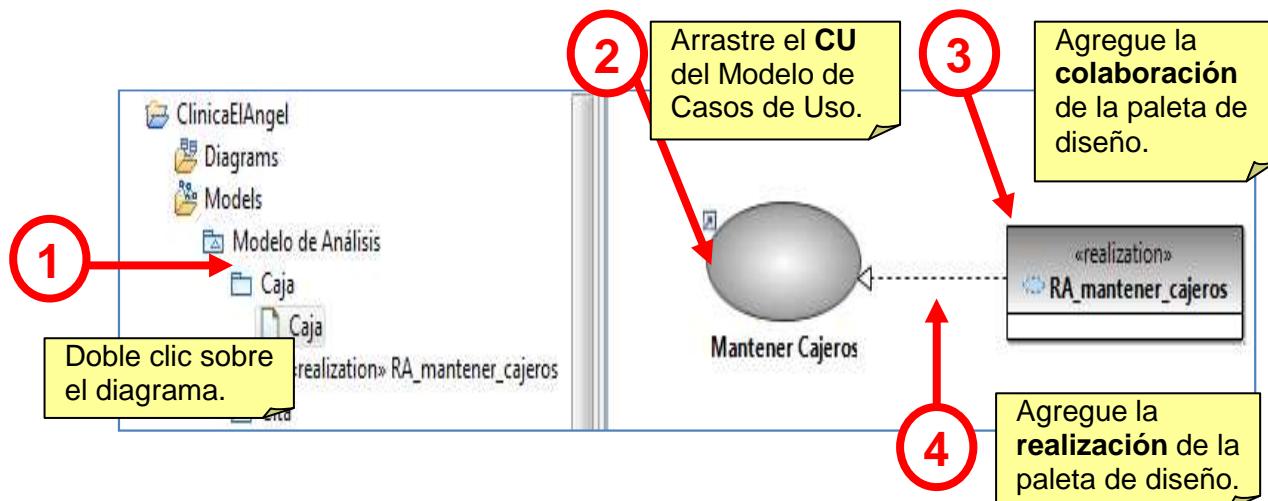
Ninguno.

## **7. Requisitos Especiales**

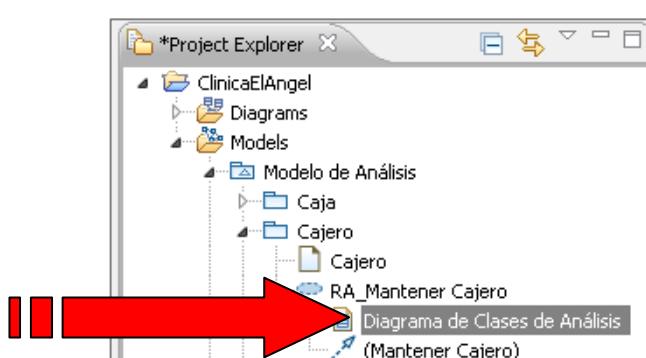
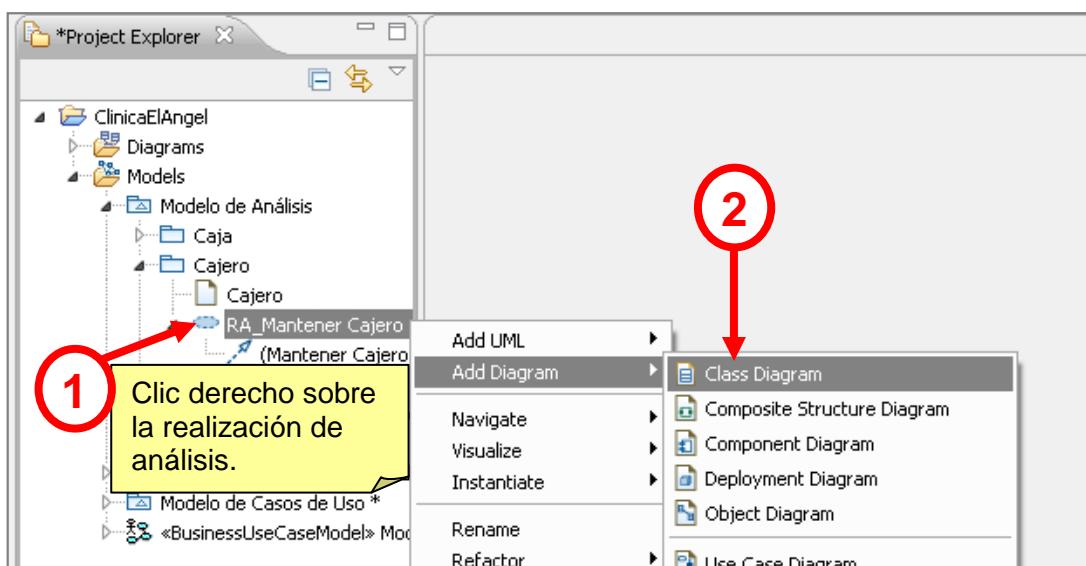
Ninguno.

## Creación de la realización de análisis de casos de uso

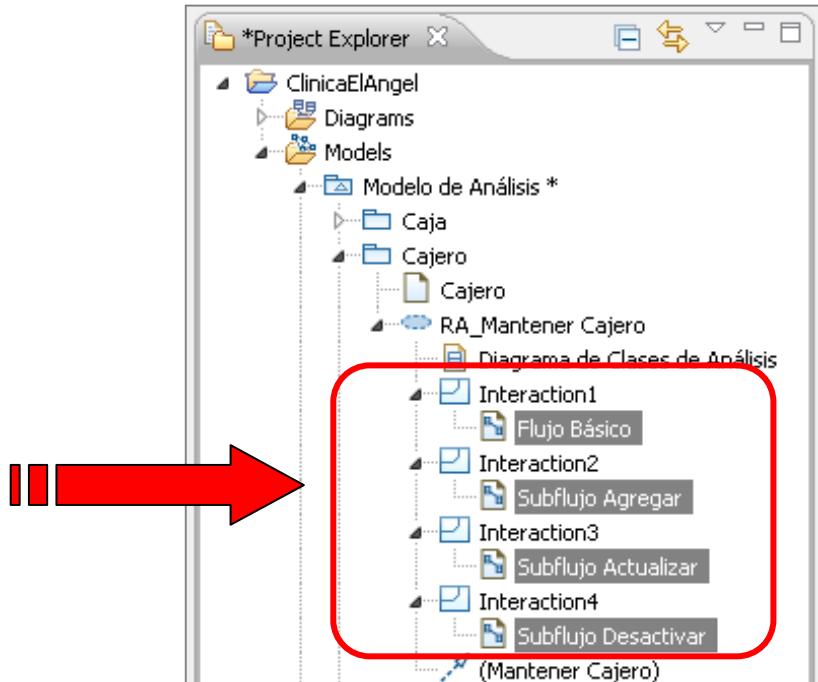
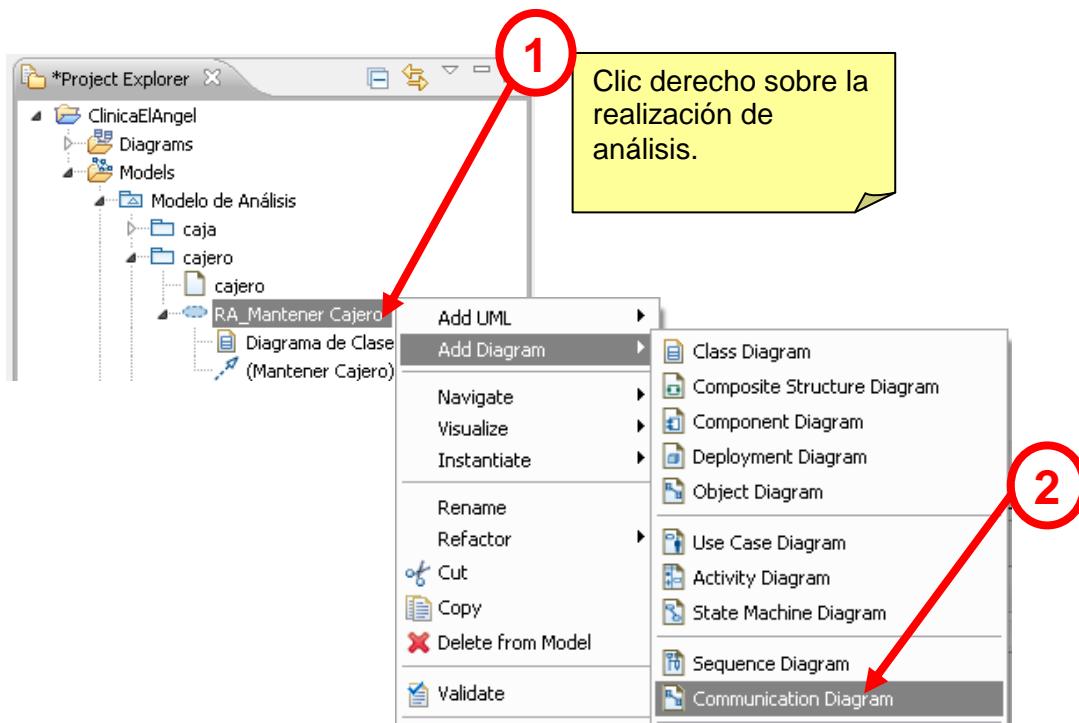
- Sobre el diagrama de formato libre del paquete de análisis **Caja**, represente la realización de análisis del caso de uso.



- Desde el explorador de proyectos, agregue un diagrama de clases con el nombre "Diagrama de Clases de Análisis".

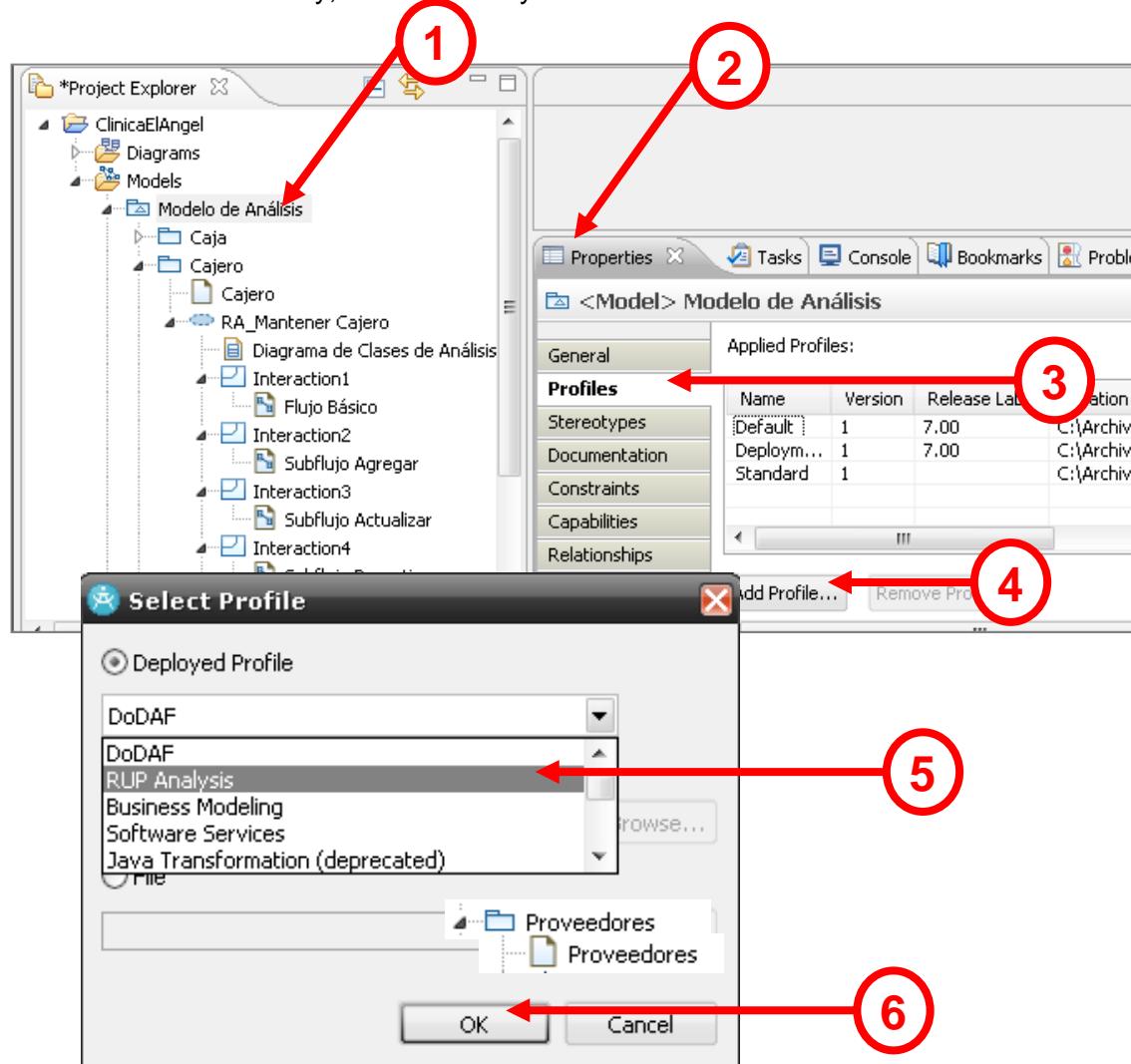


3. A continuación, agregue los diagramas de comunicación para el flujo básico y subflujos.

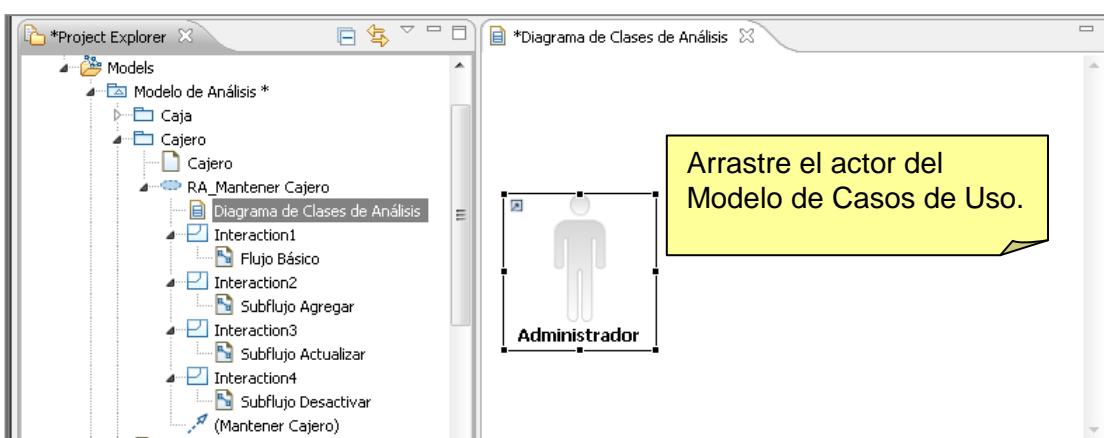


## Creación del Diagrama de clases de análisis

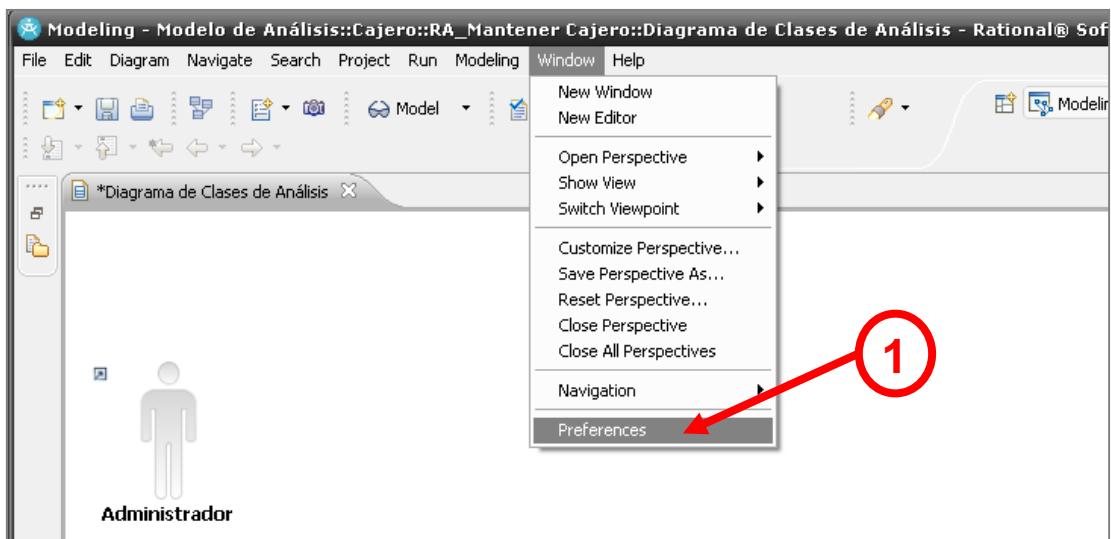
1. Agregue el perfil *RUP Análisis*. Este perfil contiene los estereotipos de las clases de análisis: boundary, control y entity.



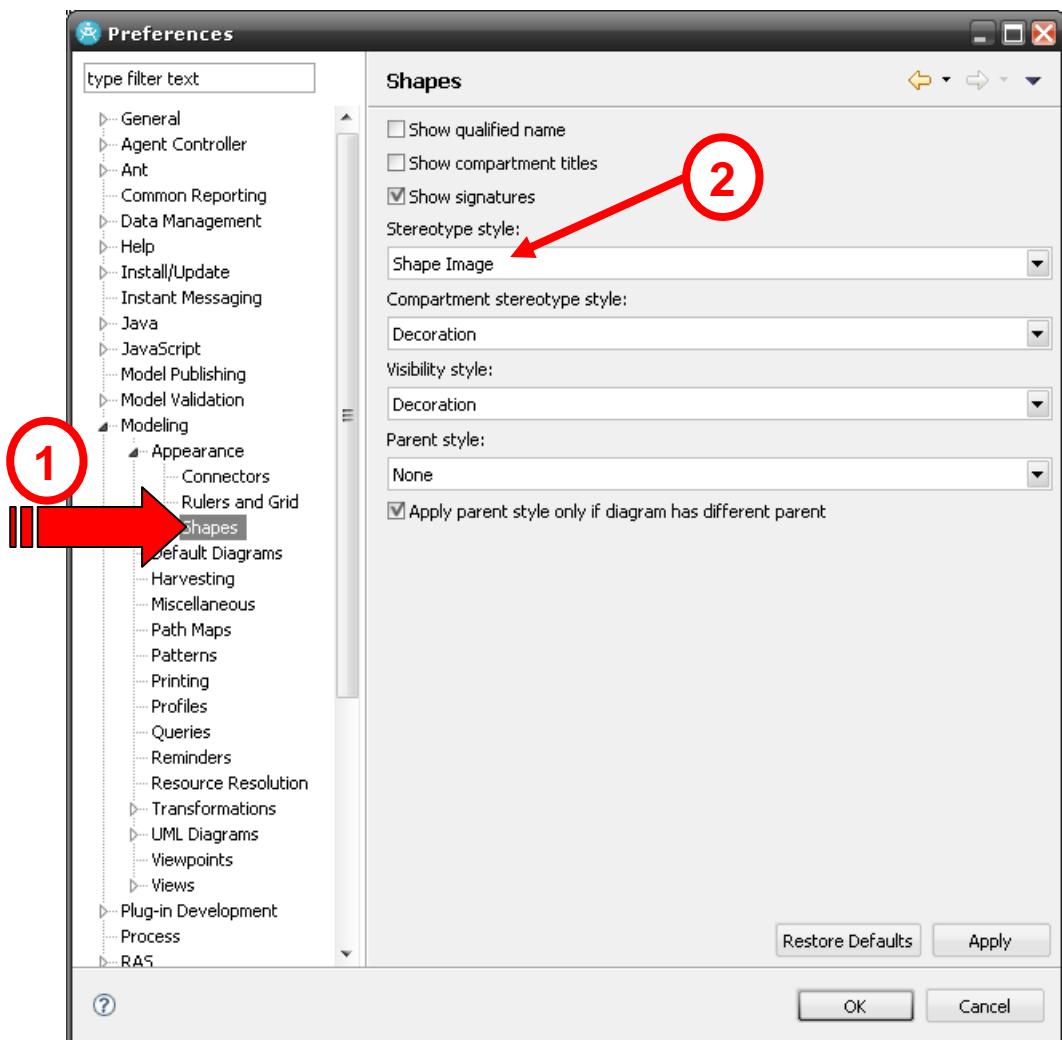
2. Ahora realice el diagrama de clases de análisis. Empiece arrastrando el actor sobre el diagrama.



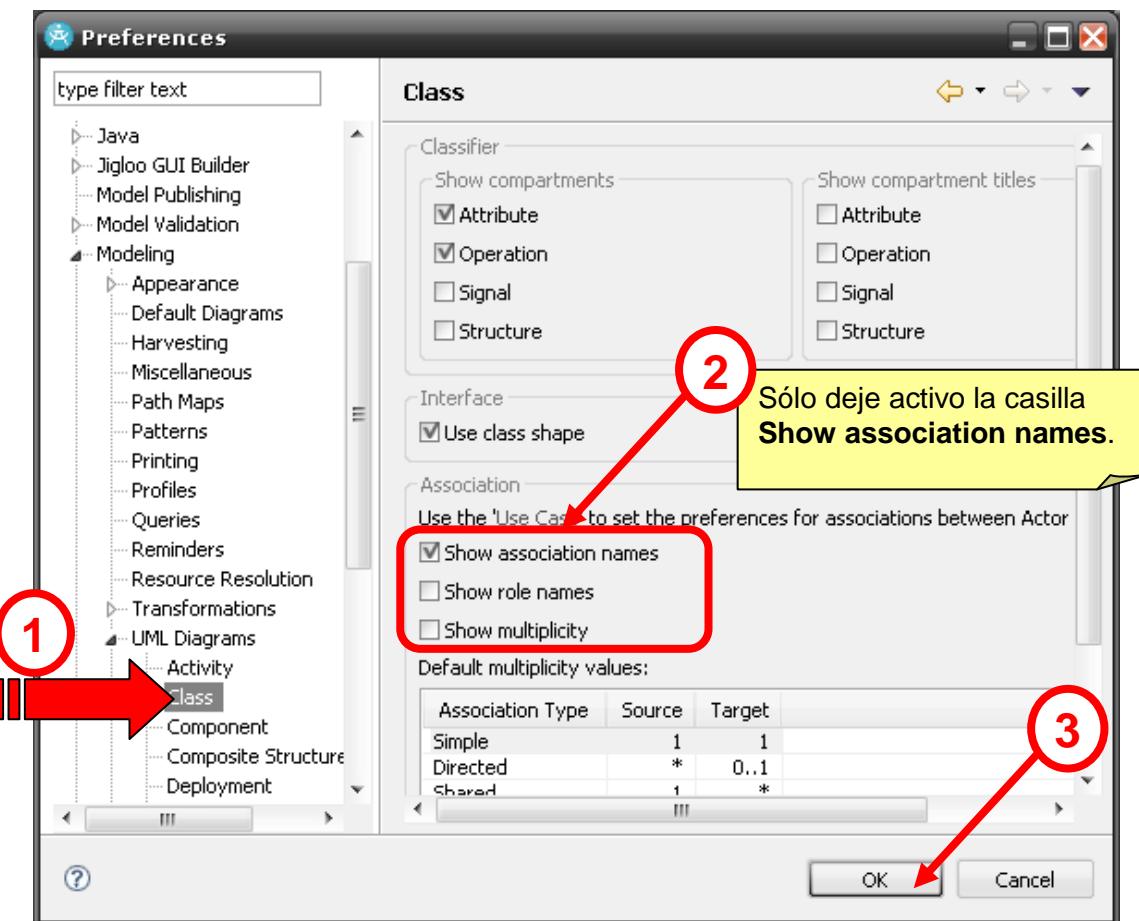
3. Configure las preferencias para crear el diagrama de clases.



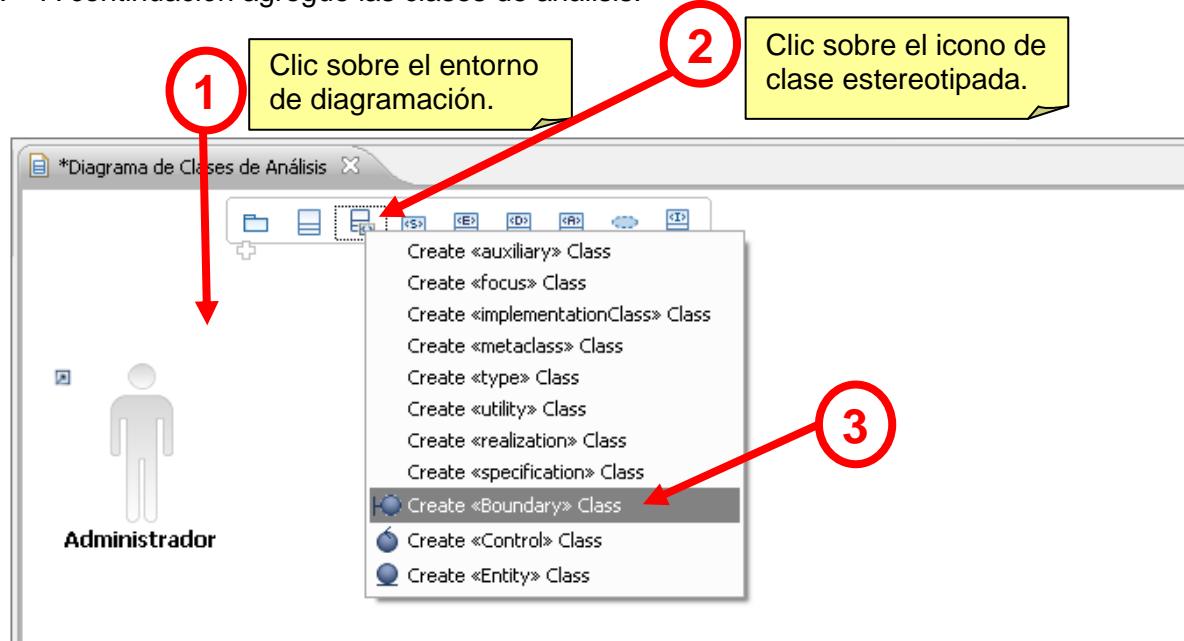
3.1. Primero configure la apariencia de las clases a forma de imagen del estereotipo:



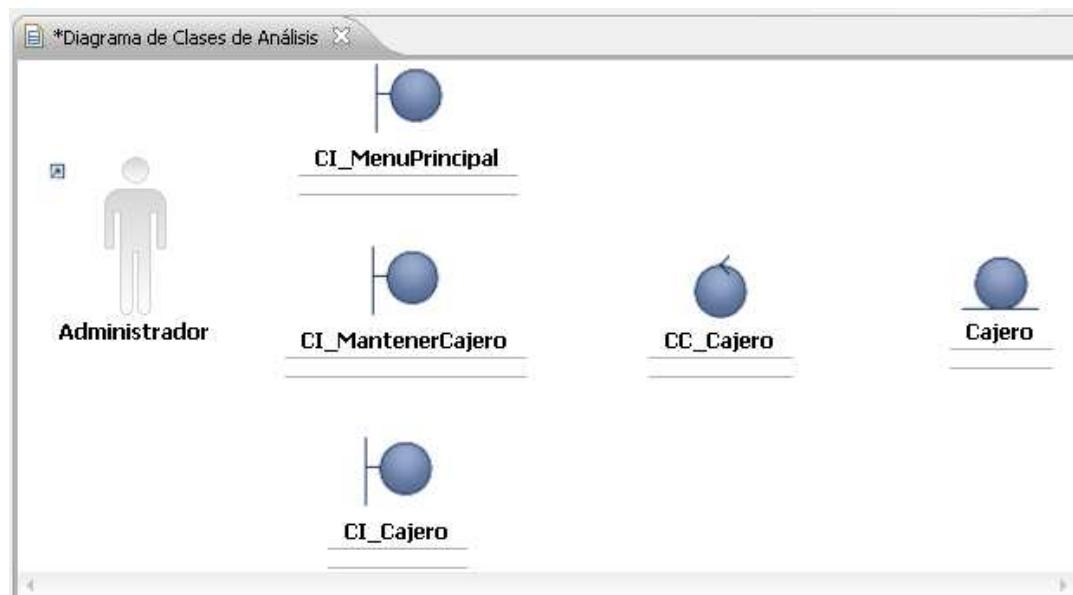
3.2. Luego, configure las asociaciones:



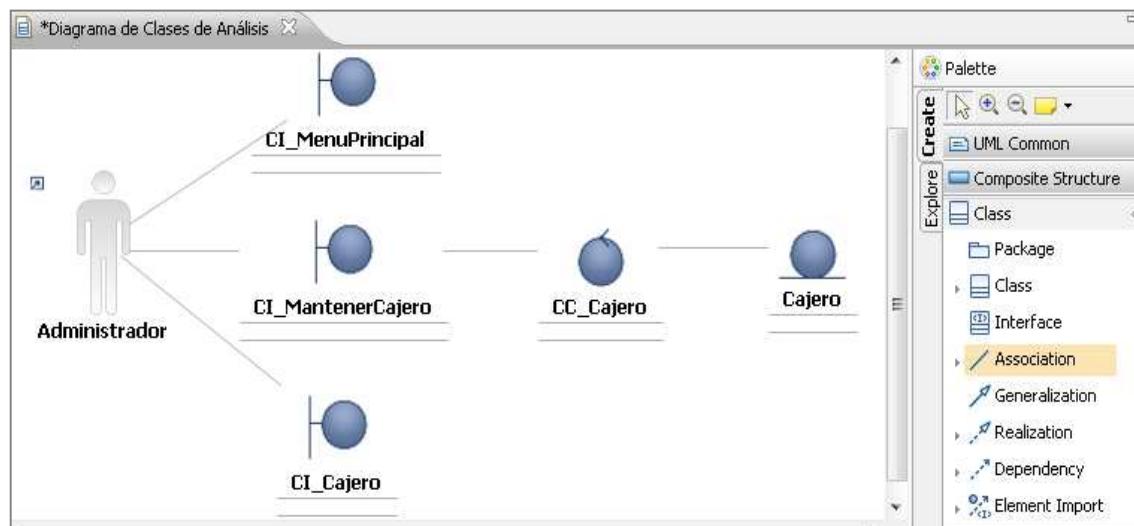
4. A continuación agregue las clases de análisis.



5. A partir de la ECU se identificaron las siguientes clases.

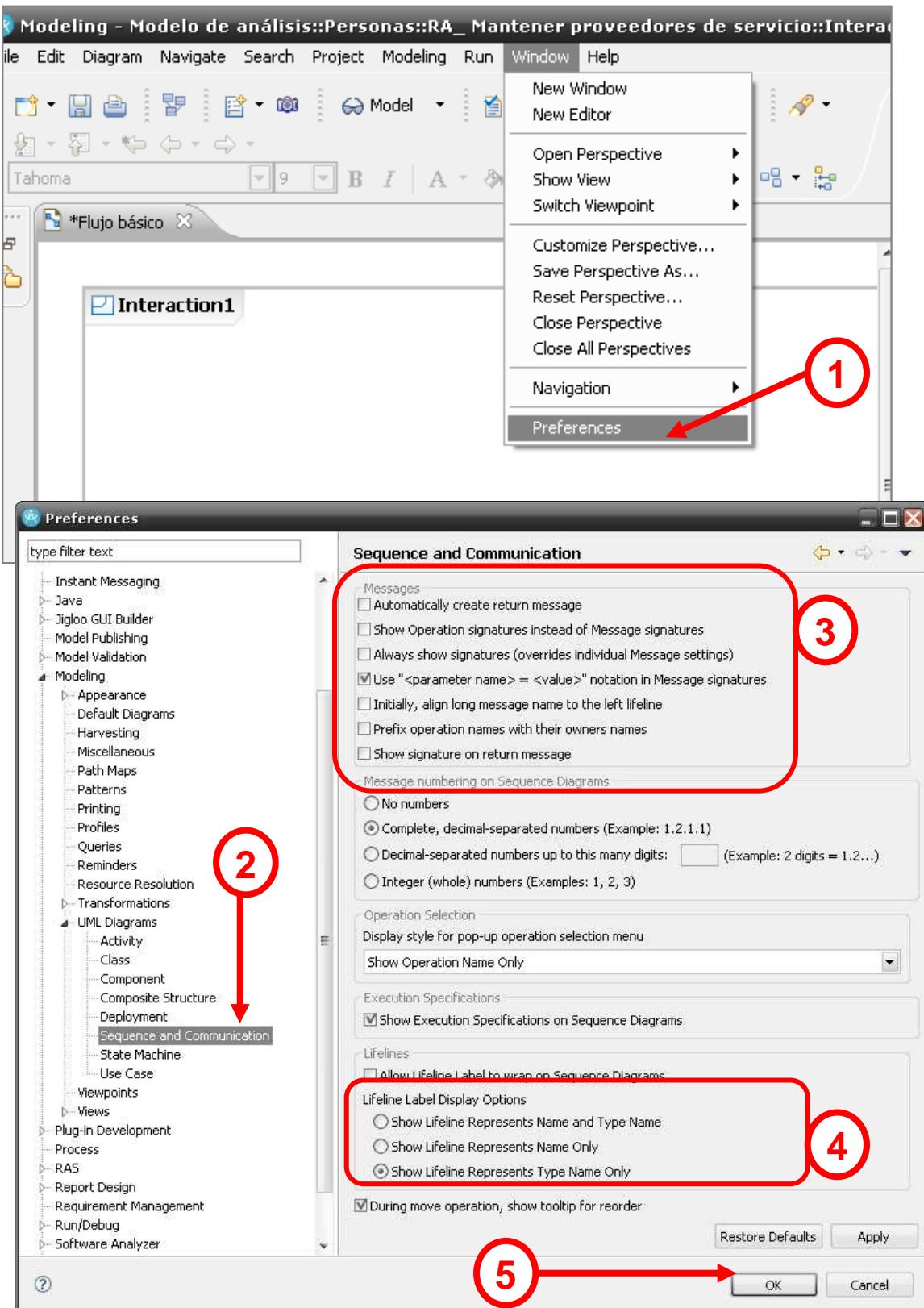


6. Por último, agregue las asociaciones entre las clases.



## Creación de Diagramas de comunicación

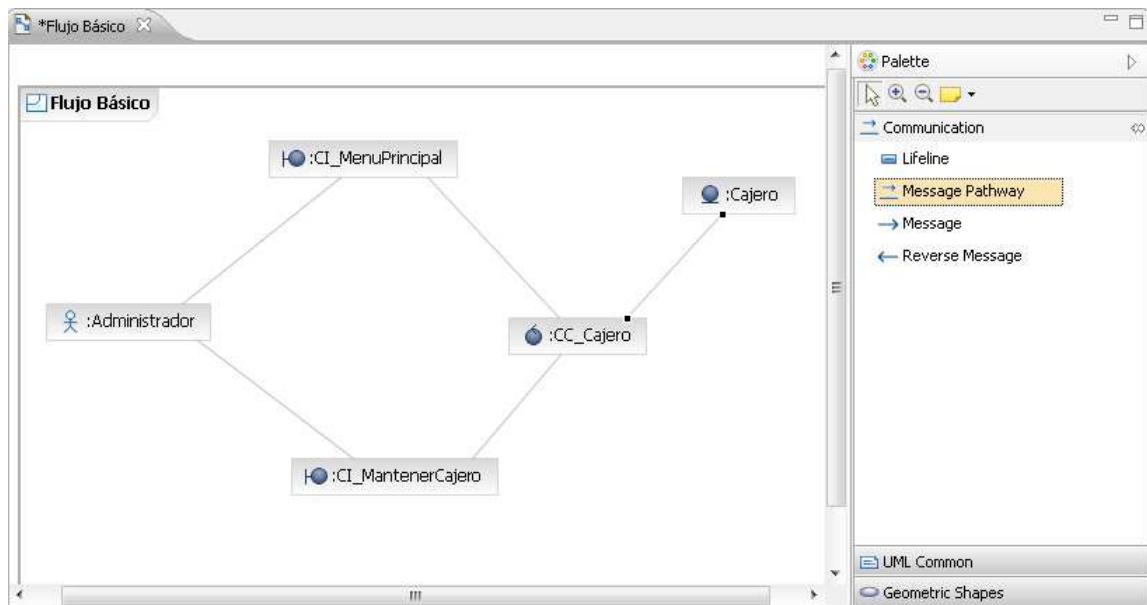
- Configure las preferencias de diagramas de comunicación.



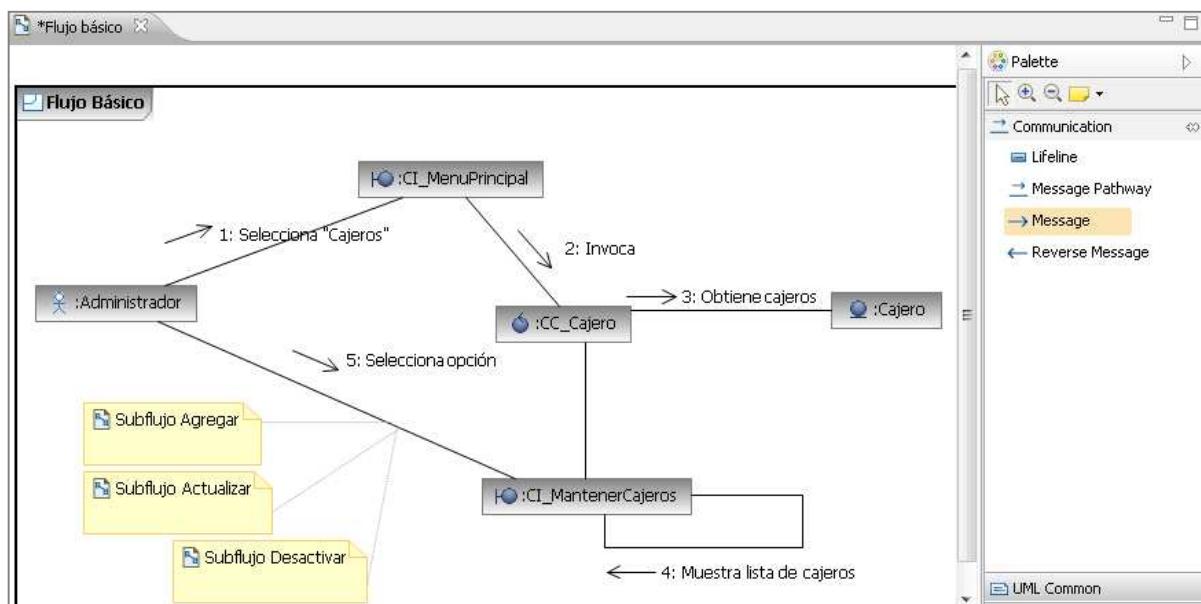
2. Empezaremos con la diagramación del flujo básico. Para ello, arrastre los objetos que participan en el flujo básico y agregue los enlaces (*Message Pathway*).

La dirección de los enlaces debe ser de...

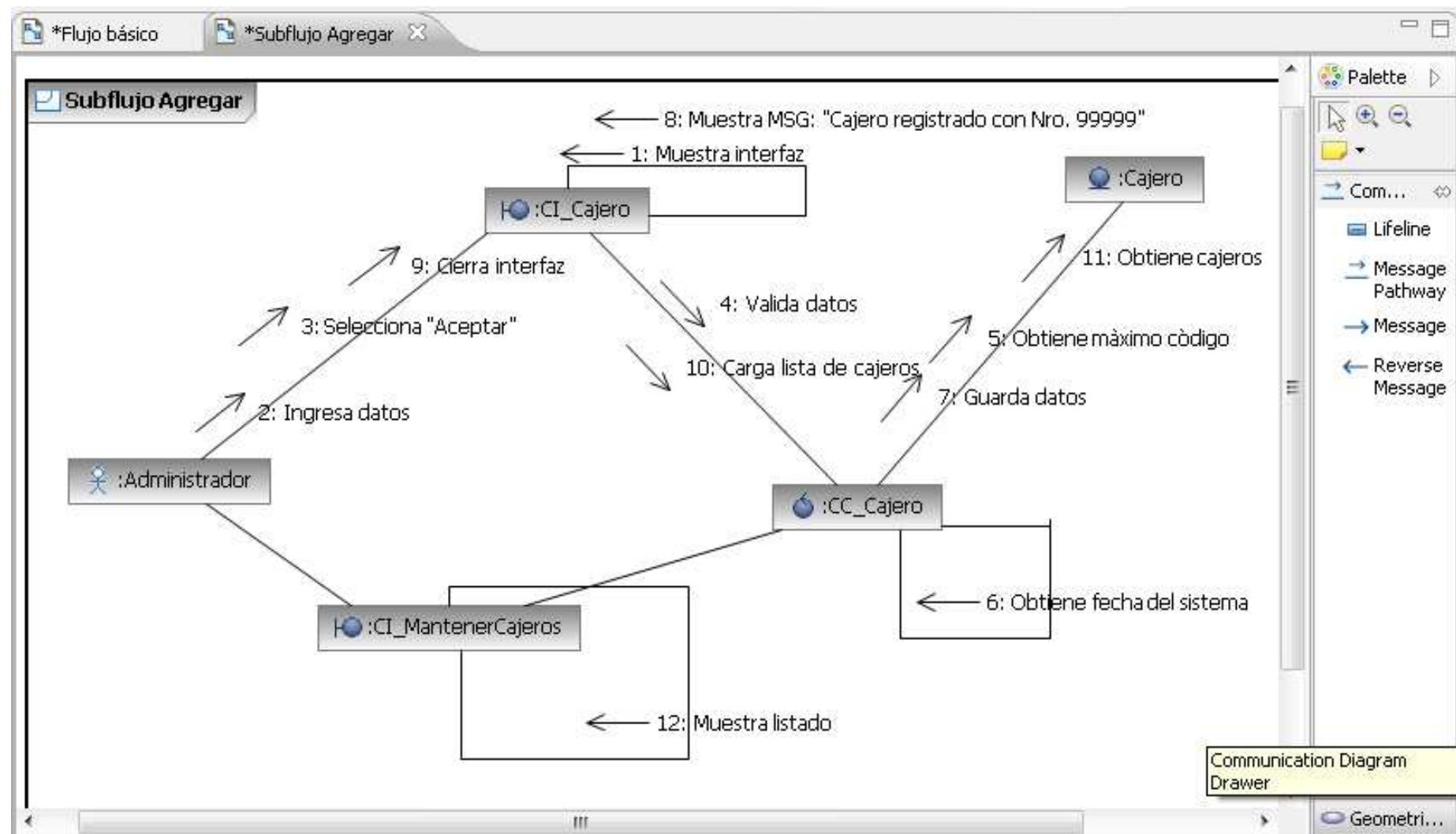
- ACTOR A CLASE INTERFAZ
- CLASE INTERFAZ A CLASE CONTROL
- CLASE CONTROL A CLASE ENTIDAD



3. A continuación, agregue los mensajes (*Message*) sobre los enlaces de acuerdo al flujo descrito en la ECU.



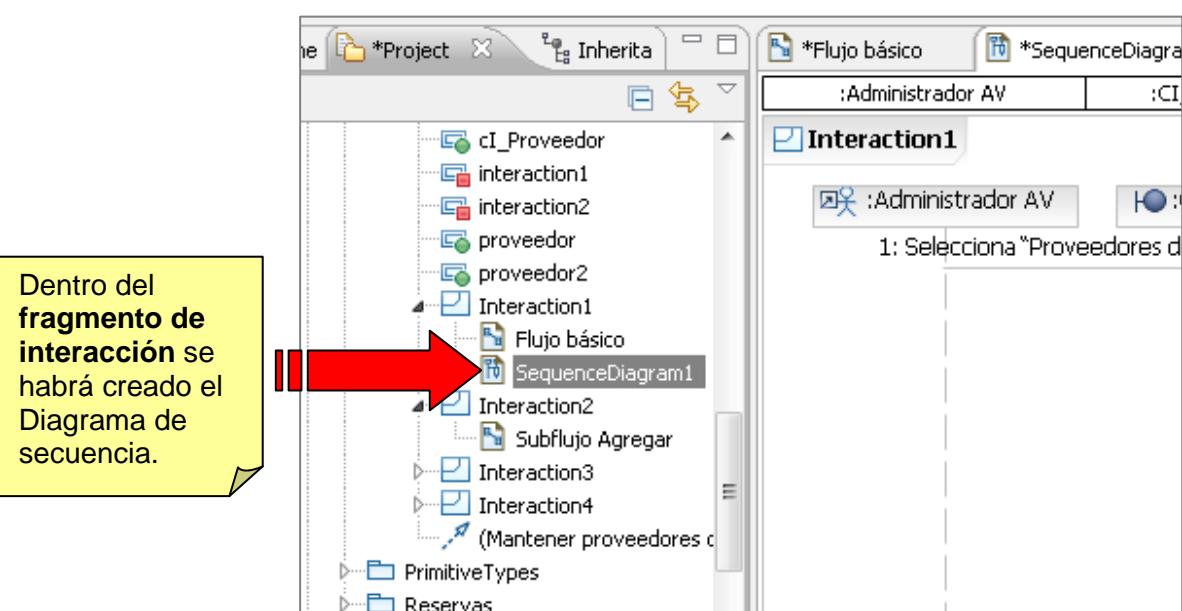
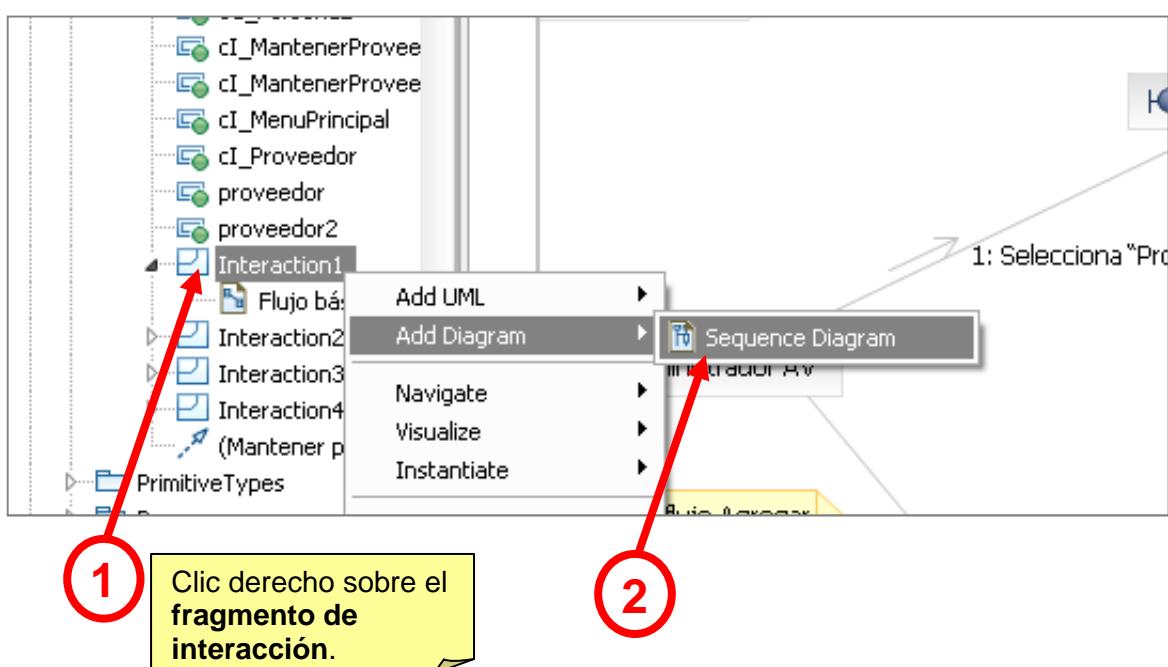
4. Luego, realice el Diagrama de comunicación del subflujo Agregar.



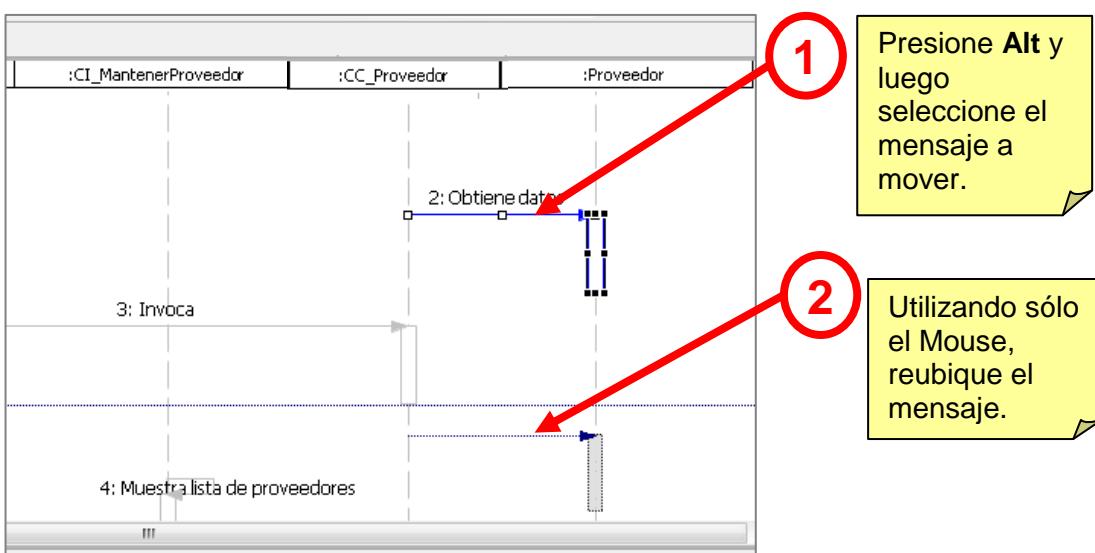
## Reordenar un mensaje en el Diagrama de comunicación

Si desea reordenar los mensajes que se encuentran enumeradas secuencialmente, utilice un Diagrama de secuencias. Para ello, siga los pasos que se indican a continuación.

1. Desde el explorador de proyectos, cree un diagrama de secuencia que contendrá los elementos de un diagrama de comunicación seleccionado.

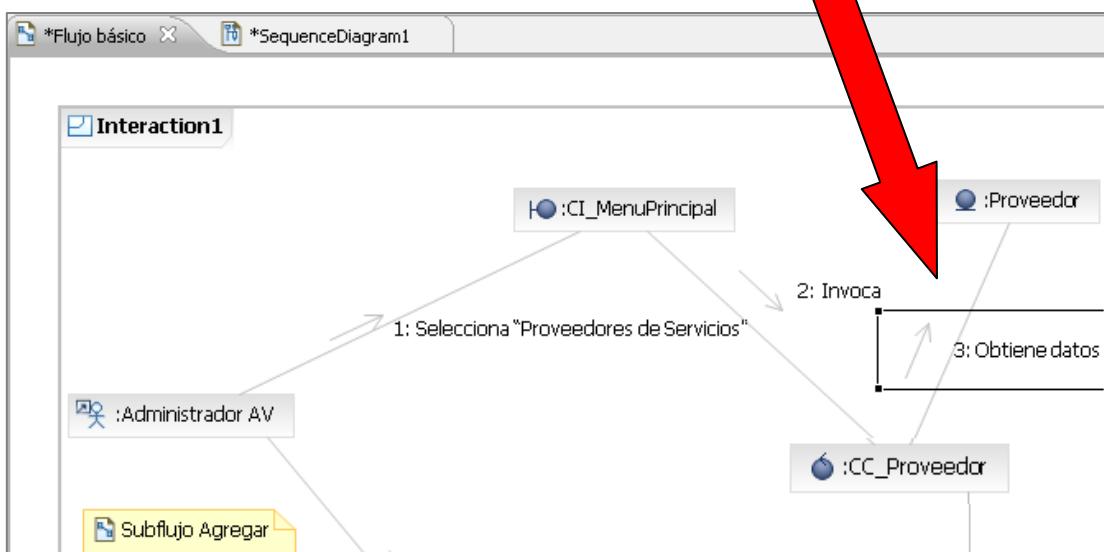
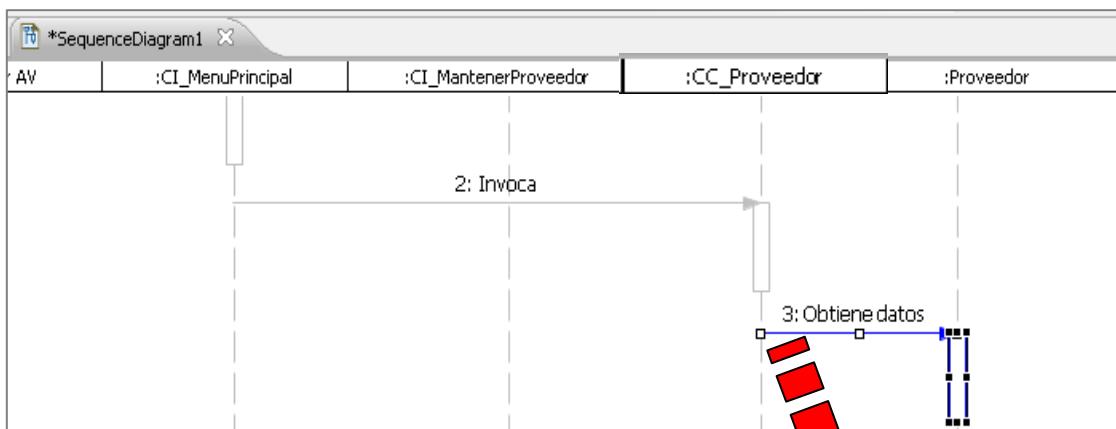


2. En el diagrama de secuencia, podrá mover el orden de los mensajes.



3. Por último, el mensaje aparecerá con otro número de orden, tanto en el diagrama de secuencia como en el de comunicación. Habiendo realizado el proceso de reordenamiento, puede eliminar el diagrama de secuencia.

**NOTA:** Sobre el diagrama de secuencia, también puede agregar nuevos mensajes.



## ACTIVIDAD PROPUESTA

1. Complete la realización del caso de uso **Mantener Cajero**. Para ello, realice los diagramas de comunicación de los subflujos Actualizar y Desactivar.
2. A partir de la **Especificación de un caso de uso** de su proyecto, realice los Diagramas de comunicación del flujo básico y subflujos. Traer para la próxima clase.

## Resumen

- ❑ Para el caso propuesto “Clínica El Ángel” se crearon los siguientes artefactos para un caso de uso:
  - Realización de análisis
  - Diagramas de clases de análisis
  - Diagramas de comunicación del flujo básico y subflujos.
- ❑ Las realizaciones de los casos de uso exige la identificación de clases de análisis, es decir, las clases de entidad, interfaz y control.
  - Una clase de entidad representa el almacenamiento permanente de información.
  - Una clase de interfaz representa interacciones con los actores (usuarios y otros sistemas externos).
  - Una clase control representa el control de interacción entre clases.
- ❑ El diagrama de clases muestra la estructura del caso de uso, mientras el diagrama de comunicación describe su comportamiento.
- ❑ Para crear los diagramas de clases debe considerar la siguiente relación:

ACTOR - CLASE INTERFAZ - CLASE CONTROL - CLASE ENTIDAD

- ❑ Un diagrama de comunicación muestra la colaboración dinámica entre los objetos. Presenta los siguientes elementos: objeto, enlace, mensaje y notas que hacen referencia a otros diagramas de comunicación.

## TEMA 3 REALIZACIONES DE ANÁLISIS DE CASOS DE USO

Análisis de Casos de Uso: Realización de análisis de un caso de uso buscar

A partir de la Especificación de Caso de Uso realice los siguientes artefactos:

1. Realización de análisis del caso de uso
2. Diagrama de clases de análisis
3. Diagrama de comunicación del flujo básico
4. Diagrama de comunicación de los flujos alternativos

### Especificación de caso de uso: Buscar Historia Clínica

#### 1. Descripción:

El caso de uso permite buscar la historia clínica de un paciente por número de historia clínica o nombres y/o apellidos.

#### 2. Actor(es)

Enfermera.  
Recepcionista.

#### 3. Flujo de Eventos

##### 3.1. Flujo Básico

1. El caso de uso comienza cuando es invocado por otro caso de uso base.
2. El sistema muestra la interfaz “BUSCAR HISTORIA CLINICA” con los campos: número de historia clínica, nombres, apellido paterno y apellido materno, y una lista con los datos del resultado de consulta: número de historia clínica, nombres, apellido paterno y apellido materno del paciente. Además, incluye las opciones: **Buscar**, **Aceptar** y **Cancelar**.
3. El actor ingresa el criterio de búsqueda (número de historia clínica o nombres y/o apellidos).
4. El actor selecciona “Buscar”.
5. El sistema muestra la relación de historias clínica de pacientes que coinciden con el criterio de búsqueda.
6. El actor selecciona una Historia Clínica.
7. El actor selecciona “Aceptar”.
8. El sistema carga los datos en la interfaz del caso de uso base que lo invocó y finaliza el caso de uso.

##### 3.2. Flujos Alternativos

###### 1. Clientes no encontrados

En el paso 5, si el sistema no muestra ninguna Historia Clínica por el criterio de búsqueda muestra el MSG: “No se encuentra la historia clínica para el criterio ingresado”. El caso de uso continúa en el paso 3 o si el actor selecciona “Cancelar”, finaliza el caso de uso.

###### 4. Pre Condiciones

1. El actor se ha identificado en el sistema.
2. Lista de historias clínicas disponibles.

## 5. Post Condiciones

Ninguna.

## 6. Puntos de Extensión

Ninguno.

## 7. Requisitos Especiales

Ninguno.

## 8. Prototipos

### Buscar Historia Clínica

Ingrese aquí los datos de la historia clínica de un paciente que desea buscar.

#### Criterios de búsqueda

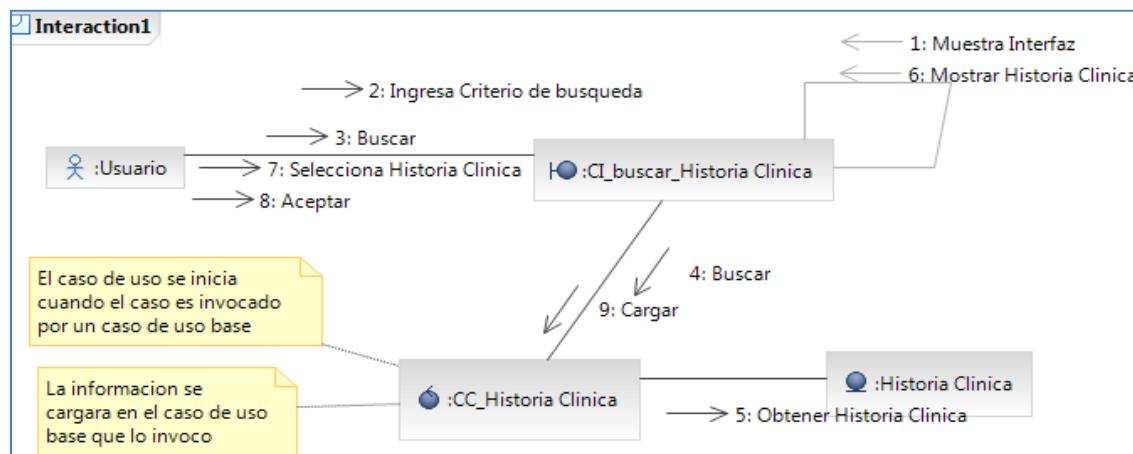
**Nota:** Para buscar historias clínicas ingresando solo un fragmento de su número, nombres y/o apellidos use el carácter "%".

Nº HC :	<input type="text"/>	Nombre :	<input type="text"/>
Apellido paterno :	<input type="text"/>	Apellido materno :	<input type="text"/>
<input type="button" value="Buscar"/>			

### Diagrama de Clases de análisis



### . Diagrama de Comunicación – Buscar Historia Clínica



## CASO PRÁCTICO Nº 2

A partir de la Especificación de Caso de Uso realice los siguientes artefactos:

1. Realización de análisis del caso de uso
2. Diagrama de clases de análisis
3. Diagrama de comunicación del flujo básico
4. Diagrama de comunicación de los flujos alternativos

### Especificación de caso de uso: Generar Cita

#### 1. Descripción:

El caso de uso permite a la recepcionista de la clínica registrar una cita médica para consultas externas por especialidad.

#### 2. Actor(es)

Recepcionista

#### 3. Flujo de Eventos

##### 3.1. Flujo Básico

1. El caso de uso comienza cuando la recepcionista selecciona la opción “Generar Citas” de la interfaz del menú principal.
2. El sistema muestra la interfaz “GENERAR CITAS” con la fecha y hora de registro cargado y los siguientes campos:
  - Datos de la HC: número, nombre y apellidos del paciente.
  - Datos del médico: nombres, apellidos y especialidad.
  - Datos de la cita son: fecha y hora de la cita y consultorio.
  - Además, presenta las opciones: Buscar HC, Buscar Horarios de Médico y Grabar Cita.
3. La recepcionista selecciona “Buscar HC”.
4. El sistema **incluye el caso de uso Buscar Historia Clínica**.
5. El sistema muestra los datos de la historia clínica del paciente.
6. La recepcionista selecciona “Buscar Médico”.
7. El sistema **incluye el caso de uso Buscar Horarios de Médico**.
8. El sistema muestra los datos del médico y de la cita.
9. La recepcionista selecciona “Grabar Cita”.
10. El sistema valida los datos.
11. El sistema genera el número de cita y registra la cita con estado pendiente.
12. El sistema muestra el mensaje “Cita generada”. El caso de uso termina.

##### 3.2. Flujos Alternativos

###### 1. No existe HC

Si en el paso 6, el sistema detecta que no existe la HC del paciente, muestra el MSG “No existe HC” y ofrece la posibilidad de registrar la HC del paciente.

###### 2. No hay médicos disponibles

Si en el paso 8 el sistema detecta que no hay médicos disponibles, muestra el MSG “No hay médicos disponibles” y el caso de uso finaliza.

**3. Campos vacíos**

Si en el paso 10 el sistema detecta que alguno de los campos está vacío, muestra el MSG “Alguno de los campos está vacío” y el caso de uso continúa.

**4. Precondiciones**

1. La recepcionista está identificada en el sistema.
2. Lista disponible de historias clínica.
3. Lista disponible de médicos.

**5. Poscondiciones**

1. En el sistema quedará registrada la cita en estado pendiente.

**6. Puntos de Extensión**

En el paso 6, el sistema extiende al caso de uso Mantener HC – subflujo “Registrar HC”.

**7. Requisitos Especiales**

Ninguno.

**8. Prototipos**

**Generar Citas**

---

Fecha: 08/03/2010  
Hora: 15:32

**Datos de la HC del paciente**

Nº HC:

Paciente:

**Datos del médico**

Nombre y apellidos:

Especialidad:

**Datos de la cita**

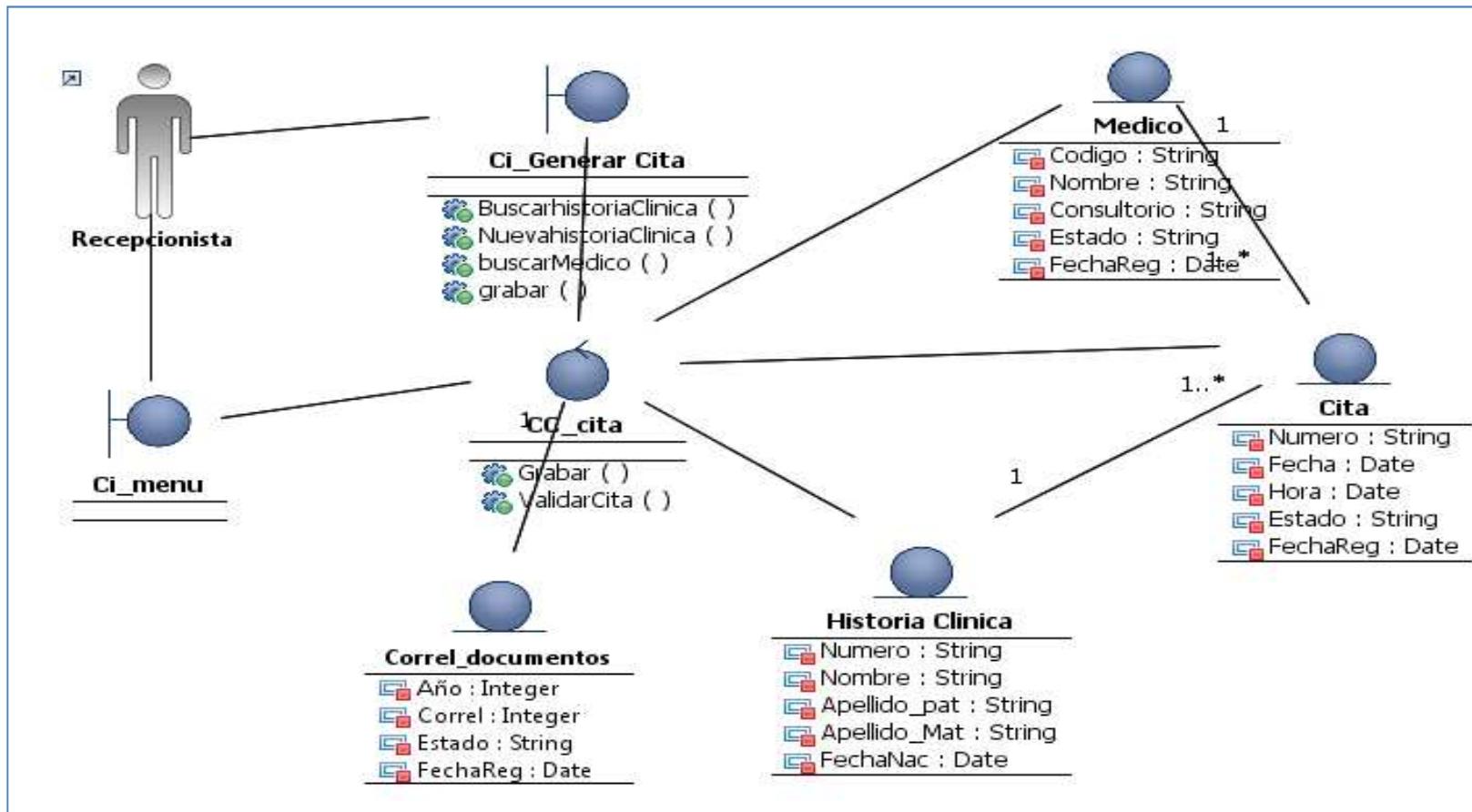
Fecha:

Hora:

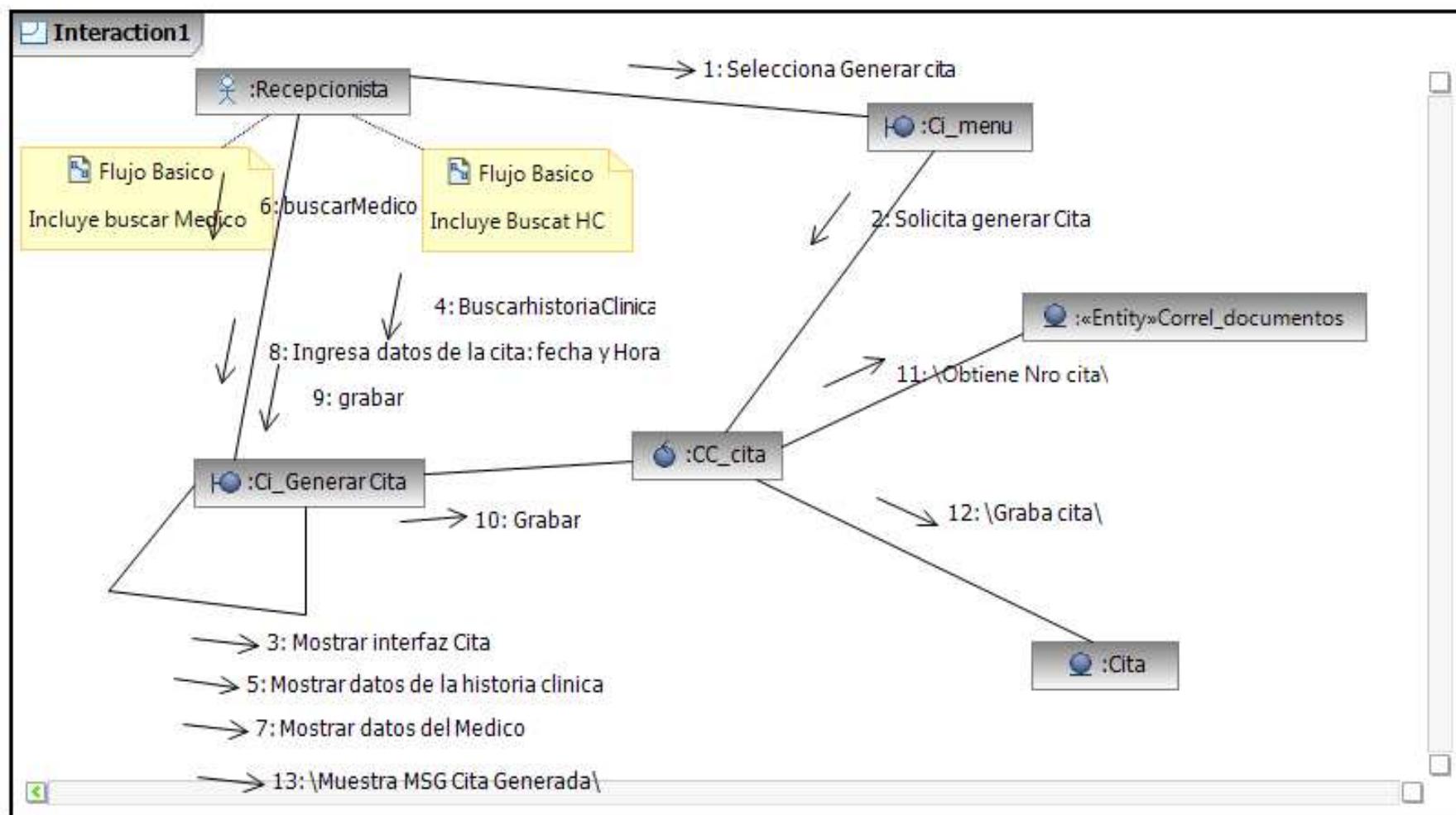
Consultorio:

---

## Diagrama de Clases de análisis



## Diagrama de Comunicación – Generar Cita



## CASO PRÁCTICO N° 1

A partir de la Especificación de Caso de Uso, realice los siguientes artefactos:

1. Realización de análisis del caso de uso
2. Diagrama de clases de análisis
3. Diagrama de comunicación del flujo básico
4. Diagrama de comunicación de los flujos alternativos

### Especificación de Caso de uso: Generar reporte tributario

#### 1. Breve descripción

El caso de uso permite al contribuyente solicitar un estado de cuenta de la información que posee en su registro de datos tales como los siguientes: datos personales, representantes legales, locales comerciales, deuda tributaria, fraccionamientos tributarios, declaraciones de impuestos y declaraciones únicas aduaneras.

#### 2. Flujo de Eventos

##### 2.1. Flujo Básico

1. El caso de uso comienza cuando el contribuyente solicita "Generar reporte tributario" en el menú principal.
2. El sistema muestra la interfaz "Reporte tributario y Aduanero" con los siguientes datos:  
Un mensaje Informando que se generará un reporte con resumen de la información que posee el contribuyente colocando como fecha de vigencia el día actual, Un *check* de selección para solicitar el envío a un correo electrónico y la casilla para colocar el correo; además, las opciones **Generar y limpiar**.  
Además, incluye una cuadricula que contiene la lista de todas las solicitudes de reportes generados.
3. El contribuyente selecciona el *check* de envío de correo e ingresa el correo electrónico
4. El contribuyente selecciona **Generar**
5. El sistema obtiene el número del Correlativo de la Solicitud de Reporte tributario e incrementa en uno para generar el nuevo correlativo de Solicitud de Reporte tributario.
6. El sistema graba la Solicitud de Reporte tributario en estado pendiente
7. El sistema obtiene los datos del contribuyente (razón social, teléfonos, dirección, tipo de contribuyente, actividad económica), los datos de los representantes legales (tipo y número de documento de identidad, fecha de inicio de representación, domicilio) , los datos de los locales (tipo y número de local, dirección, fecha de alta), los datos de las deudas (tipo de duda, monto y fecha de vencimiento) , los tipo de fraccionamiento( número y tipo, monto de la deuda y cantidad de cuotas) y las declaraciones juradas(tipo de declaración, fecha , monto pagado) genera un archivo en pdf con la información y se lo envía por correo al contribuyente y actualiza la solicitud como atendida
8. El sistema muestra la Interfaz "constancia de Solicitud" con el número de Solicitud y el MSG "Se ha generado su reporte tributario" con el Nro. 99999".

##### 2.3. Flujos Alternativos

###### <limpiar>

El contribuyente selecciona limpiar, el sistema borra el correo electrónico ingresado.

**3. Requerimientos Especiales**

No presenta.

**4. Pre Condiciones**

- El contribuyente está logeado en el sistema.

**5. Post Condiciones**

- Se graba la solicitud y se genera el reporte tributario.

**6. Puntos de extensión**

No presenta

**7.-Prototipo****Interfaz “Reporte Tributario y aduanero”**

### Reporte Tributario y Aduanero

Con esta opción usted podrá generar un nuevo reporte Tributario y Aduanero, que contendrá un resumen de información Tributaria y Aduanera perteneciente al rango de fechas del 14/07/2010 al 14/07/2011, la generación durará como máximo una hora de acuerdo a la cantidad de información del contribuyente, cuando esté listo el estado de su reporte será "Generado" en la opción de consulta y podrá abrirlo haciendo click sobre el nombre del reporte. Se le notificará a su buzón electrónico de la SUNAT (menú SOL) y a un correo electrónico particular que usted puede ingresar a continuación.

Envío a correo electrónico

[Generar](#)
[Limpiar](#)

\*Sólo se podrá generar el reporte una vez por día.

1 a 1 de 1

Nro.	Fecha Generación	Estado	Reporte Tributario/Aduanero	Enviar
1	17/05/2011 18:12:32	Generado	<a href="#">REPTRIAADU_170511</a>	

**Interfaz “constancia de solicitud”**

### Reporte Tributario y Aduanero

Se ha iniciado la generación de su Reporte Tributario y Aduanero.

La generación demorará como máximo una hora, cuando finalice el procesamiento se le enviará una notificación a su buzón electrónico y al correo electrónico ingresado: kikesnp@hotmail.com

Número de solicitud: 0023000001043





## MODELO DE DATOS

### LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la segunda unidad, el alumno crea el modelo de datos, el cual incluye el modelo conceptual, modelo lógico y modelo físico. Los artefactos serán creados utilizando la herramienta CASE *IBM InfoSphere Data Architect (IDA)*.

### TEMARIO

1. Modelo Conceptual
2. Modelo Lógico
3. Modelo Físico

### ACTIVIDADES PROPUESTAS

1. Los alumnos crean el modelo conceptual de un caso propuesto.
2. Los alumnos generan el modelo lógico de un caso propuesto.
3. Los alumnos generan el modelo físico de un caso propuesto.

## 1. MODELO CONCEPTUAL

A partir de todas las entidades identificadas en las realizaciones de casos de uso, se creará un modelo conceptual. El enfoque del modelo conceptual debe responder a la pregunta que cuestiona las razones para tener datos persistentes. El conocimiento de esta razón permite a los analistas y diseñadores de bases de datos modelar la base de datos correcta.

Las tareas a realizar para crear el modelo conceptual son las siguientes:

- Identificar atributos.
- Identificar asociaciones.
- Identificar agregaciones.
- Agregar multiplicidades en las relaciones de asociación y agregación.

### 3.1. Identificar atributos

Un atributo especifica una propiedad de una clase de análisis y, normalmente, es necesaria para identificar las responsabilidades de su clase. Consideraremos las siguientes reglas generales cuando identificamos atributos para una clase entidad:

- Los atributos corresponden a sustantivos y sus valores pueden ser sustantivos o adjetivos.
- Los atributos pueden representarse solo mostrando su nombre, su tipo e incluso su valor por defecto.
- Los atributos multivaluados deben ser clasificados como clases.
- Si una clase de análisis se hace demasiado difícil de entender a razón de sus atributos, algunos de estos podrían separarse en clases independientes.
- Asociar atributos a las clases que ellos describen más directamente. Los atributos deben ser inherentes a la clase. Es, por ello, que no se debe considerar agregar atributos de otras clases, pues no estamos identificando llaves foráneas.

### 3.2. Identificar asociaciones

La asociación es una relación entre clases que indica una conexión significativa e interesante. Está representada como una línea entre clases con nombre.

La asociación es inherentemente bidireccional. Es convencional leer la asociación de izquierda a derecha o de arriba hacia abajo.

Las asociaciones pueden ser binarias, ternarias o de mayor grado.

### 3.3. Identificar agregaciones

La Agregación indica una relación de “un todo conformado por partes”. Existen dos tipos de agregaciones: débil o compartida y fuerte o compuesta.

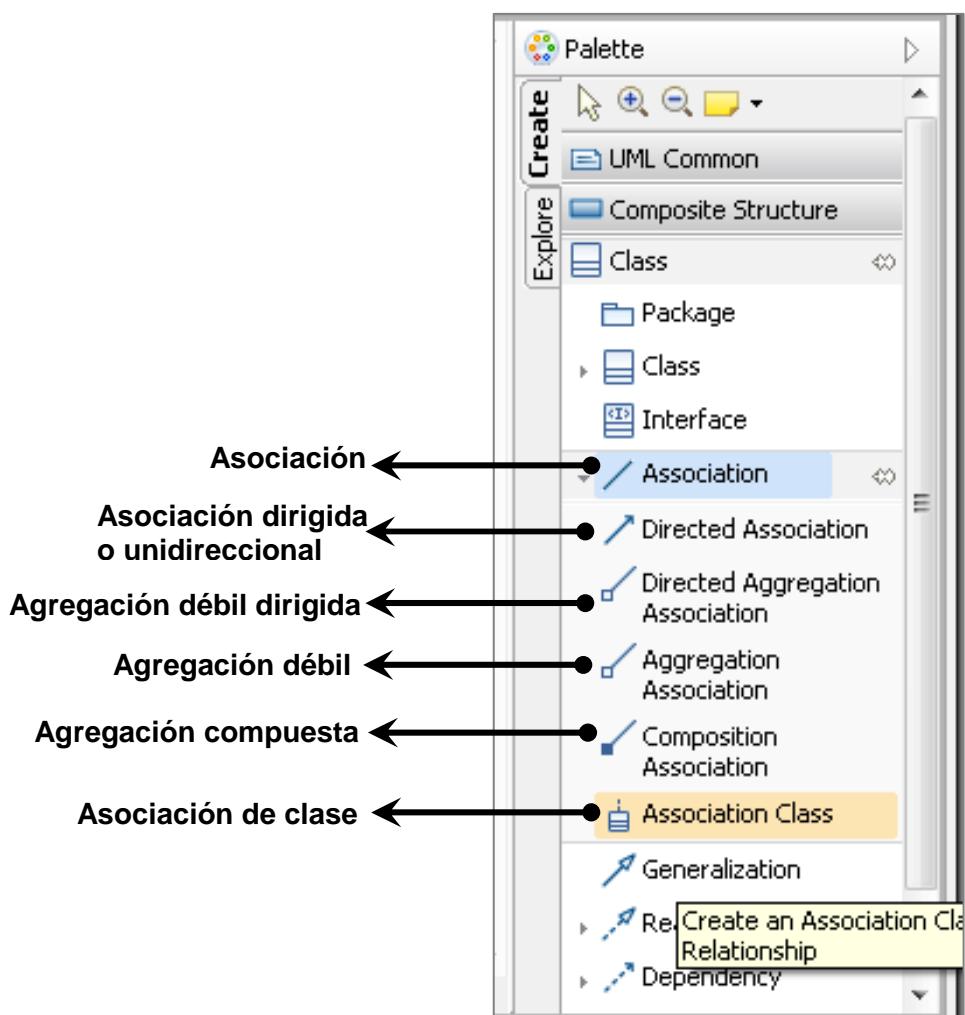
- **Agregación débil.** Es un tipo de relación utilizada para modelar la relación todo-parte entre objetos. La parte puede estar simultáneamente en varias instancias del todo.
- **Agregación compuesta.** Es un tipo de relación utilizada para modelar la relación todo-parte entre objetos. Significa que la parte es miembro de solamente un objeto todo, es decir, la existencia de la parte depende del todo. El objeto todo es el único dueño del objeto parte. La composición se representa con un rombo relleno.

### 3.4. Multiplicidad

Básicamente, la multiplicidad define cuantas instancias de la clase A pueden estar asociadas con una instancia de la clase B.

Cuando en una **asociación de clases nace la relación muchos a muchos entre las dos** y existe información asociada con la propia asociación se crea una clase asociativa.

A continuación, se muestra la paleta del RSA que contiene los tipos de relaciones entre clases:



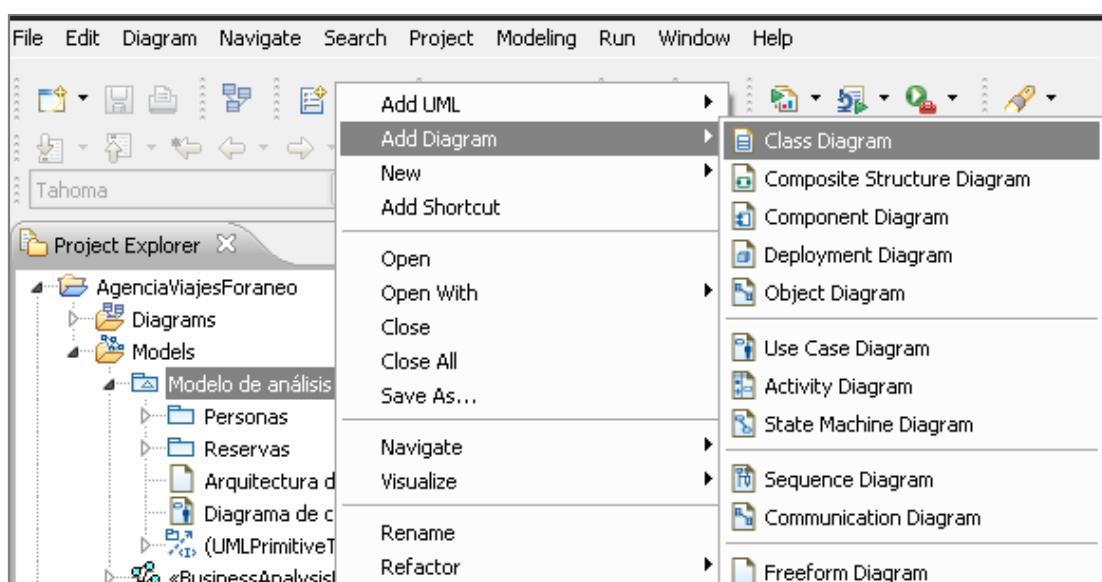
## CASO DE ESTUDIO

A continuación, se muestran los pasos para confeccionar un Modelo conceptual con otro caso de estudio; pero usted, para la creación de dicho modelo, considere los siguientes casos de uso:

- Generar Cita
- Buscar Historia Clínica
- Buscar Horario de Médicos
- Mantener Historia Clínica
- Mantener Cajero

### Creación del Modelo conceptual

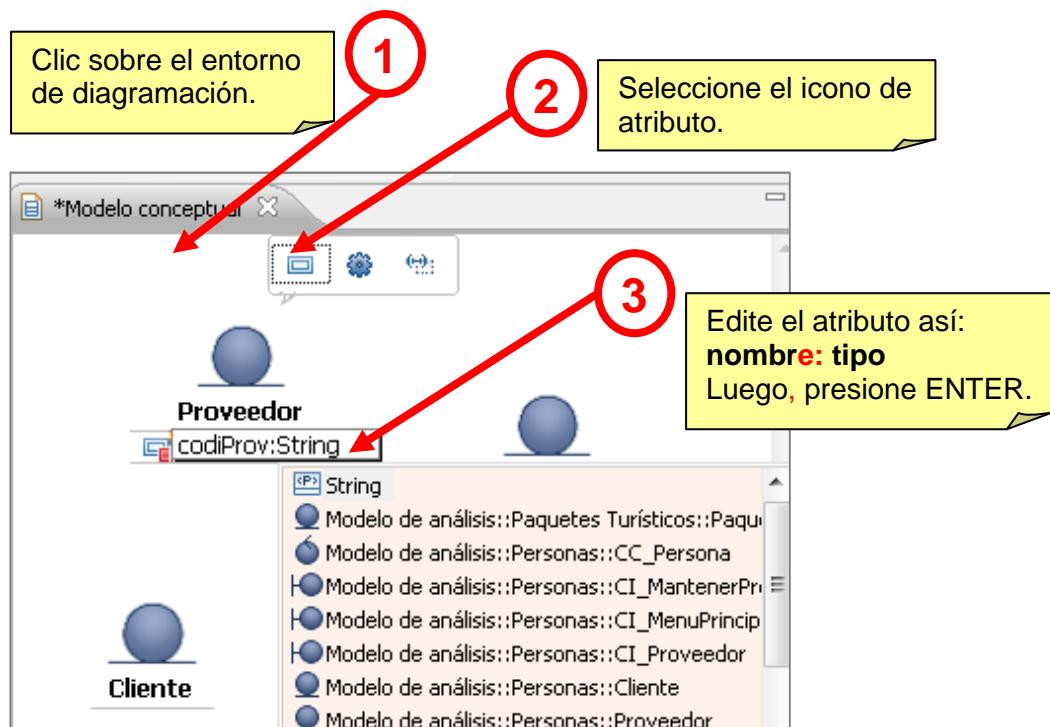
1. Sobre el Modelo de análisis agregue un diagrama de clases con el nombre **Modelo conceptual**.



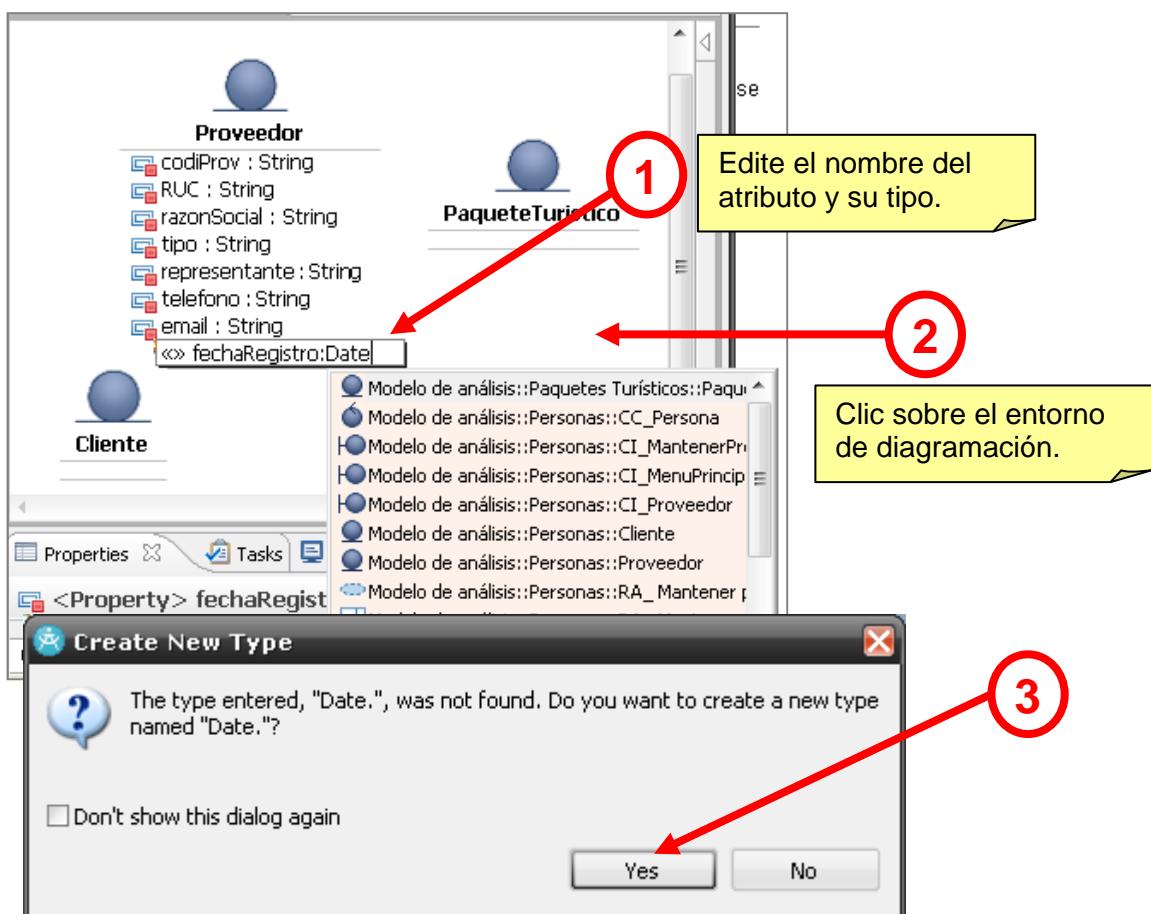
2. Arrastre al Modelo conceptual todas las clases entidad que se han obtenido en las diferentes realizaciones de casos de uso del sistema.



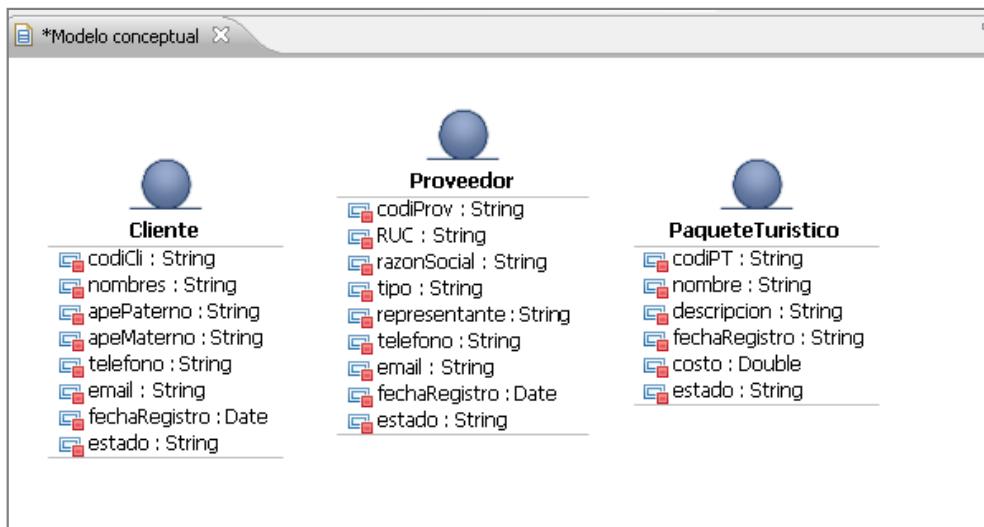
3. Luego, agregue los atributos en cada entidad.



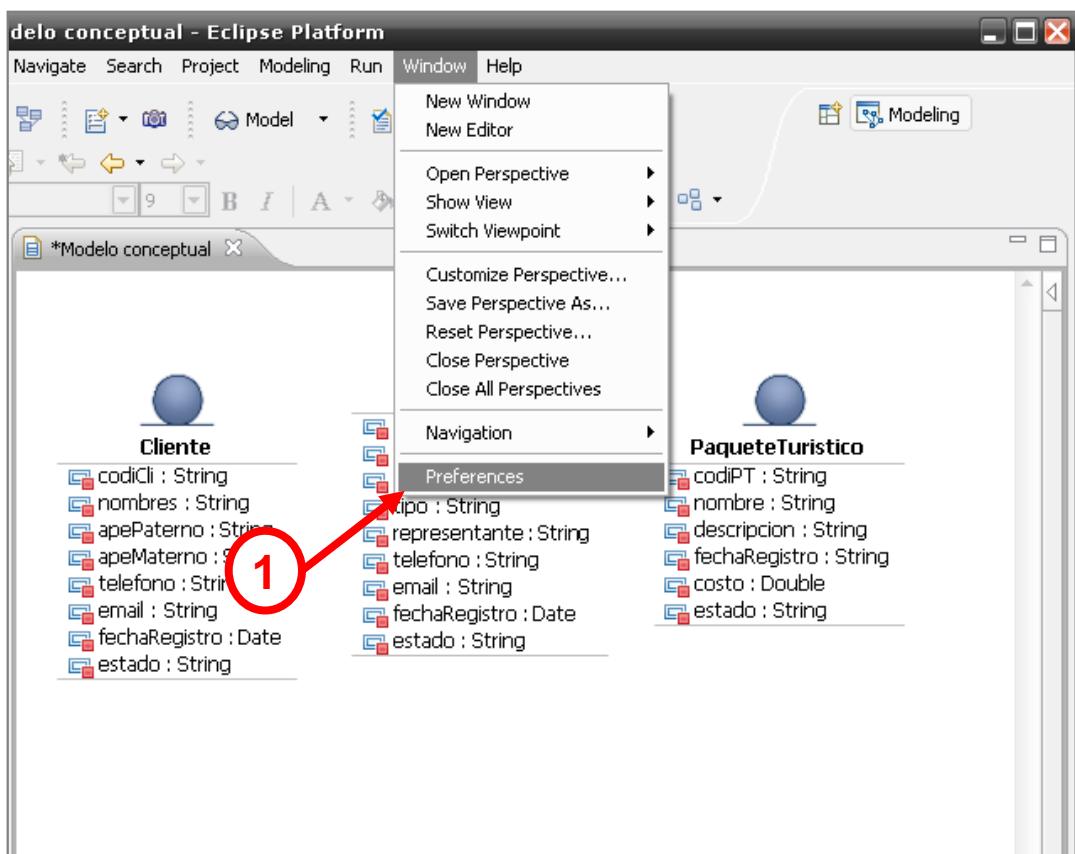
4. Si para un atributo se agrega un tipo que no existe, se sigue los pasos que se muestra a continuación.



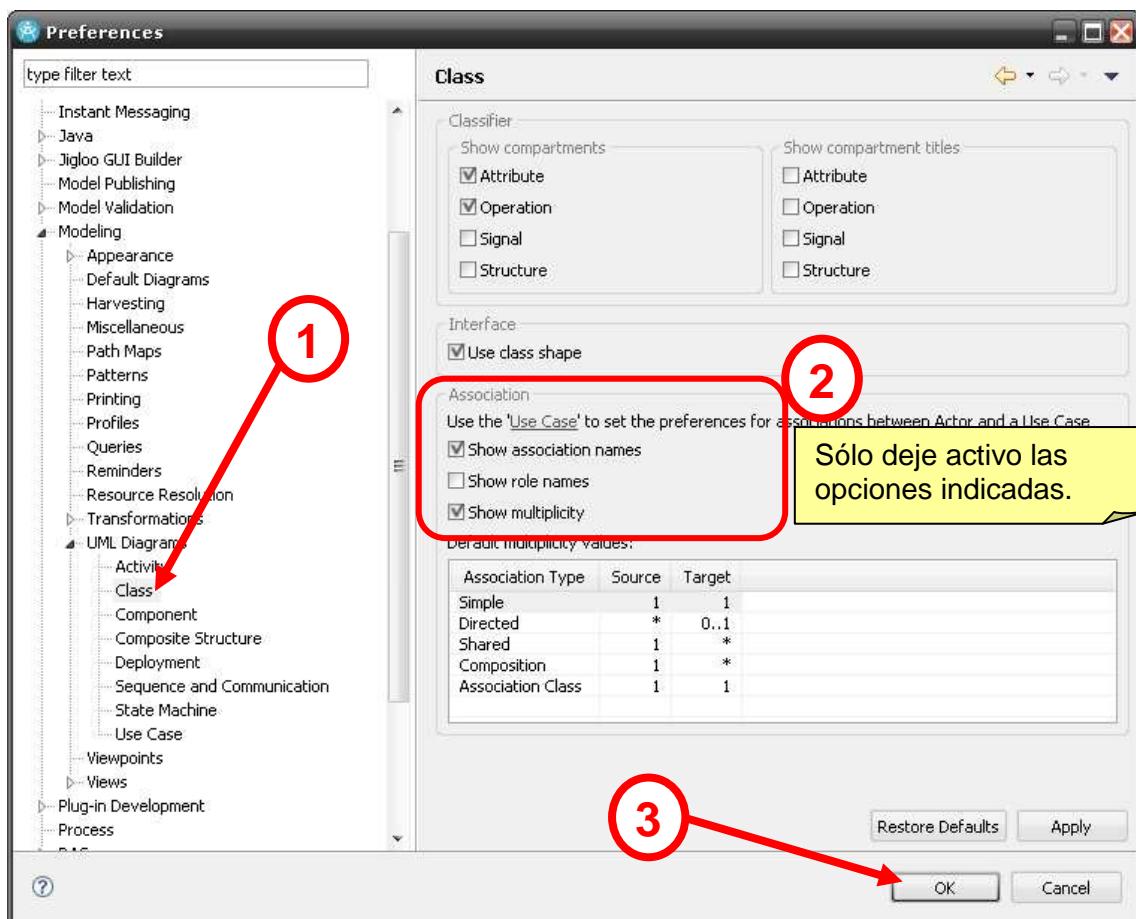
5. Complete los atributos de las demás clases.



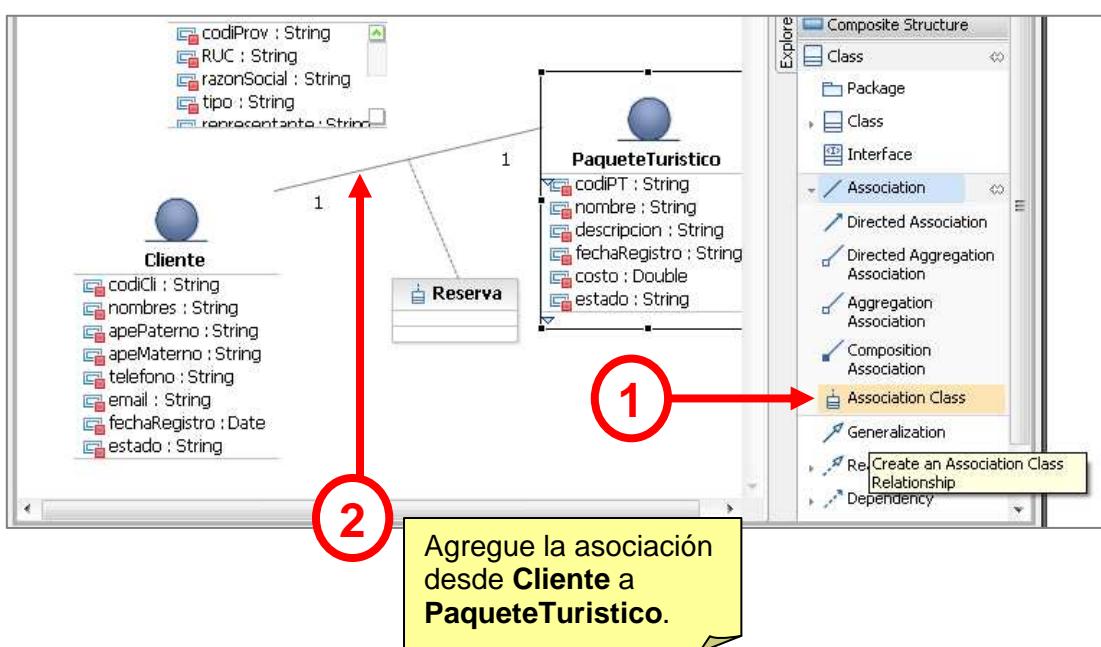
6. Configure las preferencias para el Modelo conceptual.



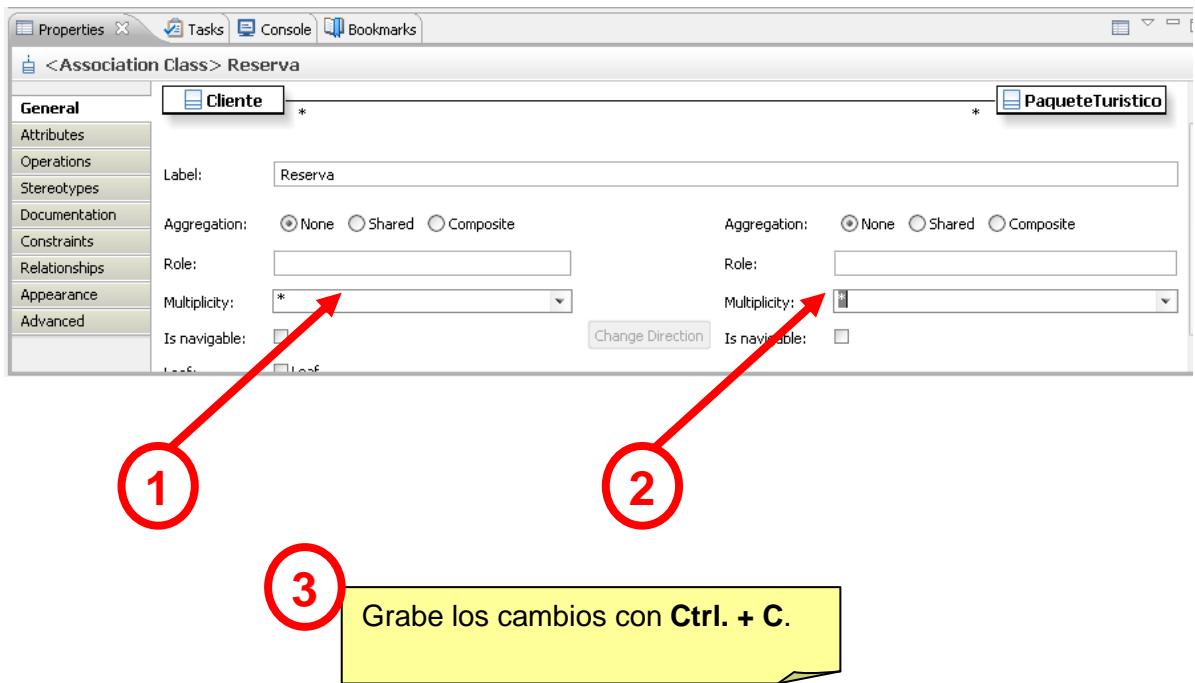
7. Seleccione categoría UML Diagrams/Class para activar o desactivar las opciones de asociación.



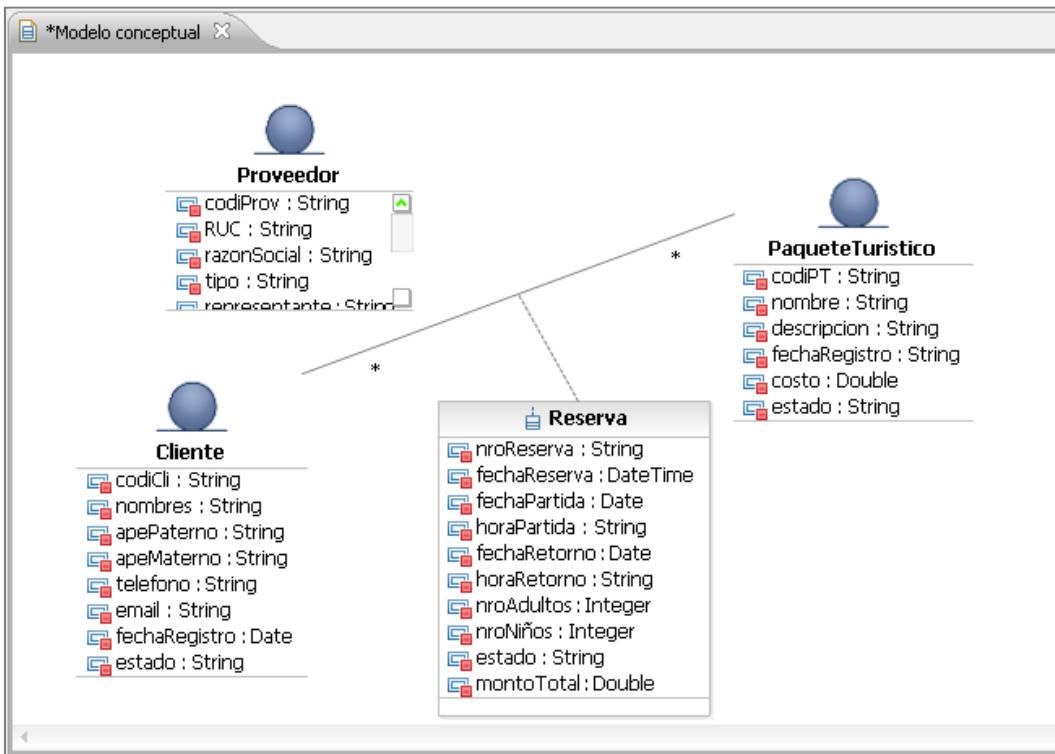
8. Agregue las asociaciones entre las clases. En el caso que se encuentre con una asociación de muchos a muchos, utilice una relación del tipo asociación de clase.



9. En la ventana de propiedades, sección general, cambie las multiplicidad de la asociación de clase a muchos-muchos.



10. Por último, se agrega los atributos a la nueva clase.



## Resumen

- ॥ Las tareas a realizar para crear el modelo conceptual son las siguientes:
  - Identificar atributos.
  - Identificar asociaciones.
  - Identificar agregaciones.
  - Identificar multiplicidades.
- ॥ El resultado de una asociación de muchos a muchos es una clase asociativa.
- ॥ La relación de agregación indica una relación de “un todo conformado por partes”. Existen dos tipos de agregaciones: débil o compartida y fuerte o compuesta.

## 2. MODELO LÓGICO

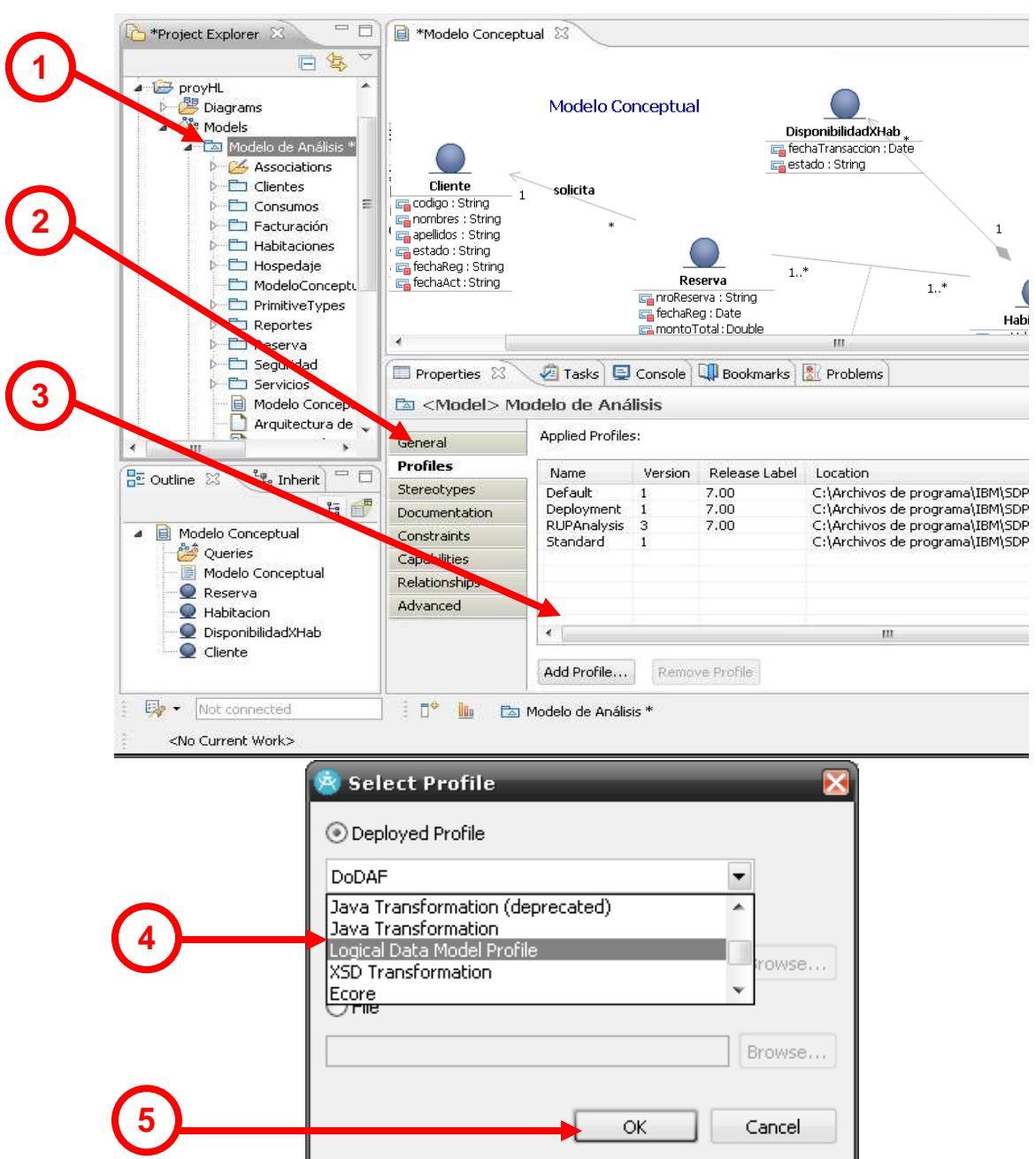
A continuación, se muestran los pasos para confeccionar un Modelo Lógico de Datos con otro caso de estudio; pero usted, para la creación de dicho modelo, considere el Modelo Conceptual del caso de estudio Clínica El Ángel.

El esquema de trabajo para crear el modelo lógico de datos es el siguiente:

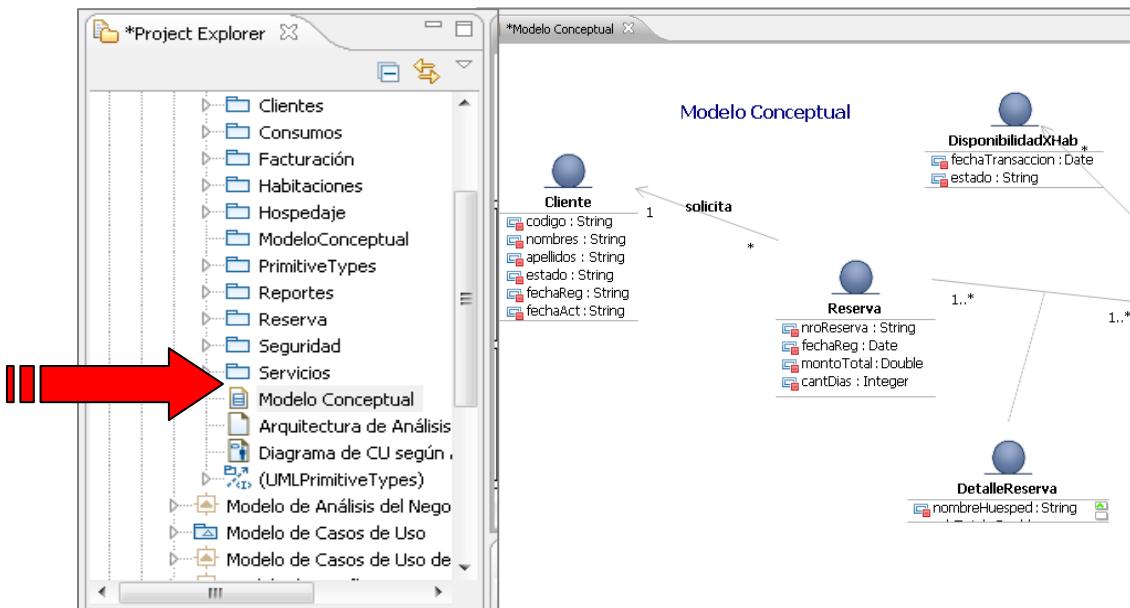
1. Configurar el Modelo que contiene el modelo conceptual
2. Transformar un modelo UML en un modelo lógico de datos

### Configuración del modelo conceptual

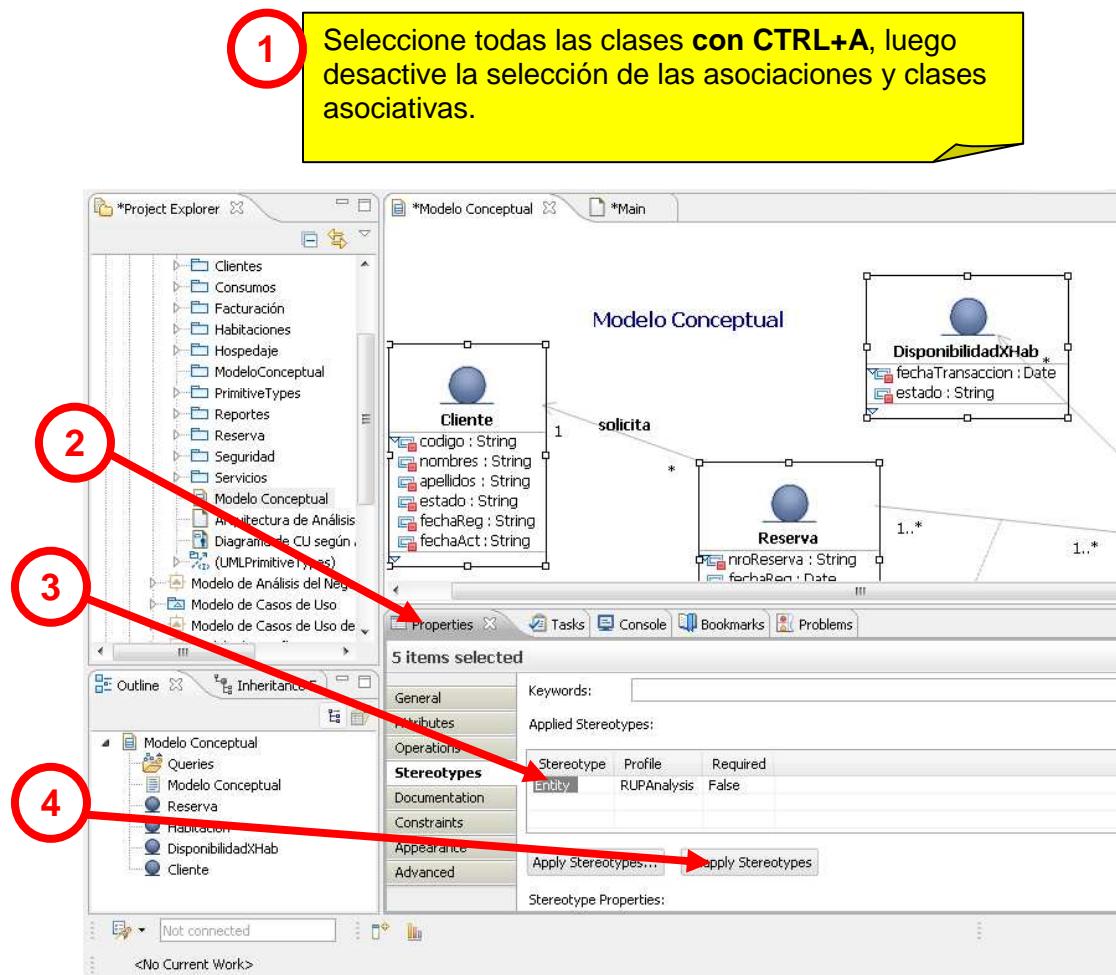
**PASO 1:** Desde la vista de explorador de proyectos, seleccione el Modelo de Análisis para agregar el perfil **Logical Data Model Profile**



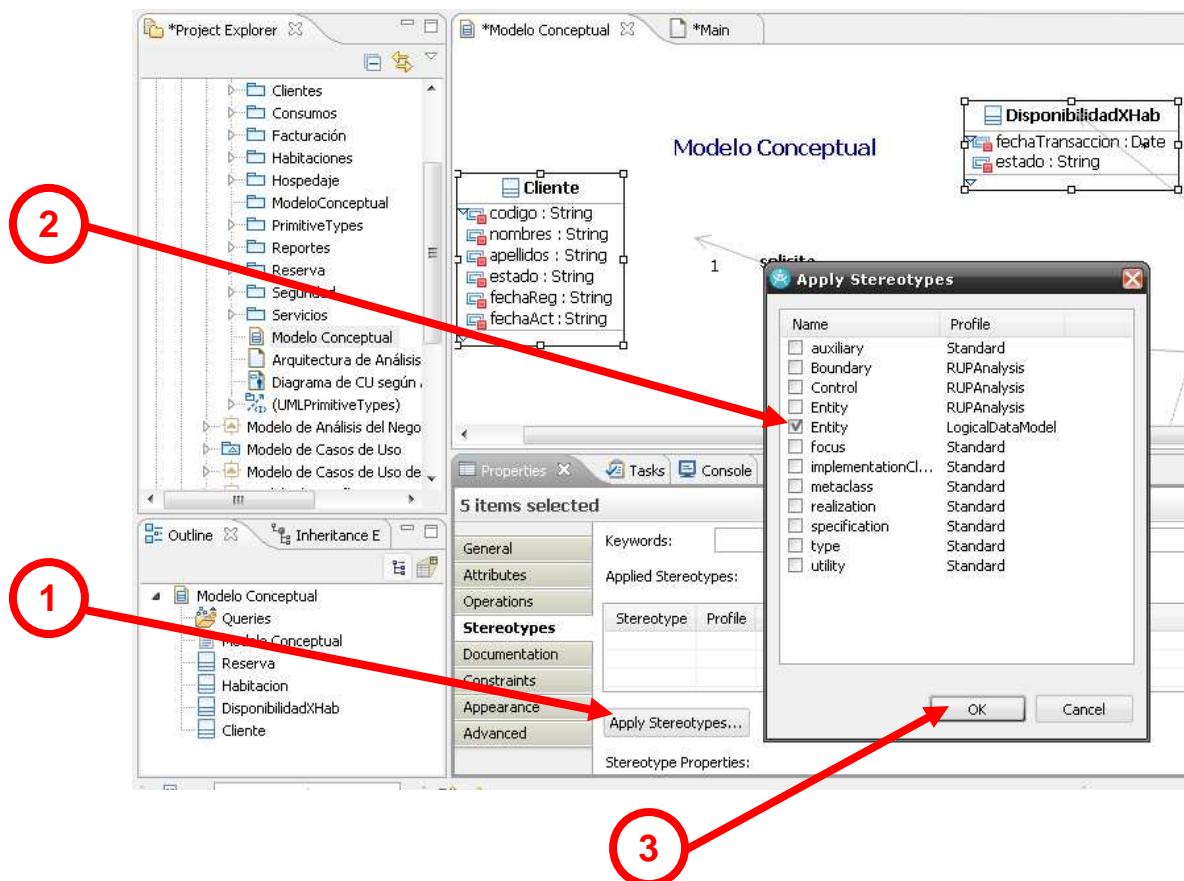
## PASO 2: Ir al modelo Conceptual del CASO



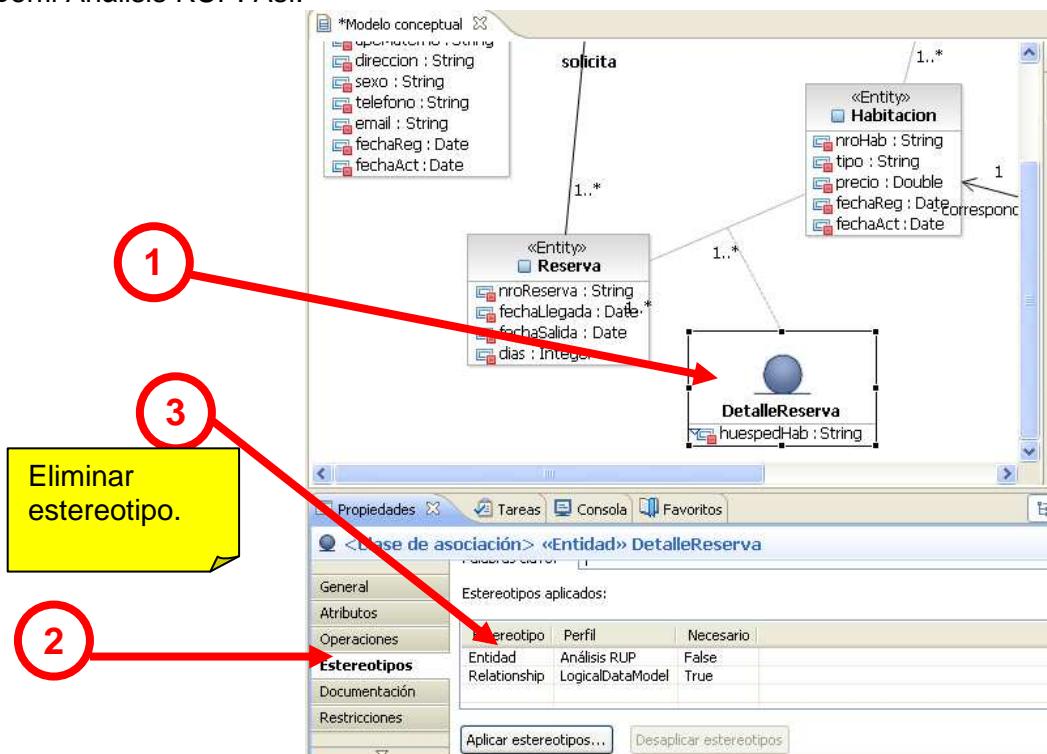
## PASO 3: Elimine el estereotipo actual de las clases de la siguiente manera:



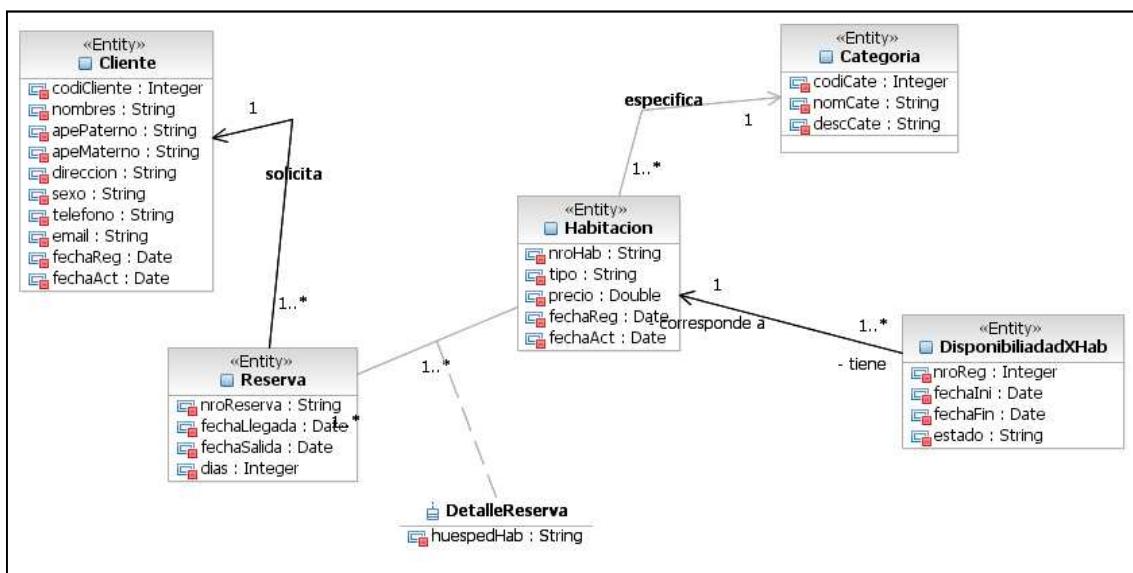
**PASO 4:** Agregue el estereotipo de las clases a **Entity** del perfil Modelo Lógico de Datos:



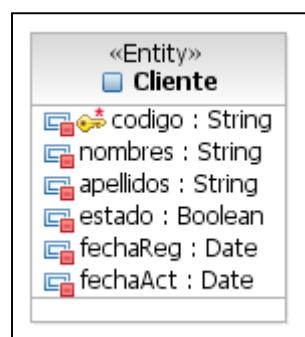
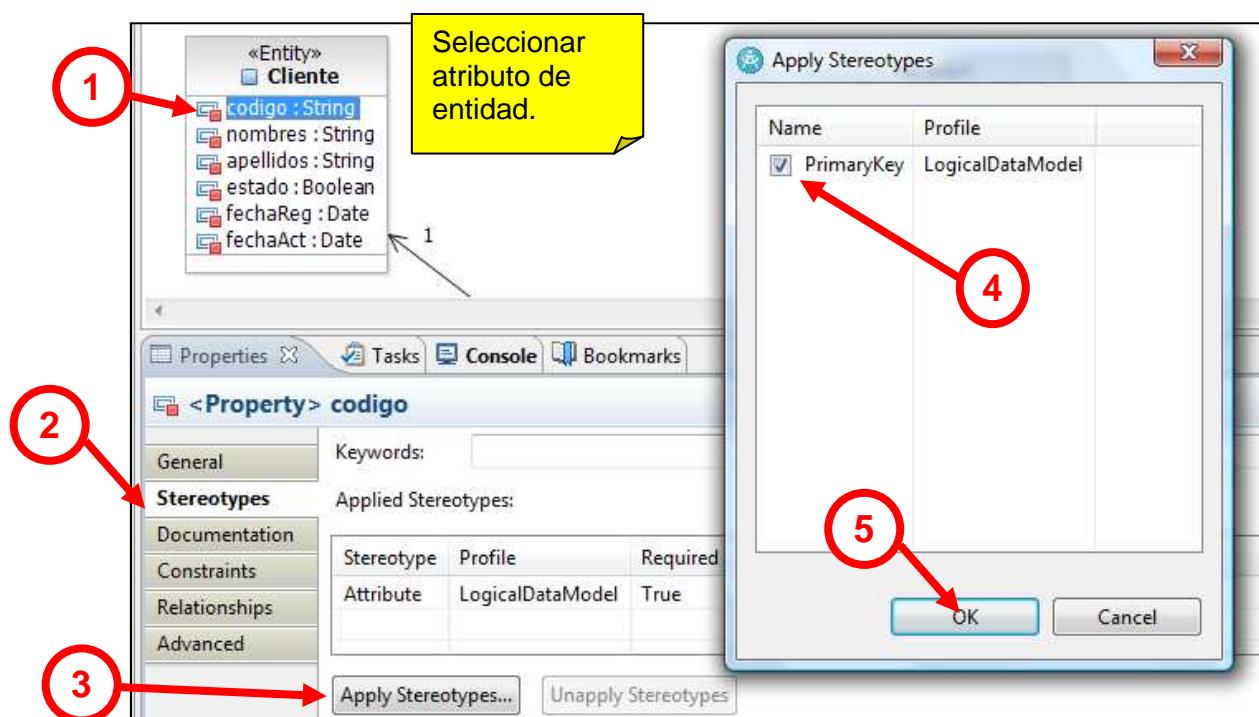
**PASO 5:** En caso de clases asociativas, debe eliminar el estereotipo de Entidad del perfil Análisis RUP. Así:



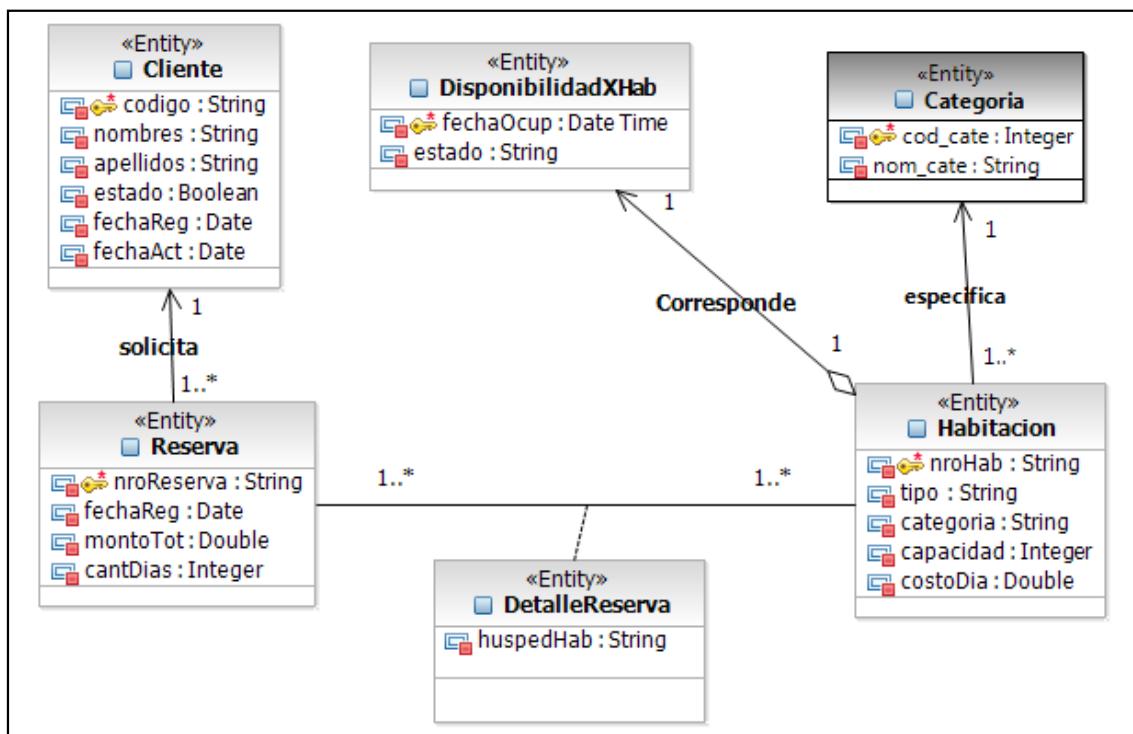
Las entidades del Diagrama habrán cambiado de estereotipo:



**PASO 6** A cada entidad, se le debe de agregar su Identificador de PrimaryKey, seleccionando el atributo que tendrá el identificador único en cada entidad.



Las entidades del Diagrama se mostrarán con el estereotipo de PK:

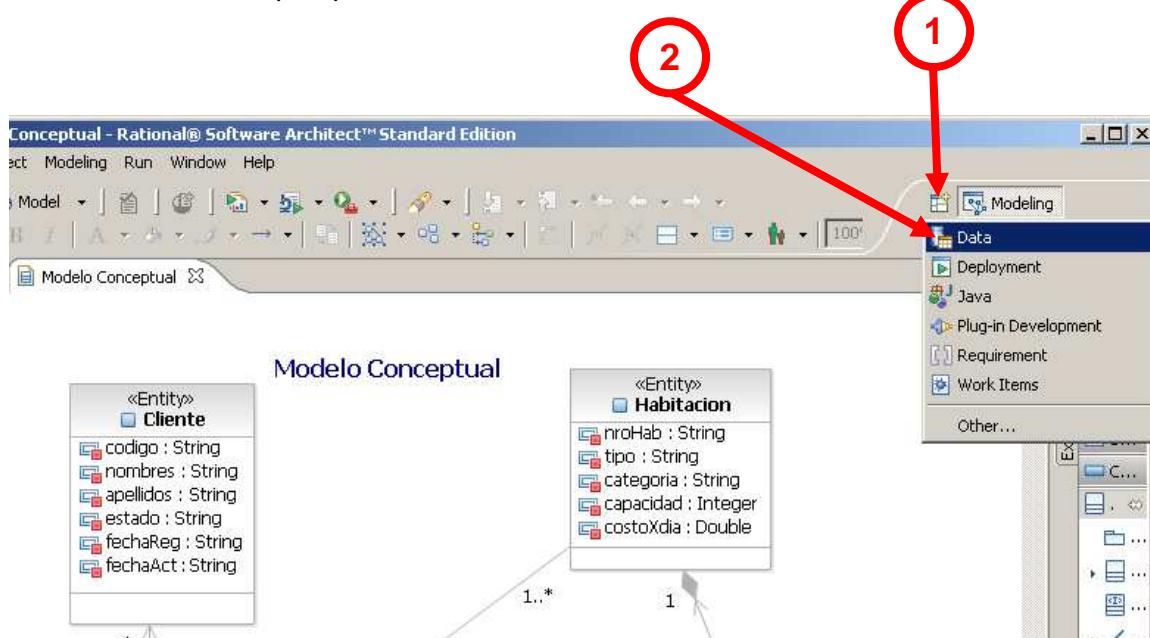


## Transformación de un modelo UML en un modelo lógico de datos

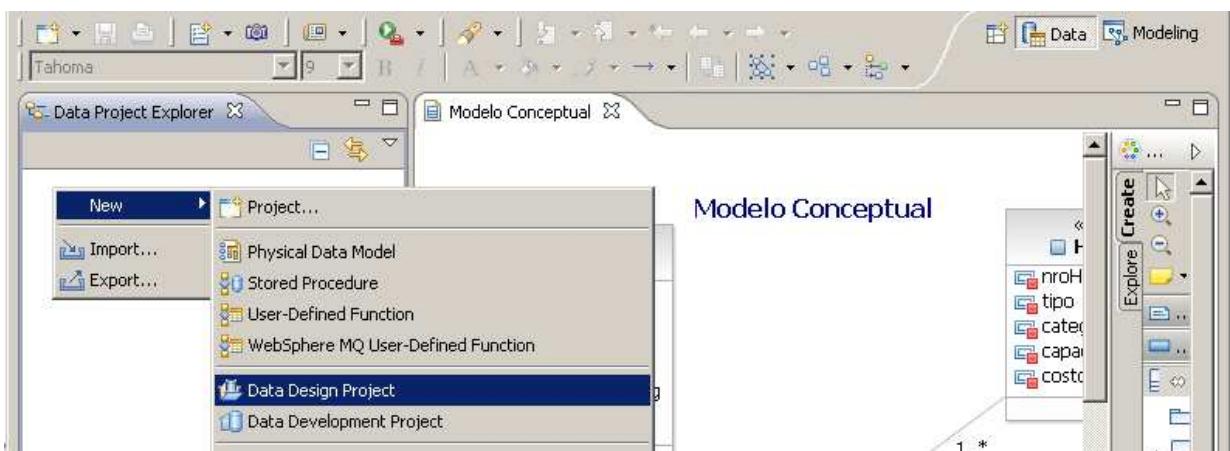
En el **RSA**, para transformar un modelo UML en un modelo lógico de datos, cambie la perspectiva a **Data** para crear un proyecto de diseño de datos. Luego, podrá ejecutar una configuración de transformación UML a LDM.

A continuación, se indican los pasos que debe seguir:

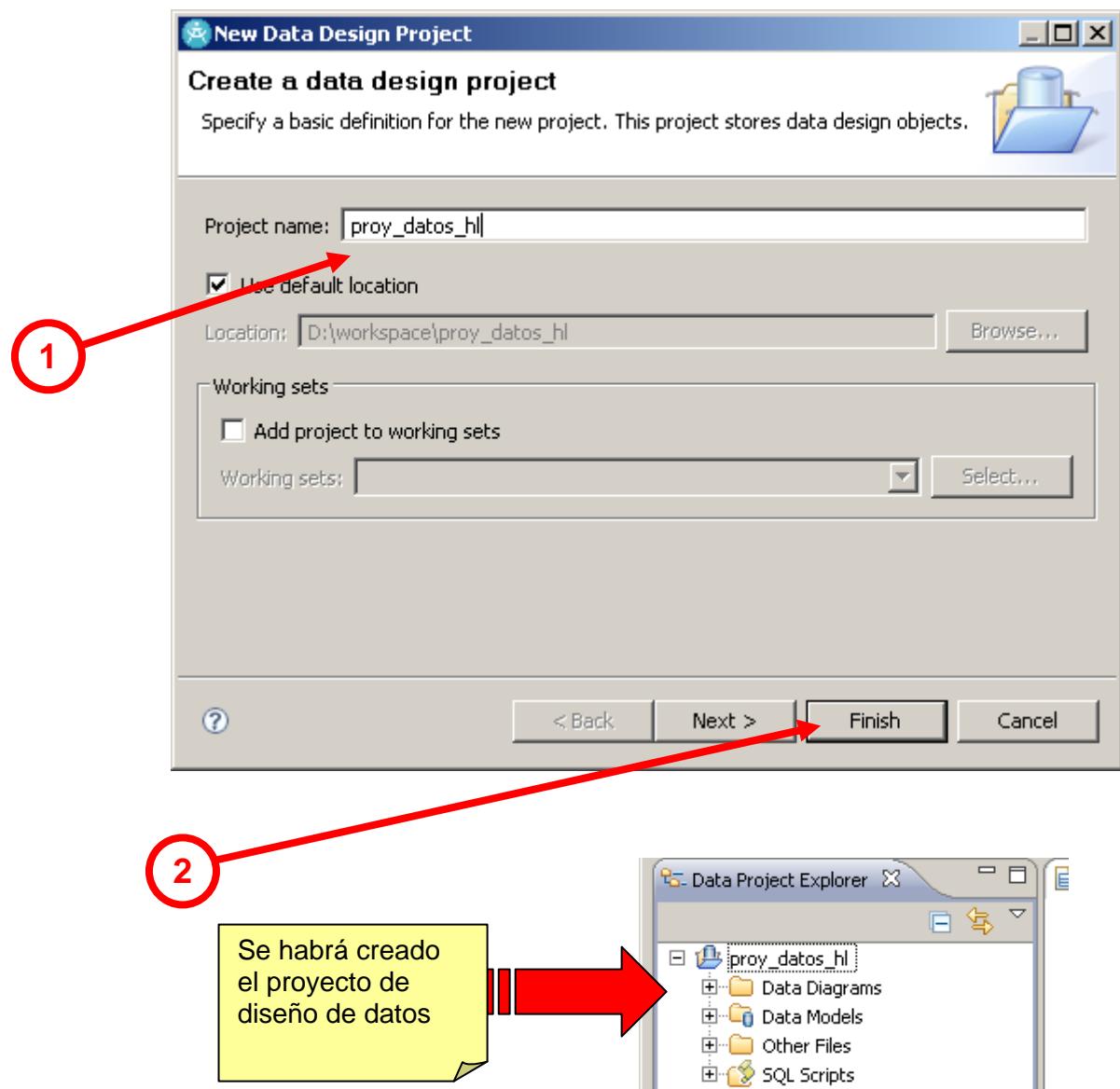
**PASO 1:** Cambie la perspectiva de modelado a datos:



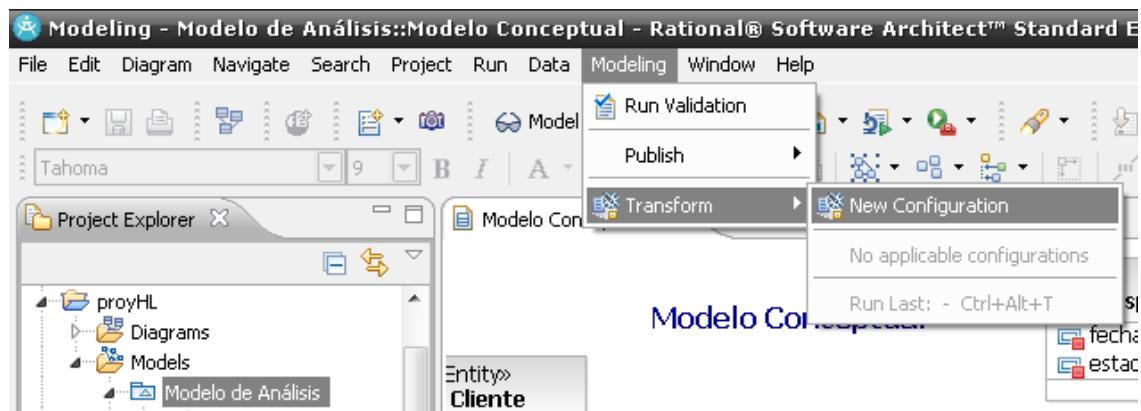
PASO 2: Cree un proyecto de diseño de datos:



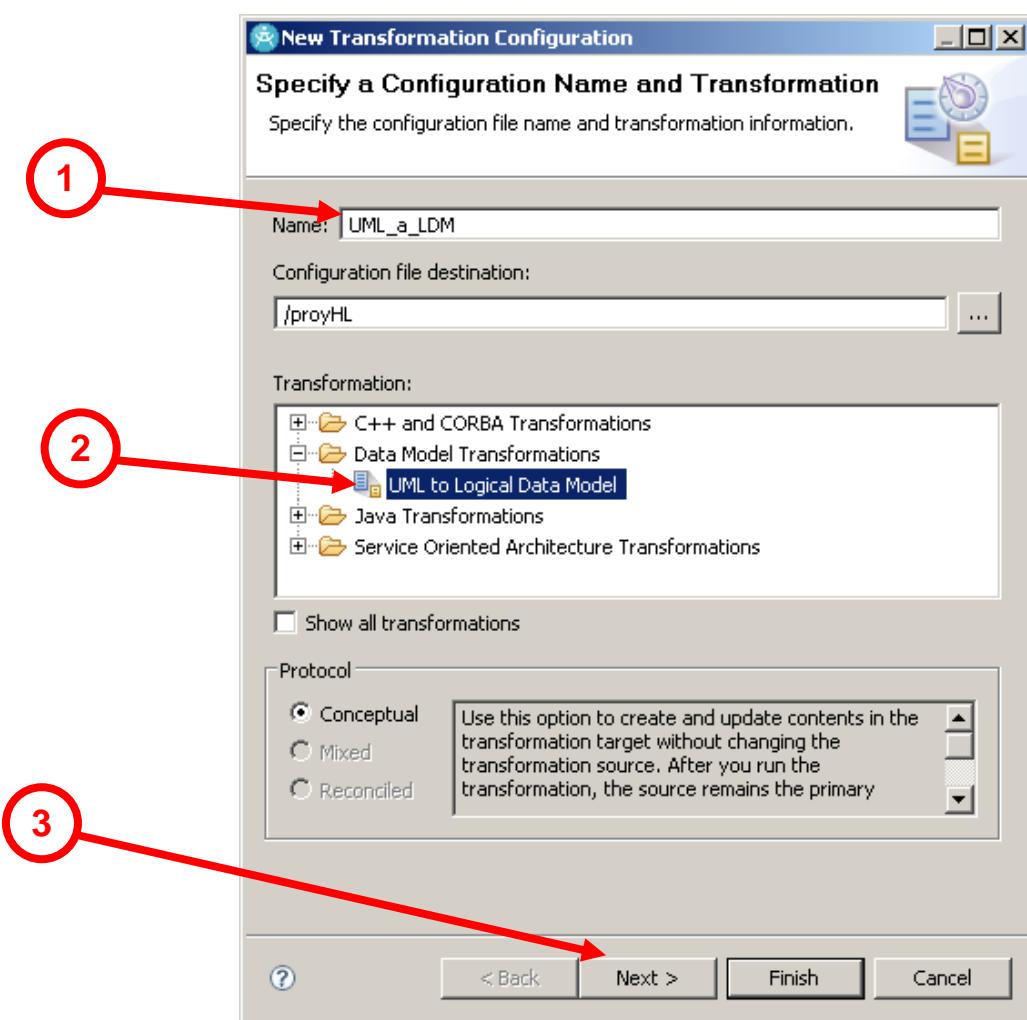
PASO 3: Edite el nombre para el proyecto, en este caso crearemos **proy\_datos\_hl**



**PASO 4:** A continuación, retorne a la perspectiva **Modeling**. Luego, seleccione el **Modelo de Análisis** del explorador de proyectos, el cual contiene el Modelo Conceptual. Luego, pulse **Modelado > Transformación > Nueva configuración** desde el menú principal.

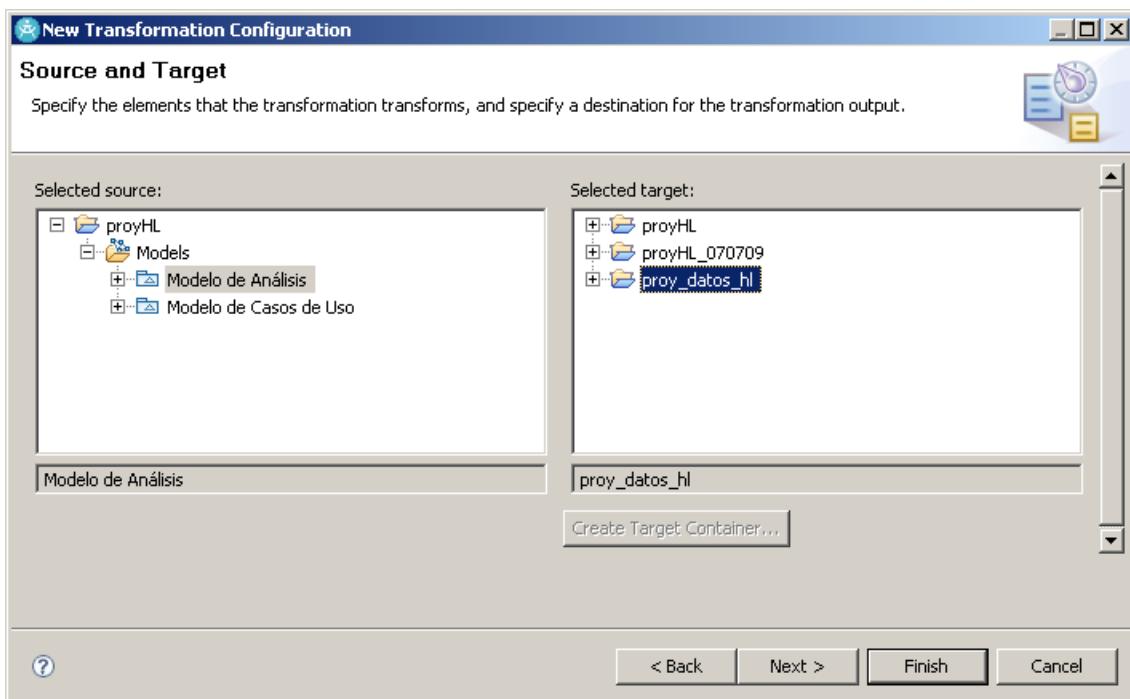


**PASO 5:** En la ventana de transformación, edite el nombre de la transformación, seleccione **UML to Logical Data Model** de la lista **Transformaciones de modelos de datos** y pulse **Next**.

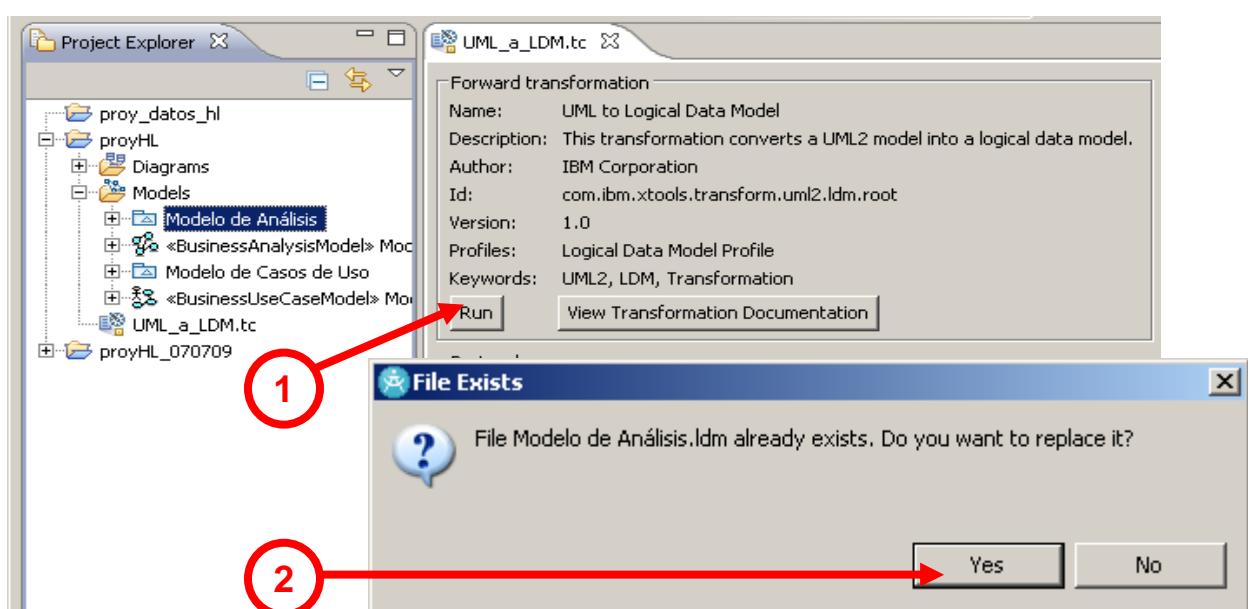


**PASO 6:** En la ventana de Origen y destino, efectúe los siguientes pasos:

- Seleccione el **Modelo de Análisis**, en el campo **Seleccionar un origen**. Luego, seleccione el proyecto de diseño de datos **proy\_datos\_hl** en el campo **Seleccionar un destino**. A continuación, pulse **Finish**.



- En la siguiente página, presione **Run** hasta que se genera el Modelo lógico de datos con éxito.



### 3. MODELO FÍSICO

A continuación, se muestran los pasos para confeccionar un Modelo Físico de Datos con otro caso de estudio; pero usted, para la creación de dicho modelo, considere el Modelo Lógico del caso de estudio Clínica El Ángel.

El esquema de trabajo para crear el modelo físico de datos es el siguiente:

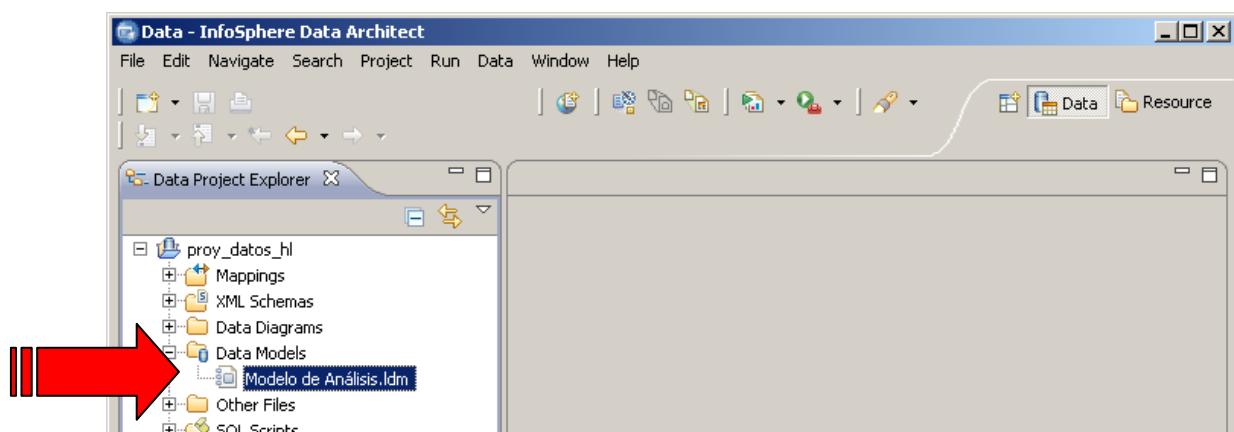
1. Transformación de un modelo lógico en un modelo físico
2. Configuración del script

#### Transformación de un modelo lógico en un modelo físico

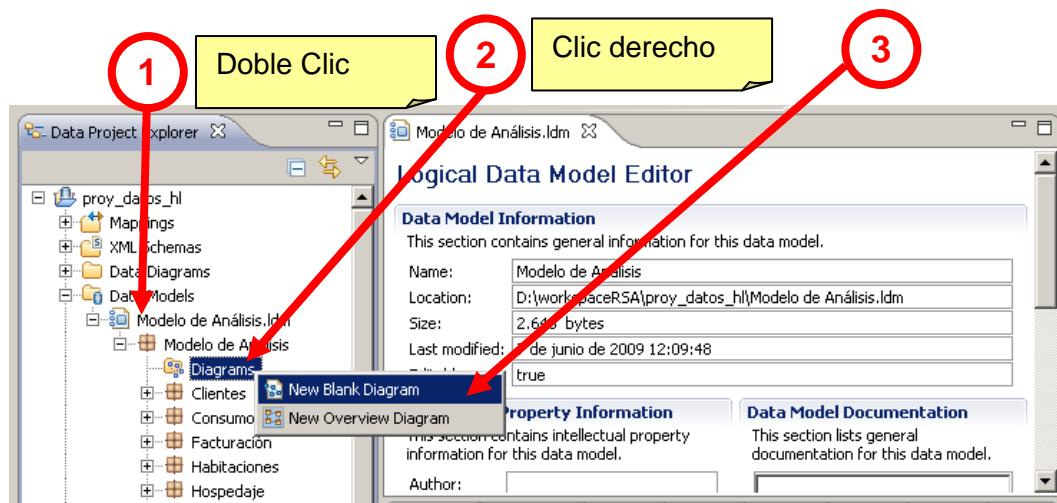
En el **InfoSphere Data Architect**, para transformar un modelo lógico en un modelo físico, importe el proyecto de diseño de datos, creado anteriormente, después de haber cambiado a la perspectiva **Data**.

A continuación, se indican los pasos que debe seguir:

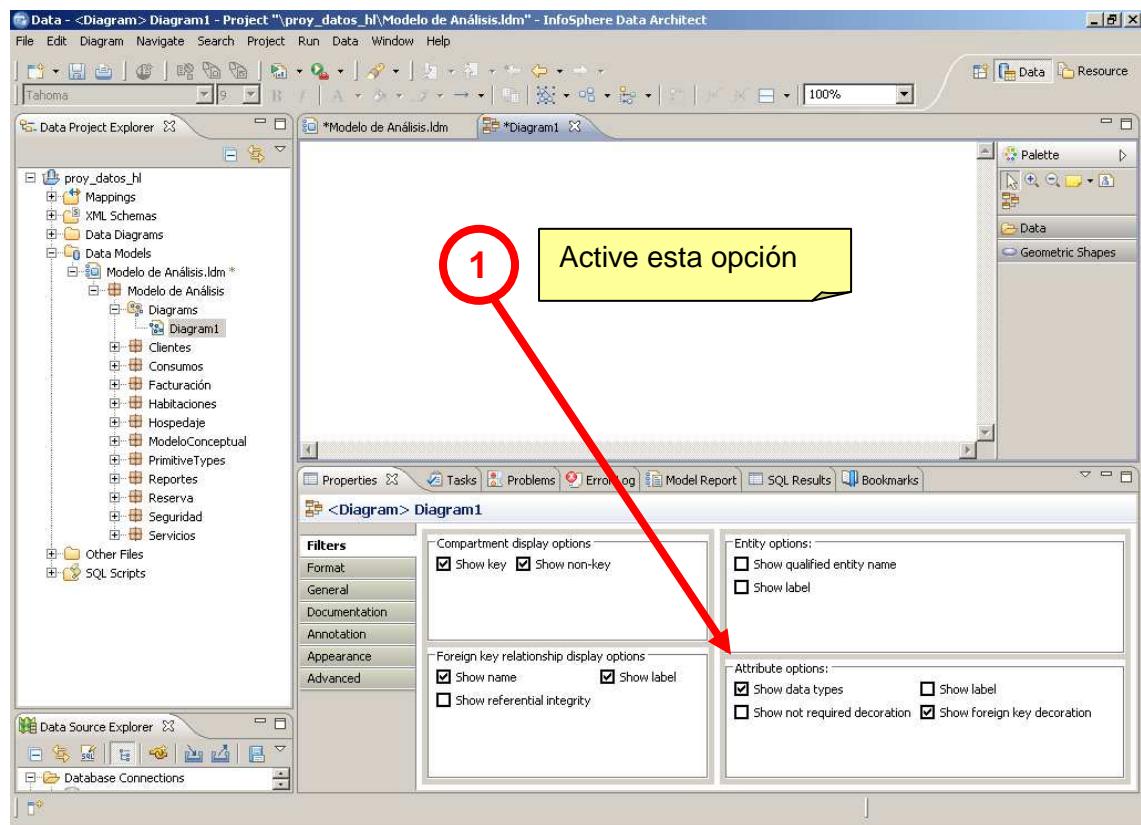
**PASO 1:** Cambie la perspectiva a **Data** e importe el proyecto para visualizar el modelo lógico:



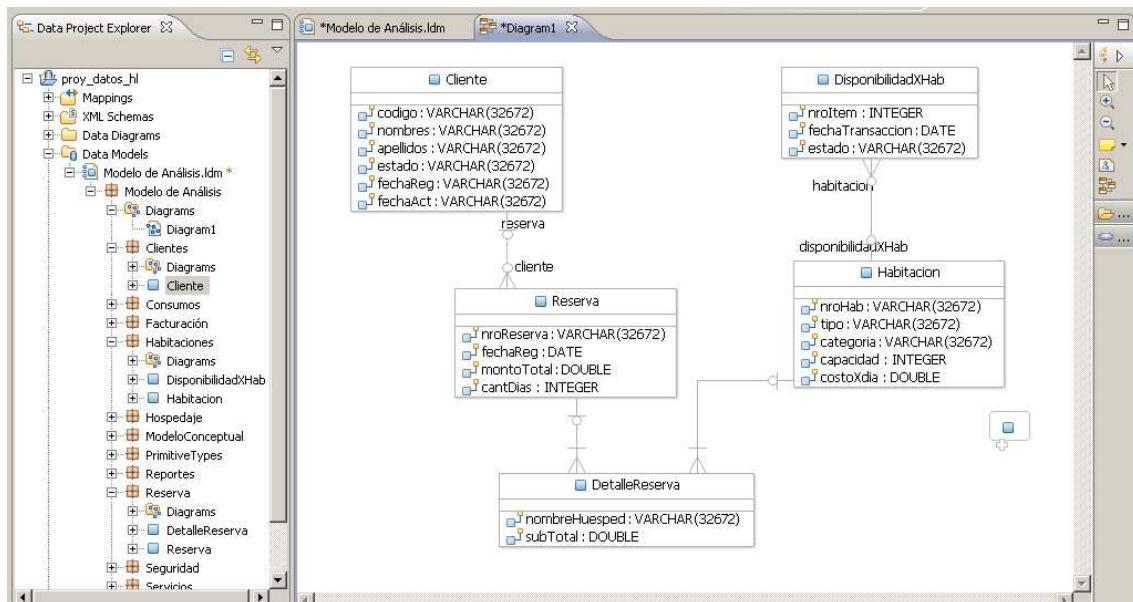
**PASO 2:** De doble clic sobre el modelo lógico para agregar un diagrama en blanco:



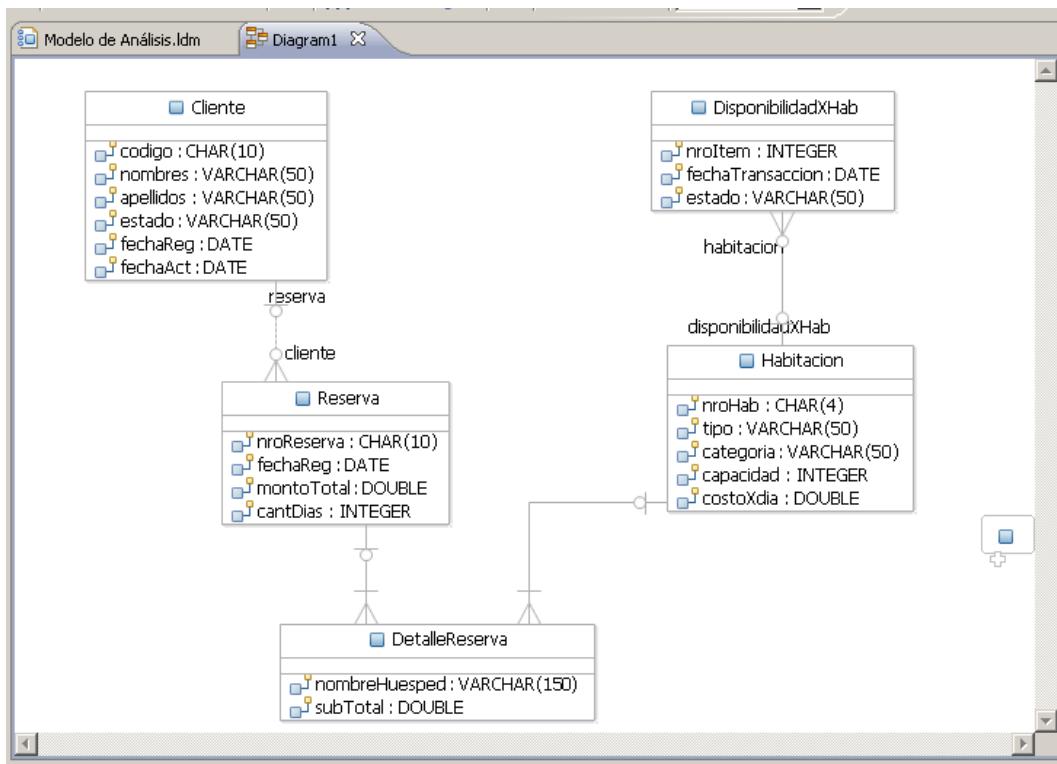
**PASO 3:** Active la opción para visualizar los tipos de datos.



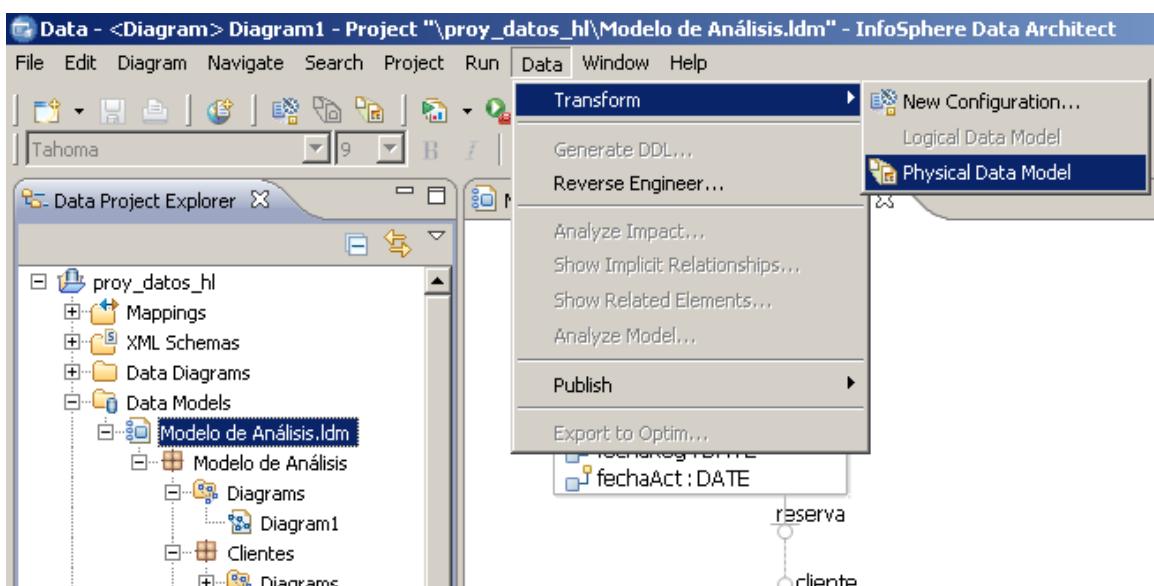
**PASO 4:** Arrastre todas las entidades, generadas en el modelo lógico, al diagrama creado para visualizar el modelo lógico:



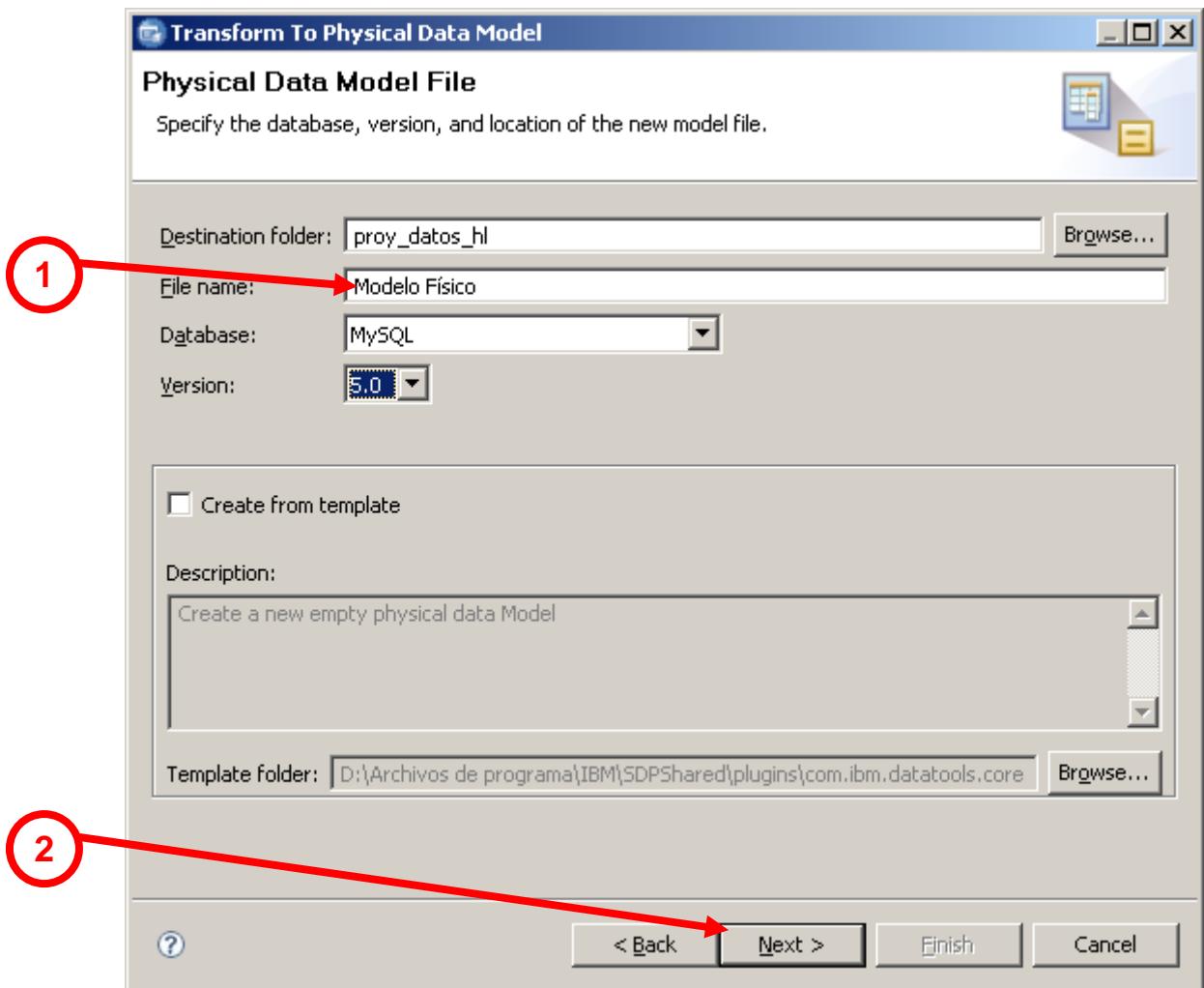
**PASO 5:** A continuación, modifique los tipos y/o longitudes de algunos campos. Al final, debe quedar así:



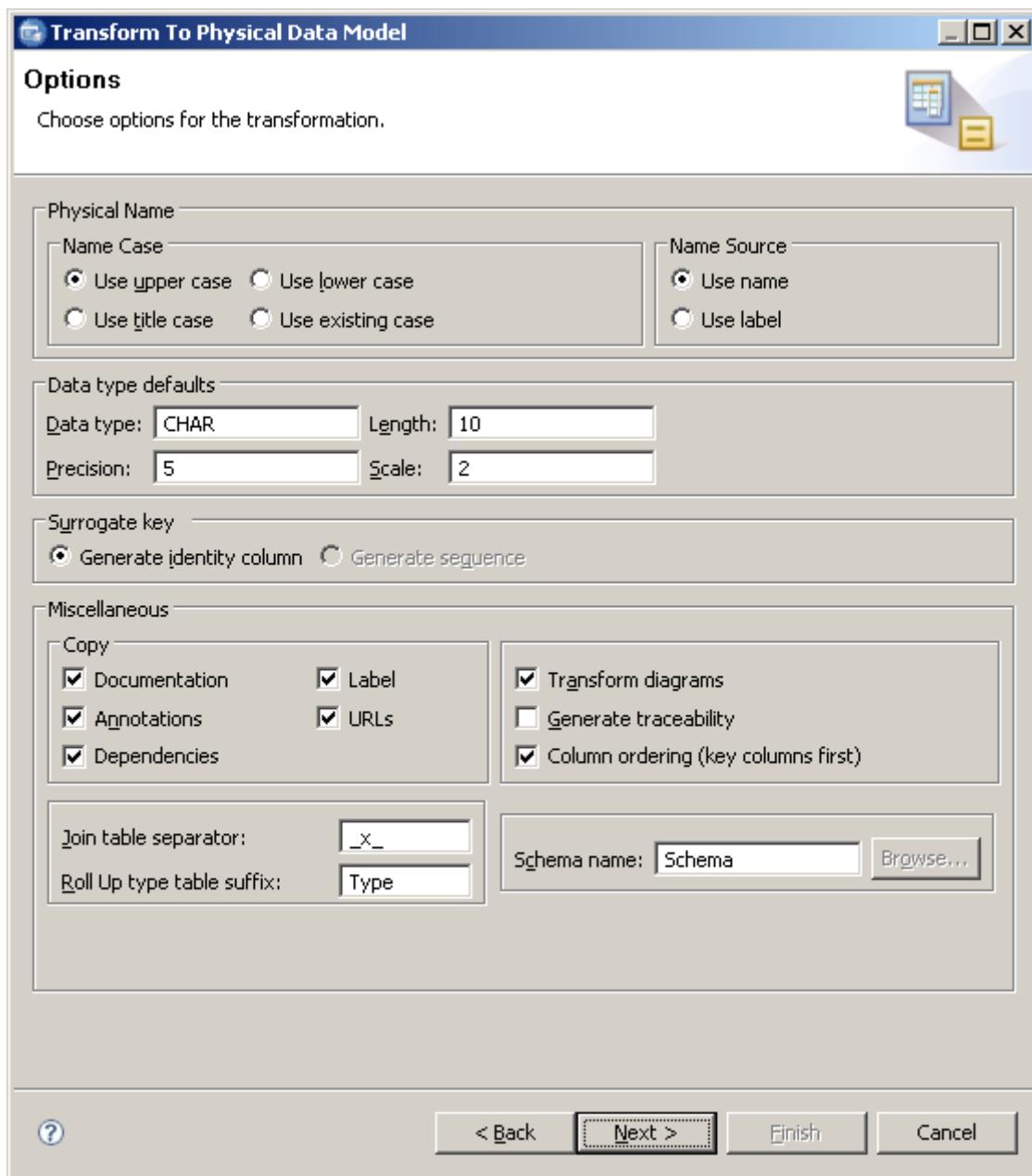
**PASO 6:** Pulse Datos > Transformación > **Modelo de Datos Físico** desde el menú principal.



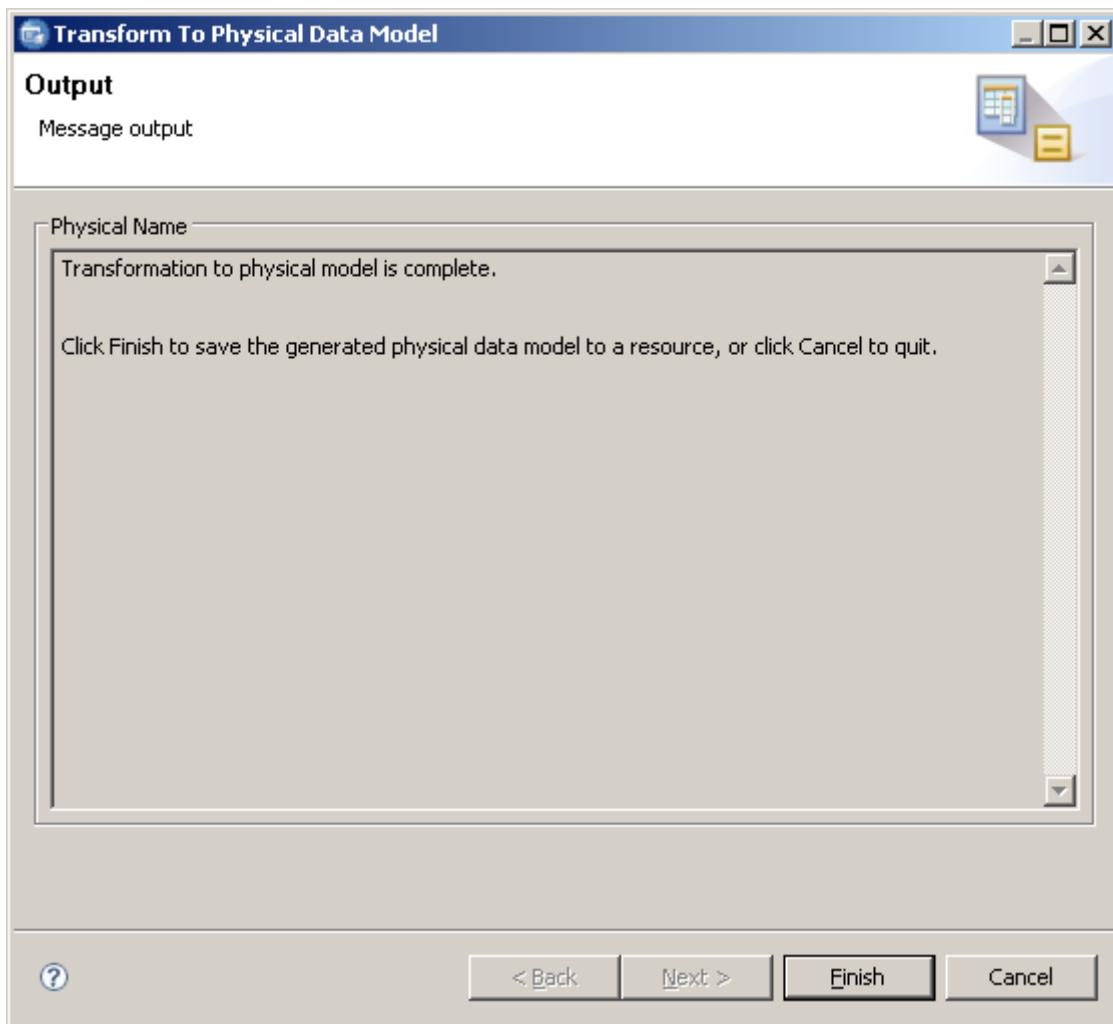
**PASO 7:** En la primera ventana de transformación pulse **Next**. A continuación, en esta ventana edite el nombre del modelo y luego pulse **Next**.



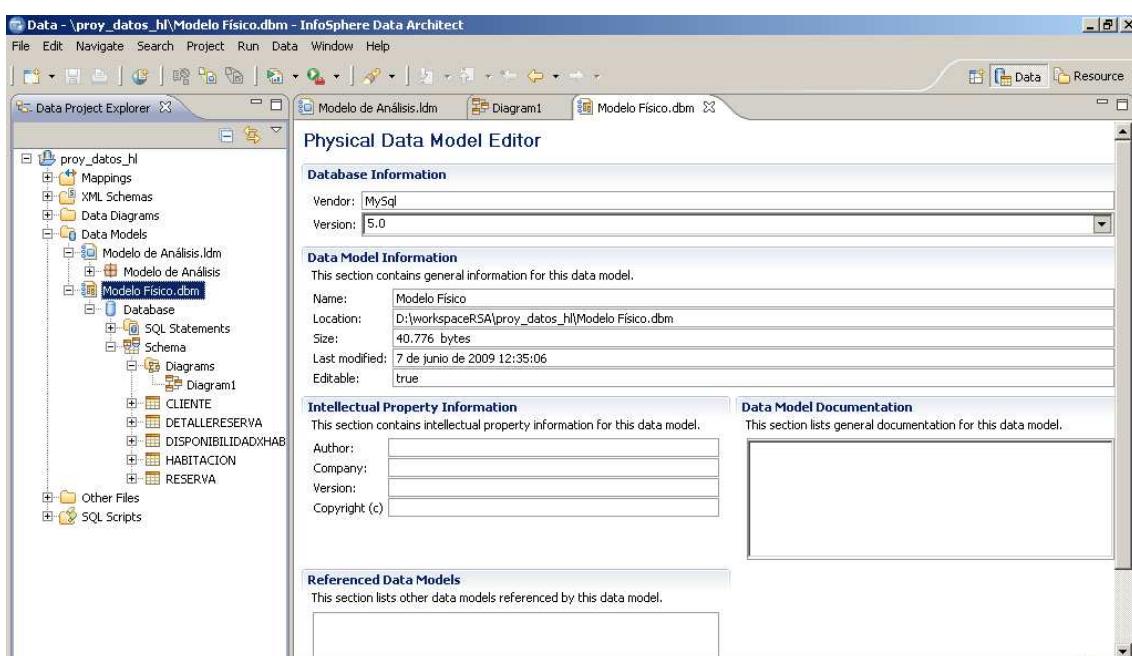
**PASO 8:** En esta ventana, es opcional cambiar la configuración presentada por defecto (si desea, cambie el nombre del esquema). A continuación, pulse **Siguiente**.



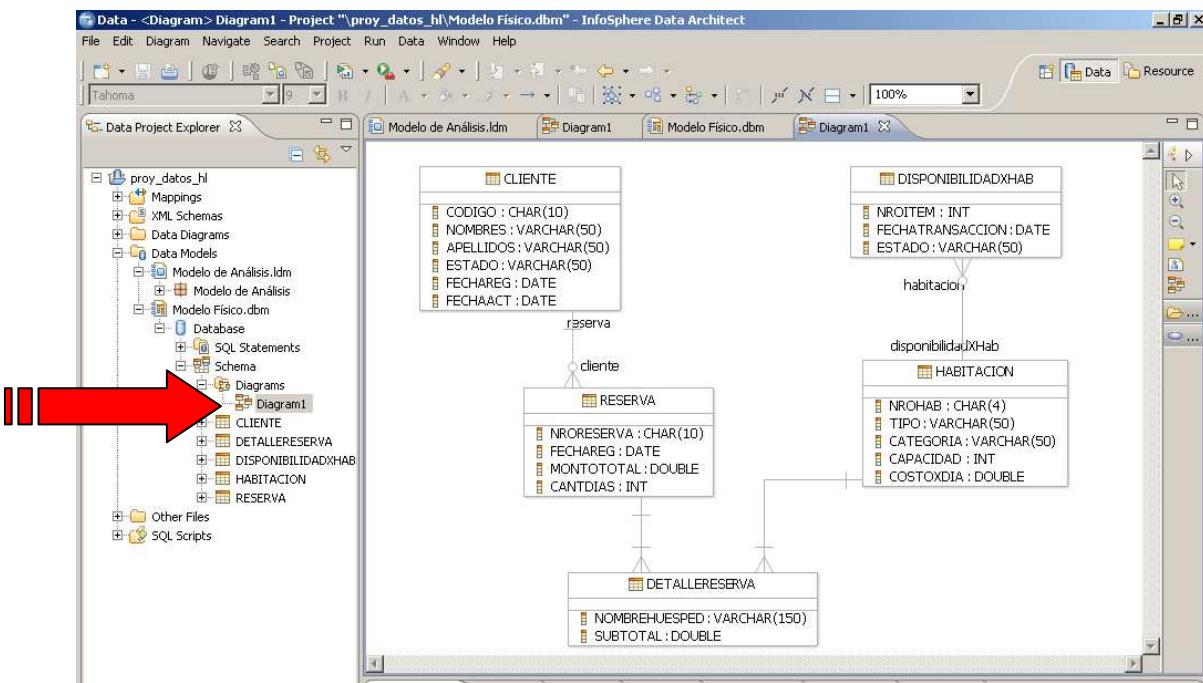
PASO 9: En esta ventana, si no aparece ningún warning, pulse **Finish**.



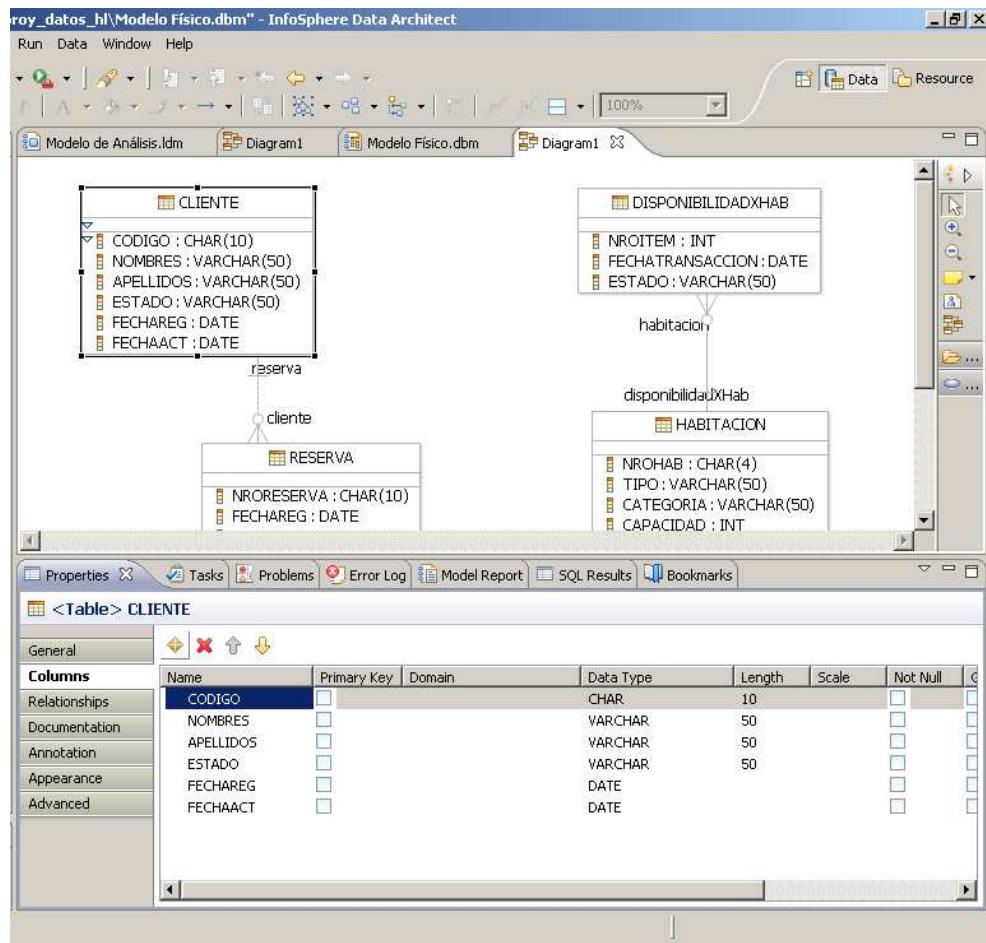
Se habrá creado el modelo físico:



**PASO 10:** Para visualizar el modelo físico, seleccione el diagrama generado.



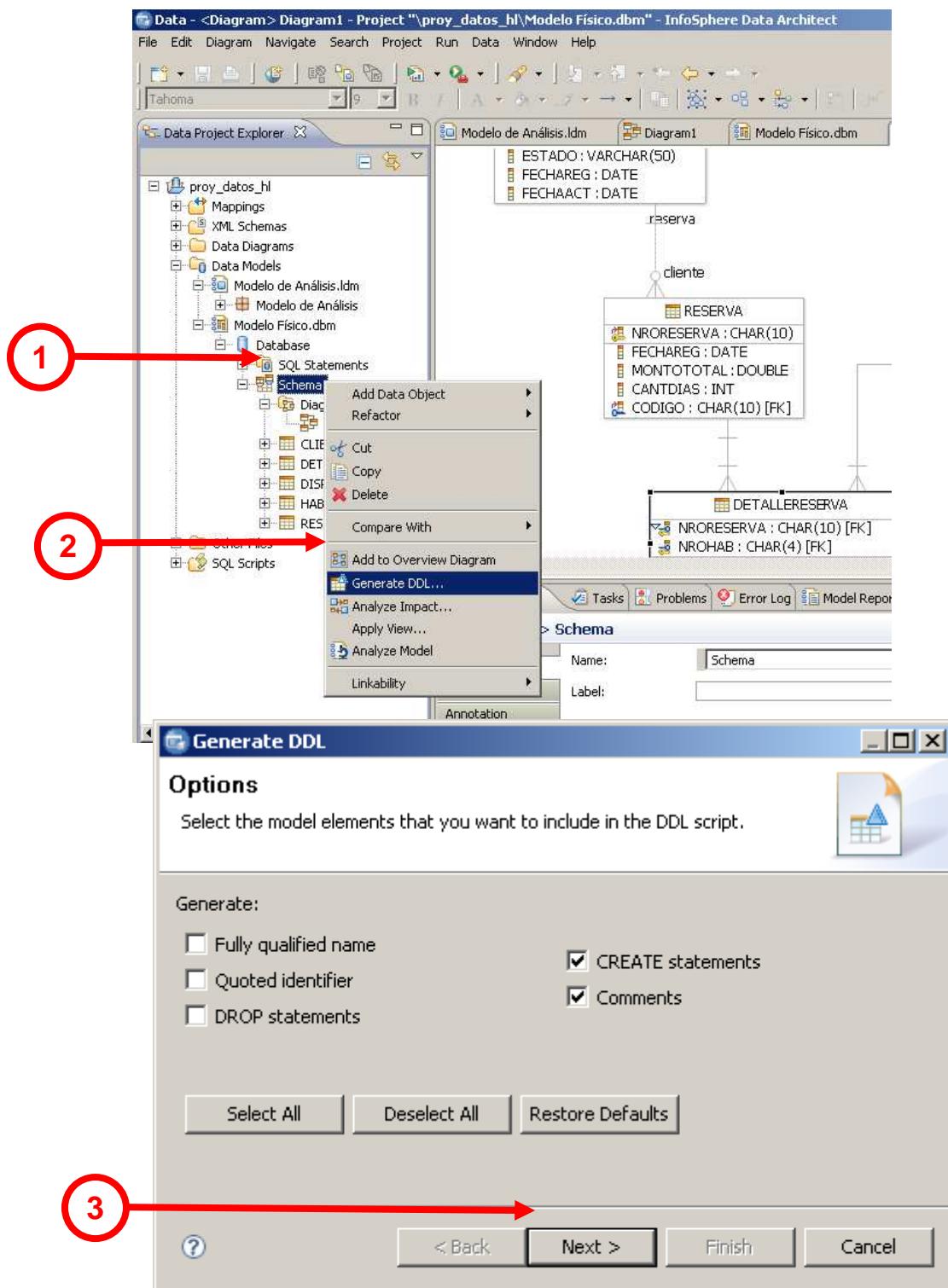
**PASO 11:** Seleccione cada tabla generada y especificar las llaves primarias. Para ello, utilice la **vista de propiedades**. Empiece con las tablas independientes, luego con las dependientes. Además, puede ordenar los campos internamente utilizando la vista de propiedades, opción columnas.



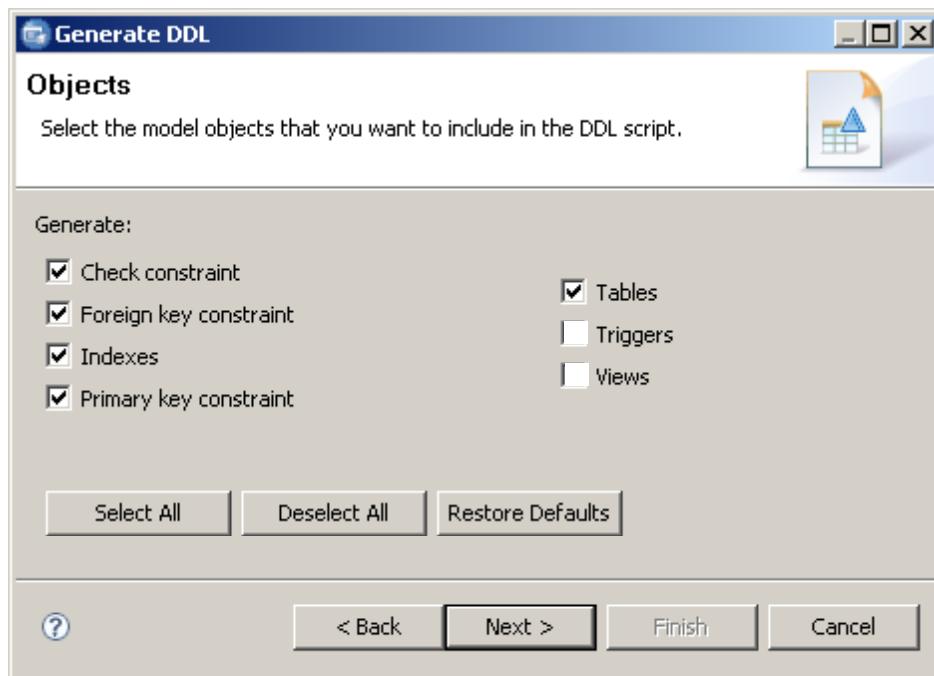
## Generación del script

En el **InfoSphere Data Architect**, para generar el script de la base datos, debe seguir los pasos que se indican a continuación:

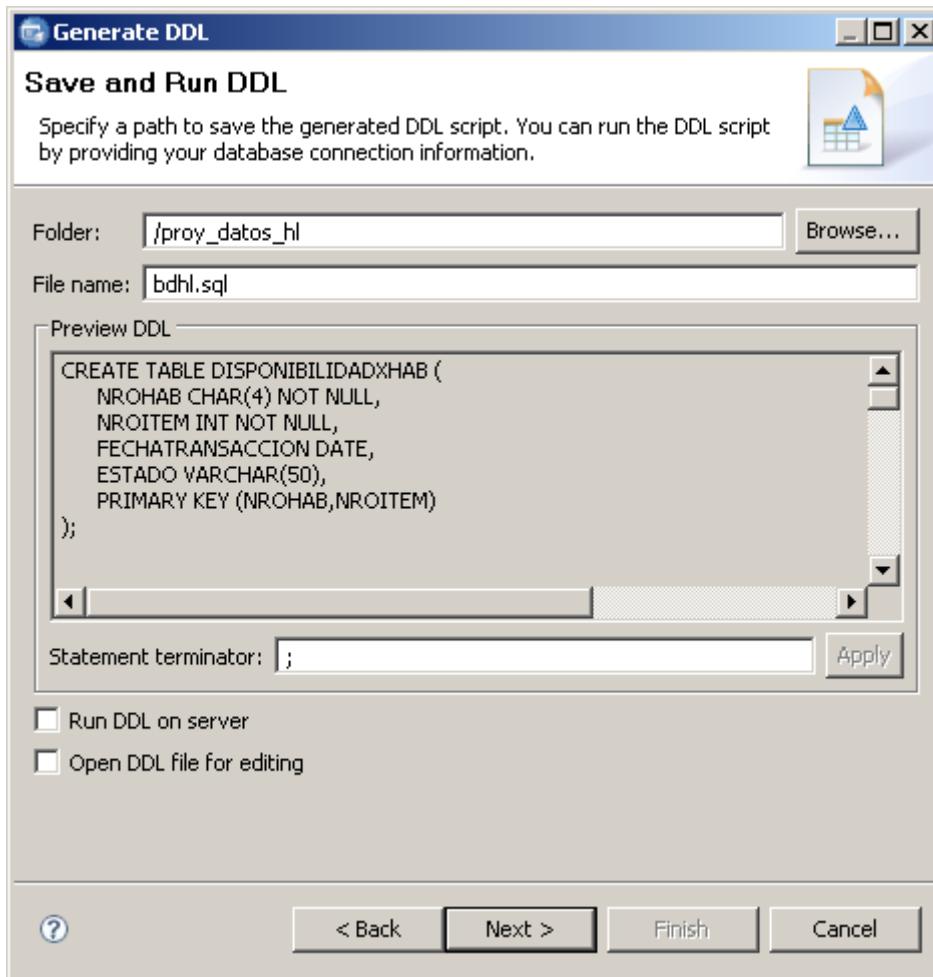
**PASO 1:** Seleccione **Generar DDL** a partir del esquema generado:



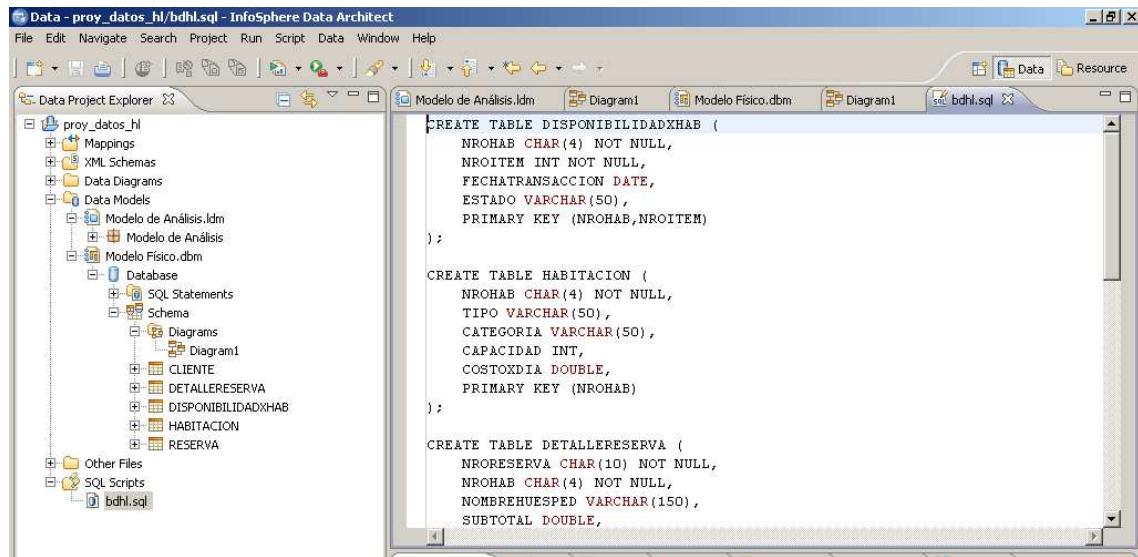
**PASO 2:** Desactive las opciones **Triggers** y **Views**. Pulse **Next**:



**PASO 3:** Pulse **Next**:



**PASO 4:** Por último, se mostrará el script de la base de datos:



The screenshot shows the InfoSphere Data Architect interface. The left pane is the Data Project Explorer, displaying a project named 'proy\_datos\_hl'. Inside the project, there are several nodes: 'Mappings', 'XML Schemas', 'Data Diagrams', 'Data Models', 'Other Files', and 'SQL Scripts'. Under 'SQL Scripts', there is a file named 'bdhl.sql'. The right pane contains the SQL code for creating three tables: 'DISPONIBILIDADXHAB', 'HABITACION', and 'DETALLERESERVA'. The code is as follows:

```
CREATE TABLE DISPONIBILIDADXHAB (
    NROHAB CHAR(4) NOT NULL,
    NROITEM INT NOT NULL,
    FECHATRANSACCION DATE,
    ESTADO VARCHAR(50),
    PRIMARY KEY (NROHAB,NROITEM)
);

CREATE TABLE HABITACION (
    NROHAB CHAR(4) NOT NULL,
    TIPO VARCHAR(50),
    CATEGORIA VARCHAR(50),
    CAPACIDAD INT,
    COSTOXDIA DOUBLE,
    PRIMARY KEY (NROHAB)
);

CREATE TABLE DETALLERESERVA (
    NRORESERVA CHAR(10) NOT NULL,
    NROHAB CHAR(4) NOT NULL,
    NOMBREHUESPED VARCHAR(150),
    SUBTOTAL DOUBLE,
```





# DISEÑO ORIENTADO A OBJETOS

## LOGRO DE LA UNIDAD DE APRENDIZAJE

Al finalizar la tercera unidad, el alumno diseña la arquitectura del *software* identificando las capas, subsistemas y componentes de la aplicación. Los artefactos serán creados utilizando la herramienta *CASE IBM Rational Software Architect (RSA)*.

## TEMARIO

Tema 1: Capas lógicas de la arquitectura

1. Definición de capas
2. Creación de subsistemas
3. Creación de componentes.

Tema 2: Diseño de casos de uso con patrón arquitectónico MVC

1. Realizaciones de diseño de casos de uso
2. Diagrama de Clases
3. Diagrama de Secuencia del flujo básico y subflujos de un mantenimiento.

Tema 3: Diseño de casos de uso con patrón arquitectónico MVC y patrón de diseño DAO

4. Realizaciones de diseño de casos de uso
5. Diagrama de Clases
6. Diagrama de Secuencia del flujo básico y subflujos de un mantenimiento.

## ACTIVIDADES PROPUESTAS

1. Los alumnos desarrollan las realizaciones de análisis de un caso de uso propuesto.

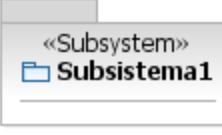
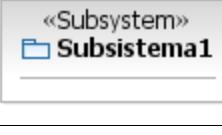
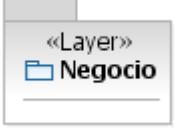
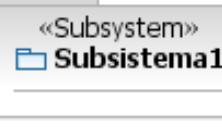
## 1. CAPAS LÓGICAS DE LA ARQUITECTURA

Antes de indicar cómo se realiza las realizaciones de diseño de un caso de uso, primero se mostrará la organización de los componentes (clases de diseño e interfaces) en capas, subsistemas y librerías que utilizaremos en el curso, aplicando patrón arquitectónico MVC:

Capa	Subsistema/Librerías	Componentes
	 	Clases estereotipadas: <ul style="list-style-type: none"> <li>• Páginas HTML: &lt;&lt;Client Page&gt;&gt; y &lt;&lt;HTML Form&gt;&gt;</li> <li>• Páginas JSP: &lt;&lt;Server Page&gt;&gt;, &lt;&lt;Client Page&gt;&gt; y &lt;&lt;HTML Form&gt;&gt;</li> </ul>
		Clase estereotipada para servlets: <<Http Servlet>>
		Clases de diseño: beans.
		Clases de diseño: clases utilitarias.

**Tabla 3.1. Capas, subsistemas, librerías y elementos de diseño según patrón arquitectónico MVC.**

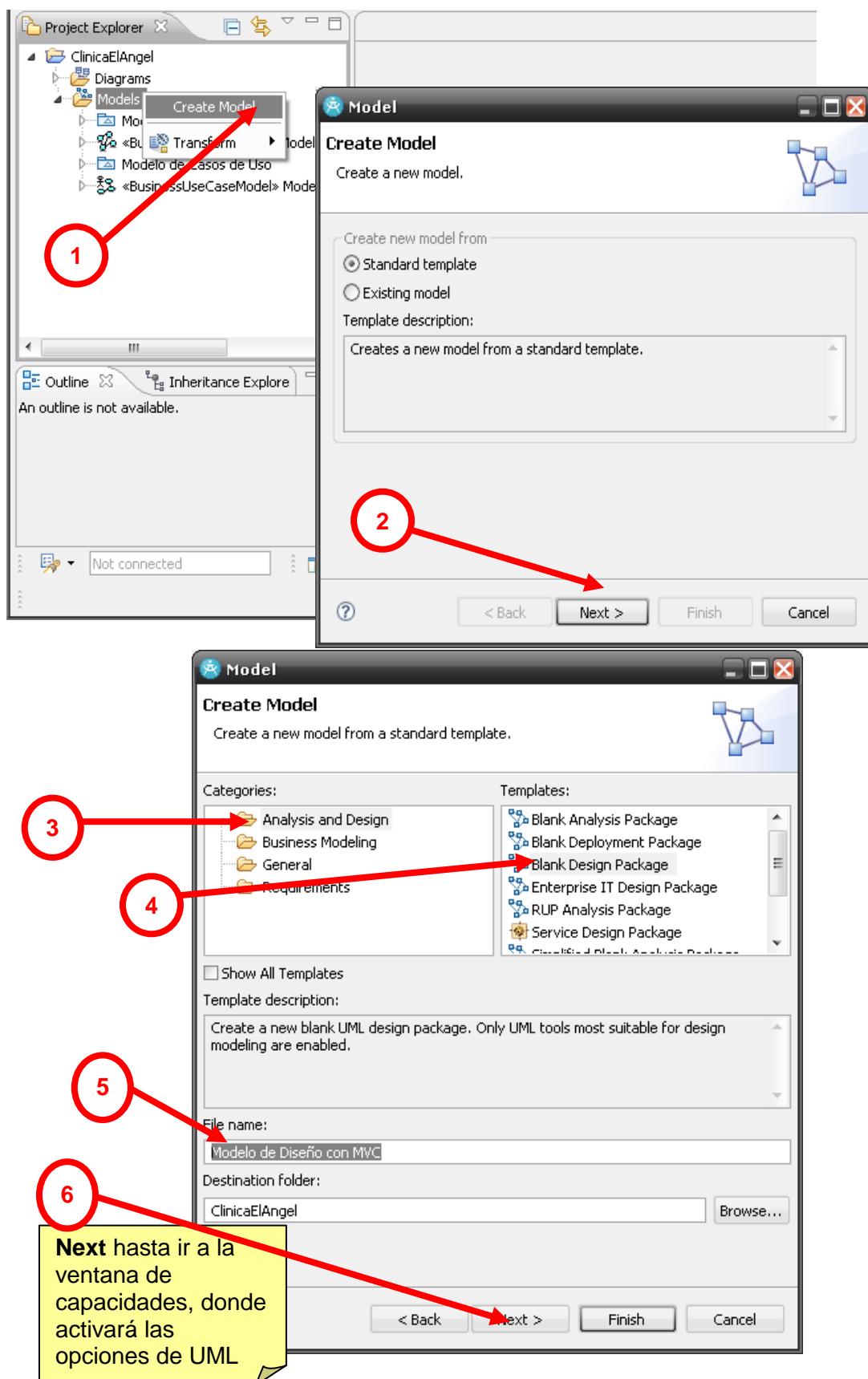
En la siguiente tabla, se muestra la organización de las clases de diseño e interfaces en capas, subsistemas y librerías que utilizaremos en el curso, aplicando patrón arquitectónico MVC y patrón de diseño DAO:

Capa	Subsistema/Librerías	Componentes
	 	Clases estereotipadas: <ul style="list-style-type: none"> <li>• Páginas HTML: &lt;&lt;Client Page&gt;&gt; y &lt;&lt;HTML Form&gt;&gt;</li> <li>• Páginas JSP: &lt;&lt;Server Page&gt;&gt;, &lt;&lt;Client Page&gt;&gt; y &lt;&lt;HTML Form&gt;&gt;</li> </ul>
		Clase estereotipada para servlets: <<Http Servlet>>
		<ul style="list-style-type: none"> <li>• Clases de diseño: servicios, beans y clases DAO.</li> <li>• Interfaces que presentan las operaciones de acceso a una tabla.</li> </ul>
		Clases de diseño: clase abstracta <i>DAOFactory</i> y sus clases hijas.
		Clases de diseño: clases utilitarias.

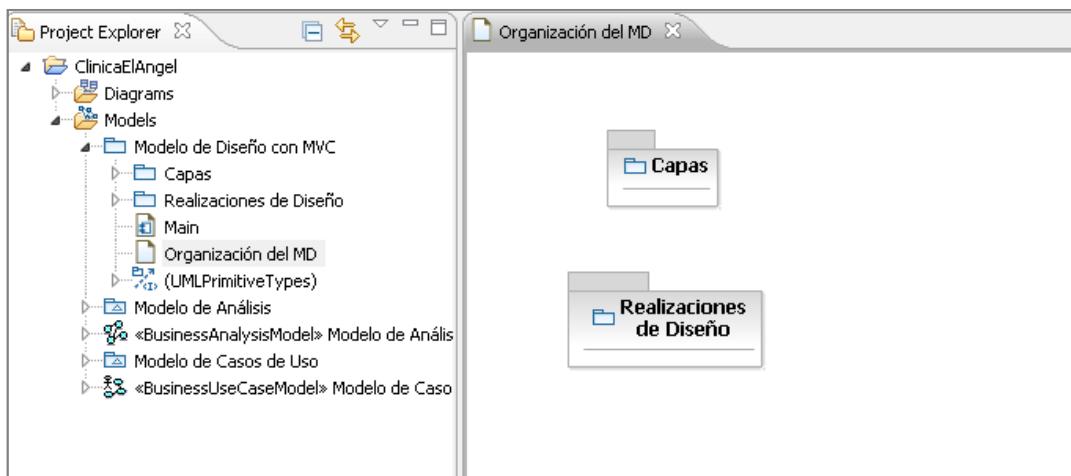
**Tabla 3.2. Capas, subsistemas, librerías y elementos de diseño según patrón arquitectónico MVC y patrón DAO.**

## Definición de capas, subsistemas y componentes de diseño según MVC

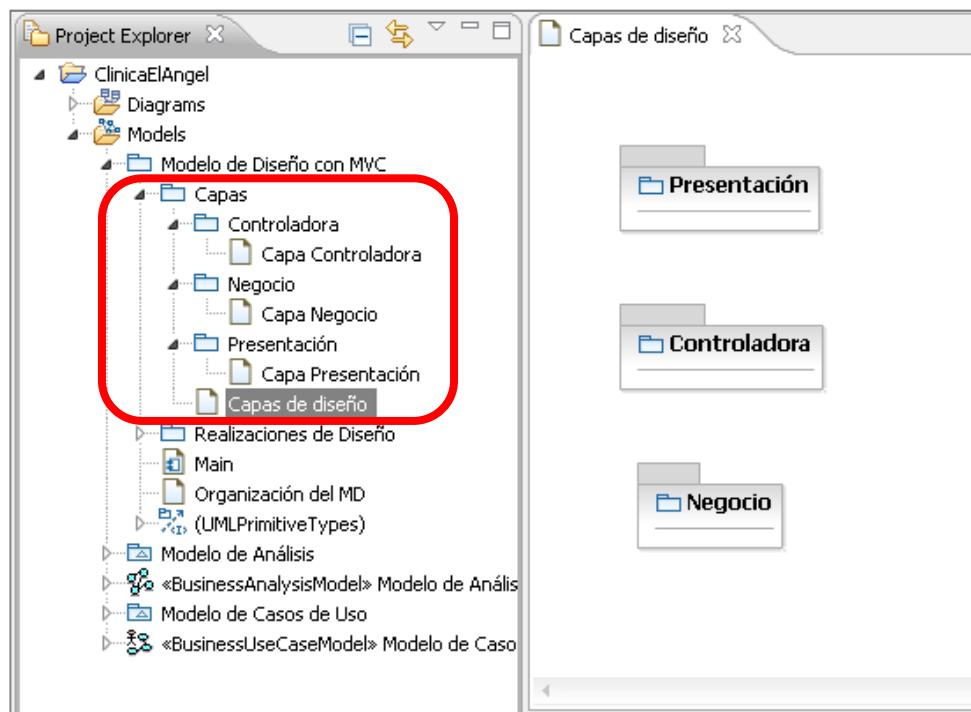
1. En la vista de explorador de proyectos, crear el Modelo de Diseño.



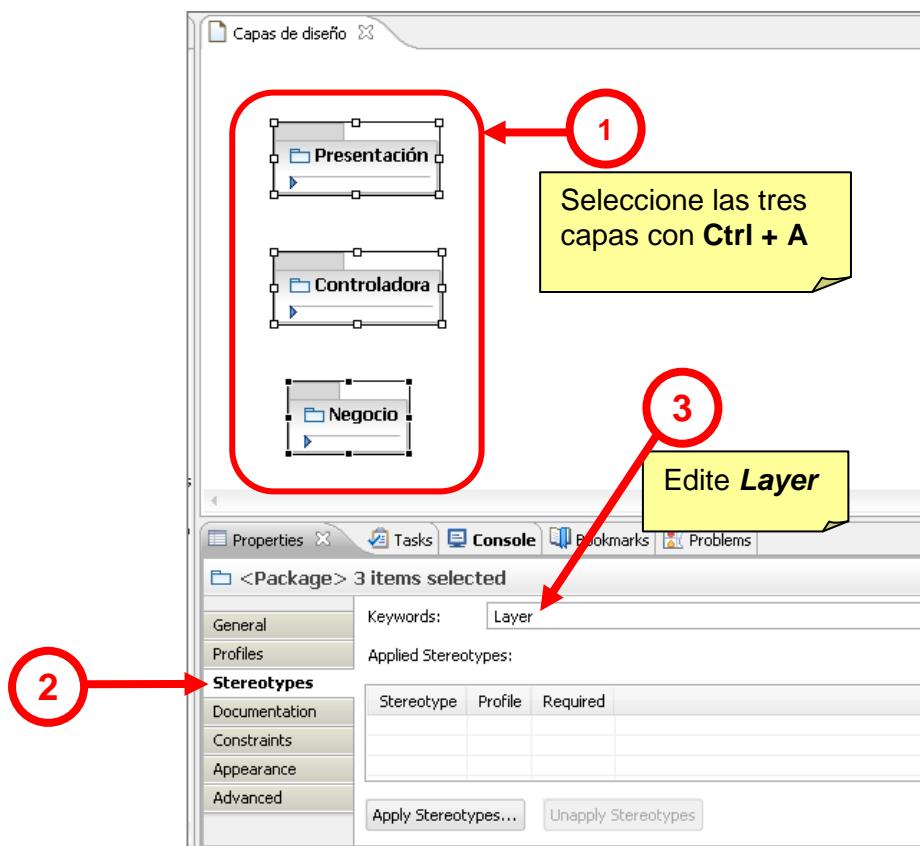
2. Cree un diagrama de formato libre de nombre “Organización del MD” y agregue dos paquetes: Capas y Realizaciones de Diseño.



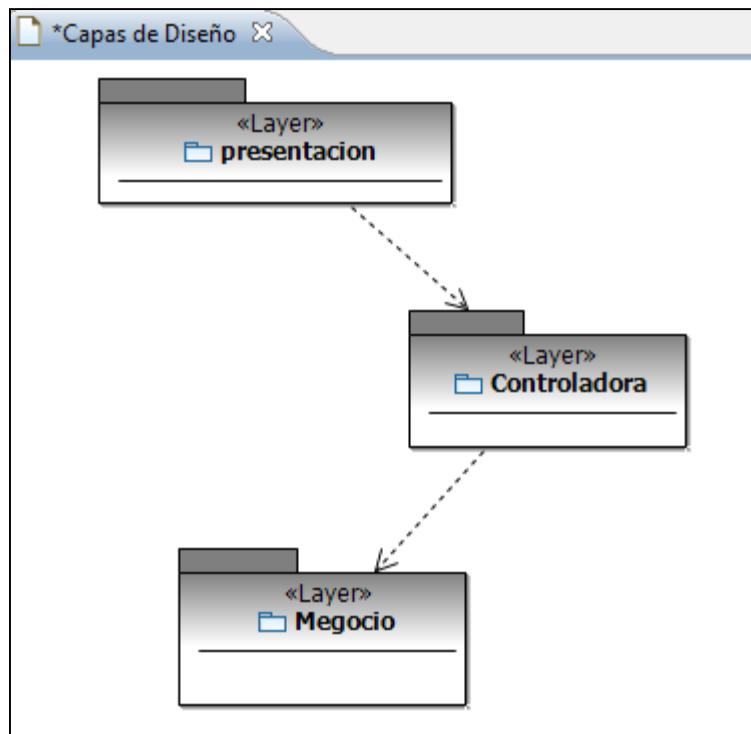
3. En Capas renombre el diagrama de formato libre a “Capas de diseño” y agregue tres paquetes: Presentación, Controladora y Negocio. Luego, renombre los diagramas de cada paquete, así:



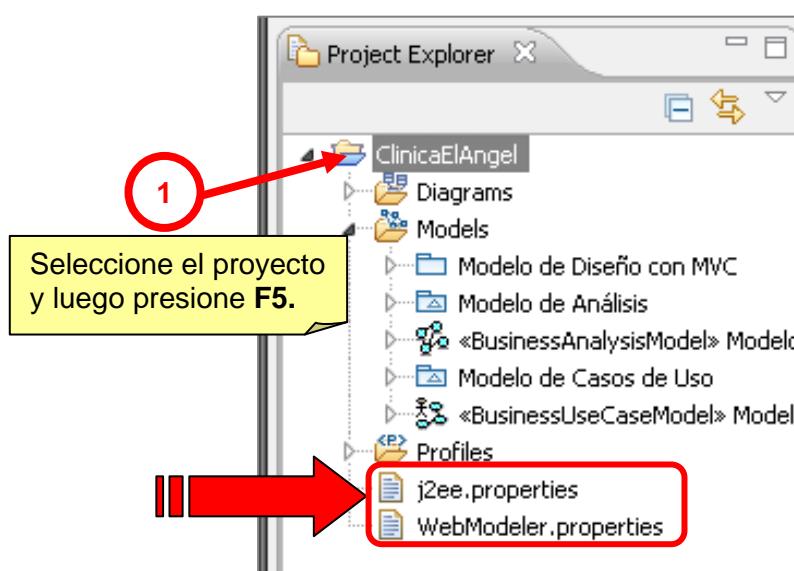
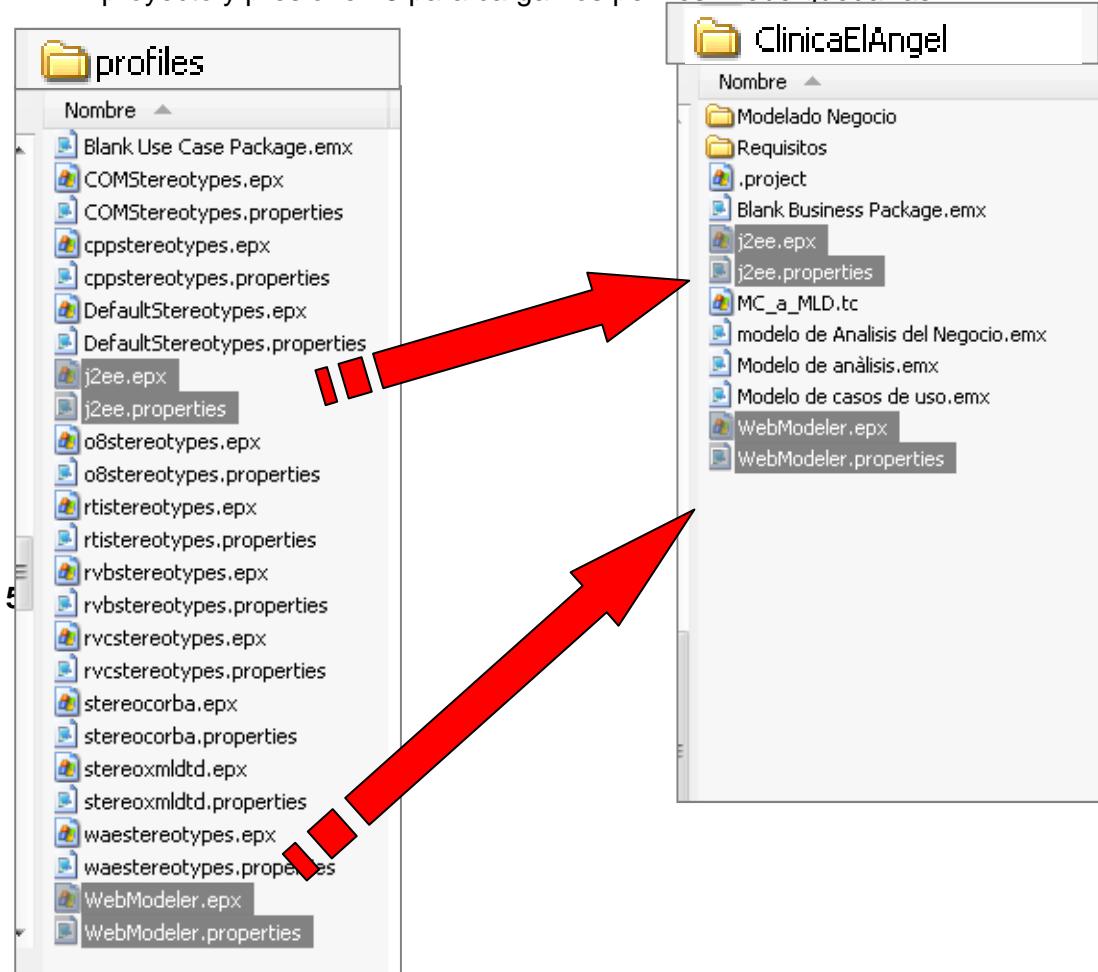
4. Cambie el estereotipo de las capas a *Layer* y relaciona las capas con dependencia.



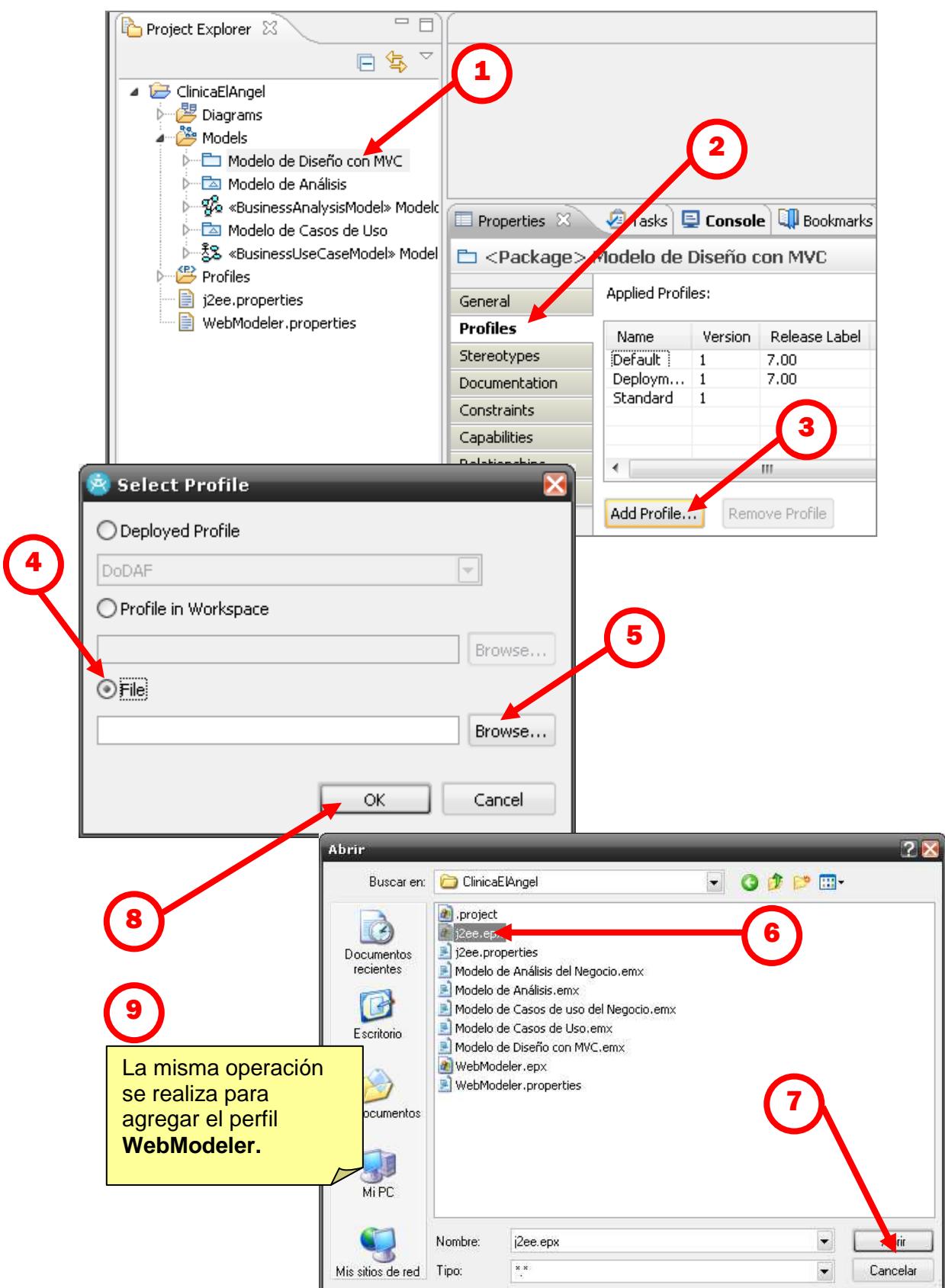
Se habrá cambiado el estereotipo de las capas:



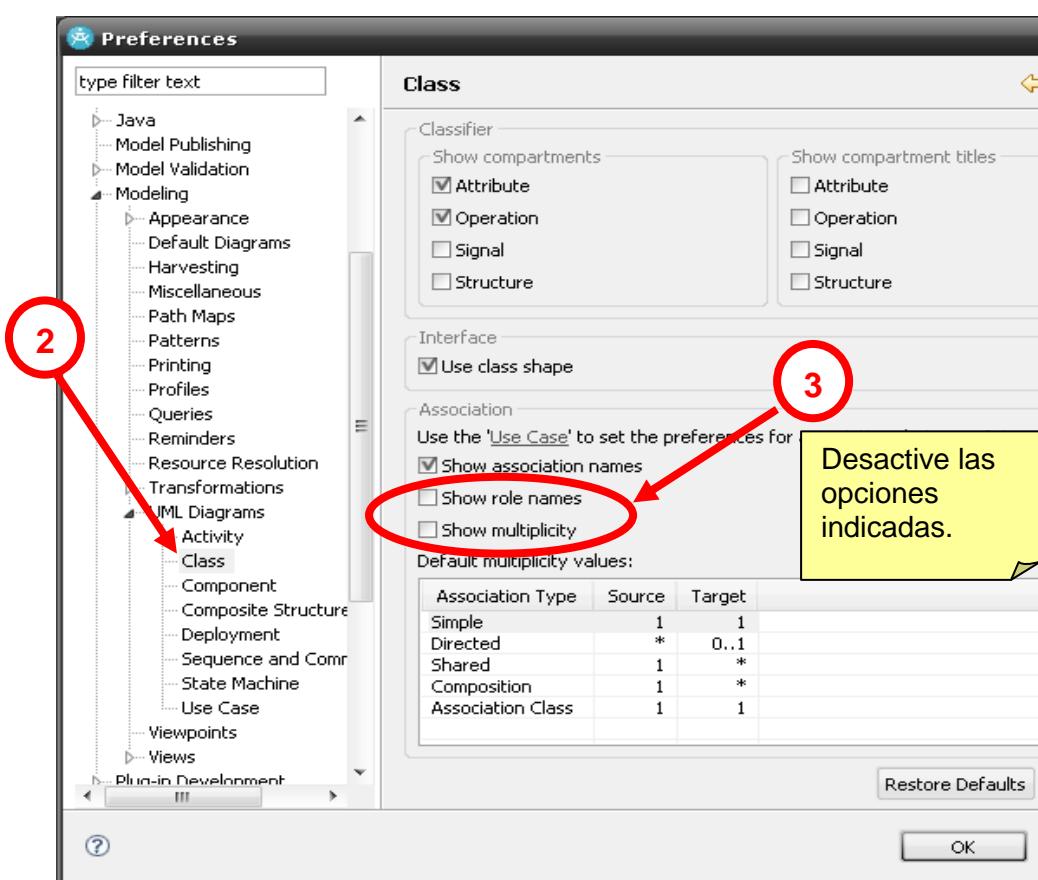
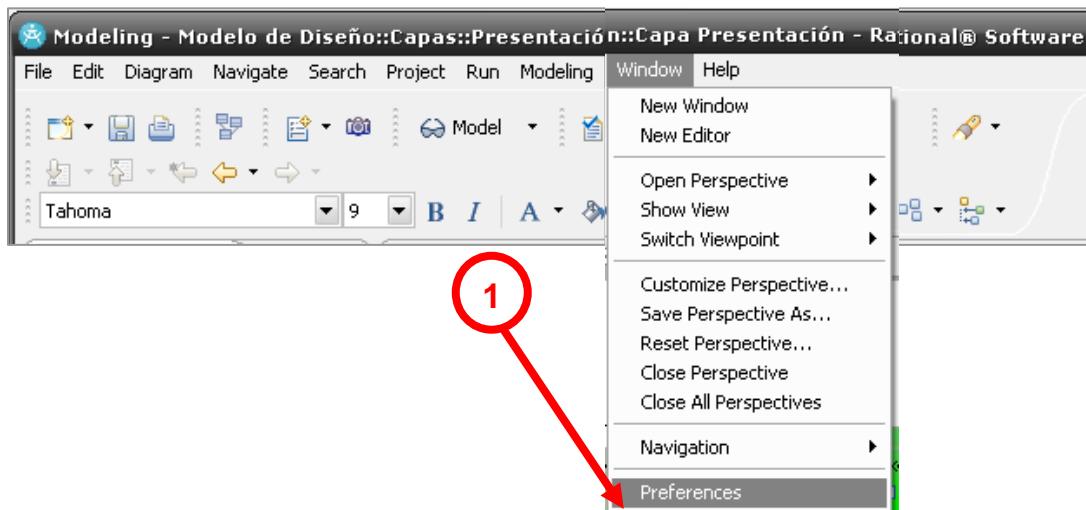
5. Antes de empezar a crear los elementos de diseño asigne los perfiles correspondientes.
- 5.1. Desde el explorador de windows, copie los perfiles **j2ee** y **WebModeler** de la carpeta *profiles* a la carpeta del proyecto:
  - 5.2. A continuación, desde el explorador de proyectos en el RSA, seleccione el proyecto y presione **F5** para cargar los perfiles. Debe quedar así:



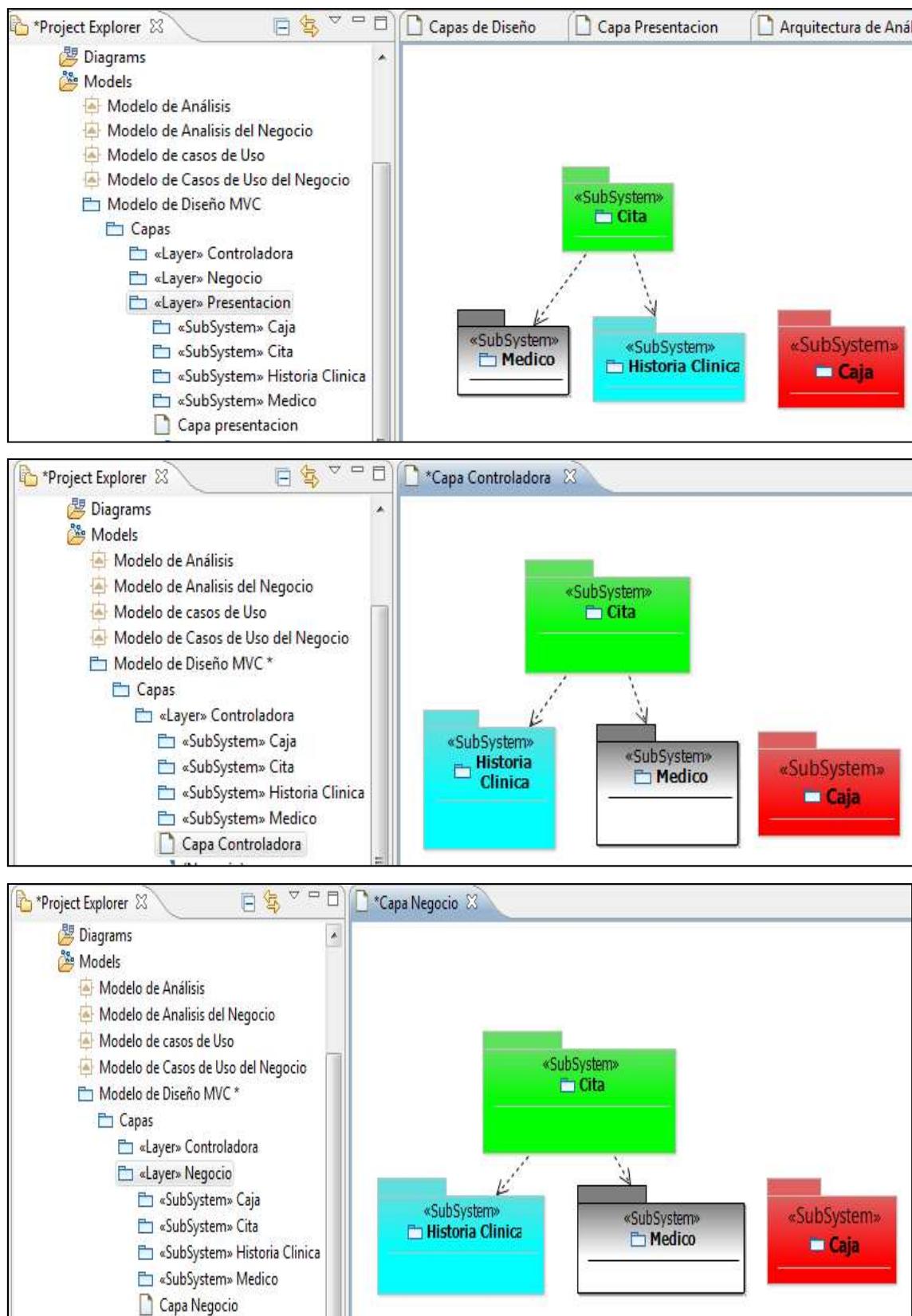
Ahora, especifique los perfiles para el modelo de diseño así:



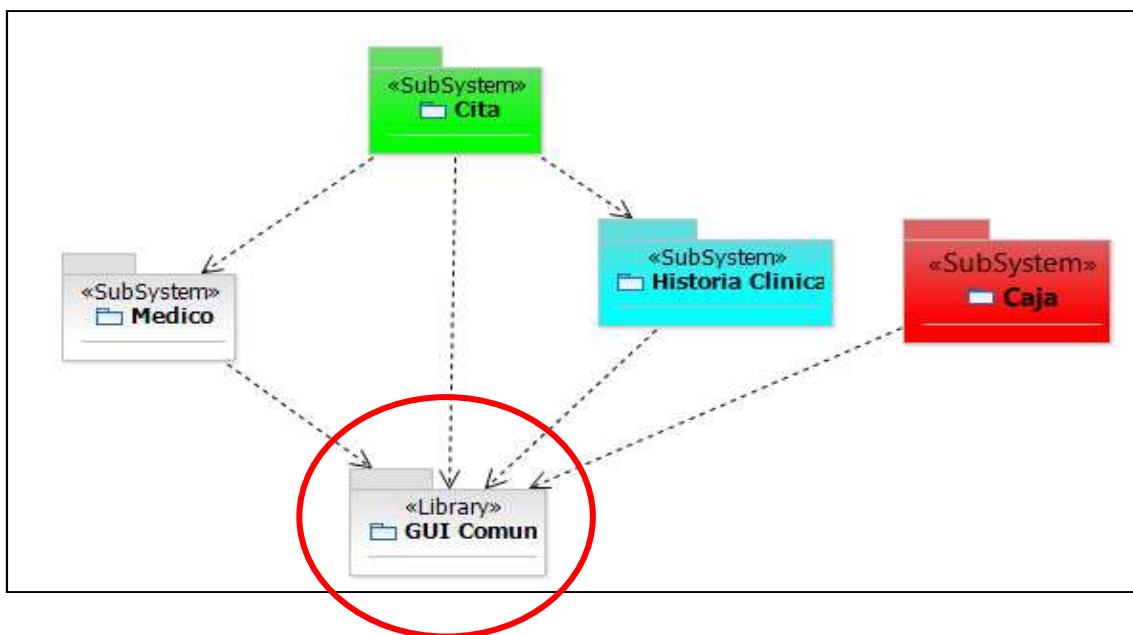
6. Ahora, para crear las clases de diseño configuromos los diagramas de clases desde **Window >> Preferences**:



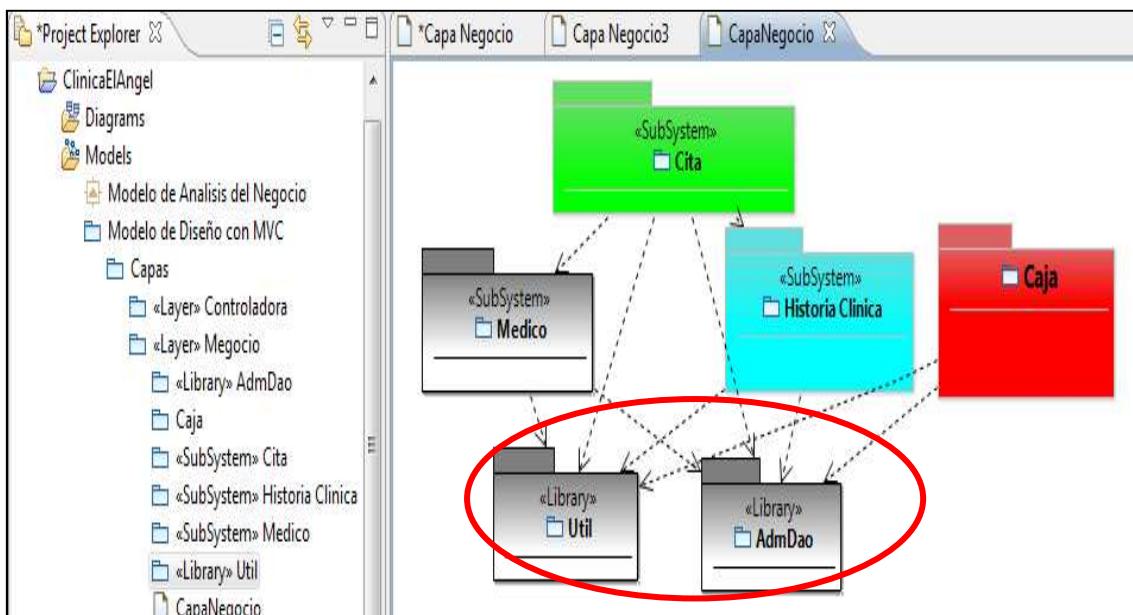
7. A continuación, en cada capa agregue los mismos paquetes que creó en la arquitectura de análisis manteniendo los mismos colores. Luego, cambie sus estereotipos a **Subsystem**. En la figura se muestra los subsistemas que contiene los casos de uso que hemos trabajado hasta el momento.



8. Ahora agregue otros paquetes que agruparán clases de uso común o clases utilitarias y, luego, asígneles el estereotipo correspondiente, según sea el caso: **Common** o **Library**. En este caso, solo se agregarán otros paquetes en las capas: Presentación y Negocio.

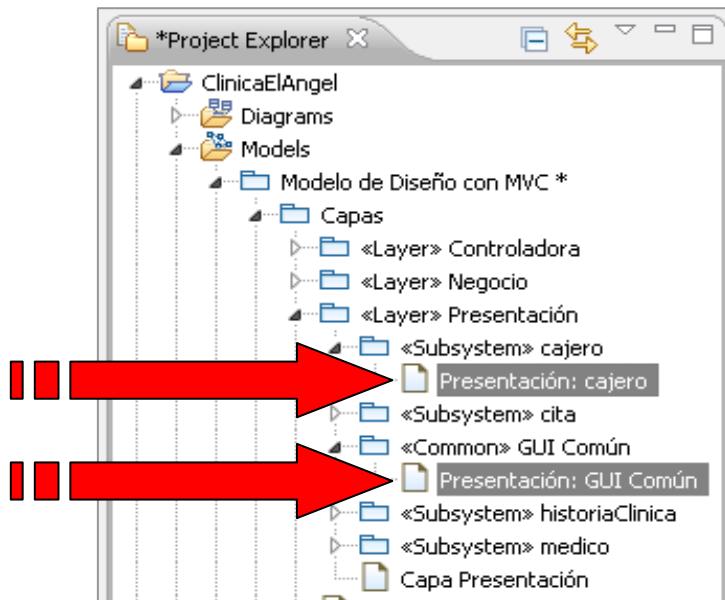


En la capa de negocio, agregar los paquetes AdmDAO y Util y asignarles el estereotipo **library**. El cambio debe mostrarse así:

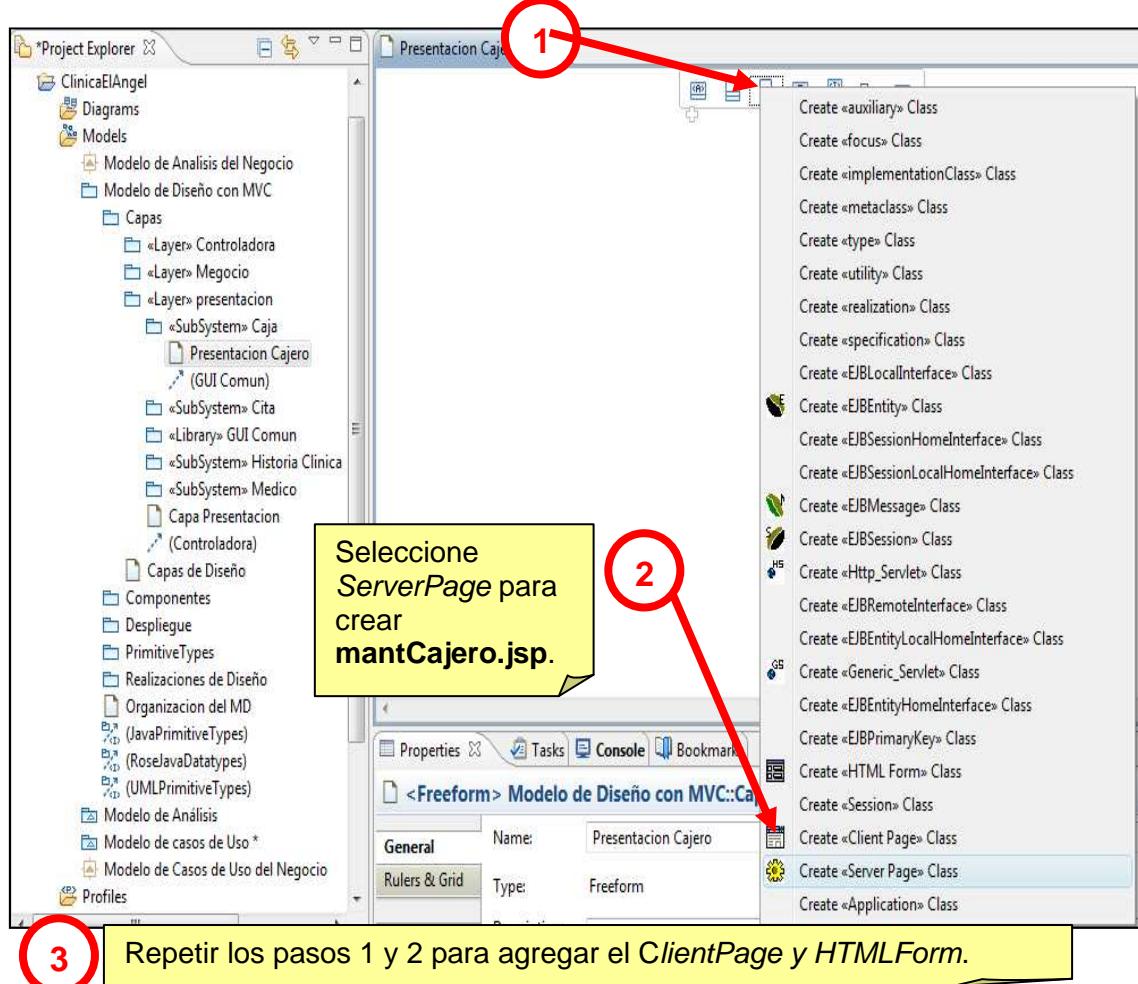


9. A continuación, empezaremos a trabajar con la **Capa Presentación** para realizar el CU Mantener Cajero.

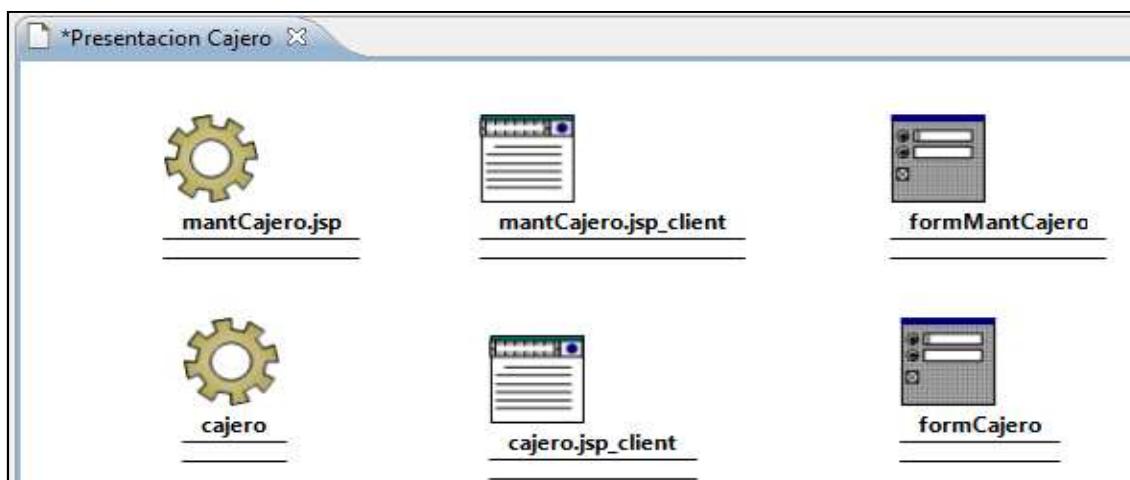
9.1. Cambie el nombre de los diagramas *Main* de los paquetes:



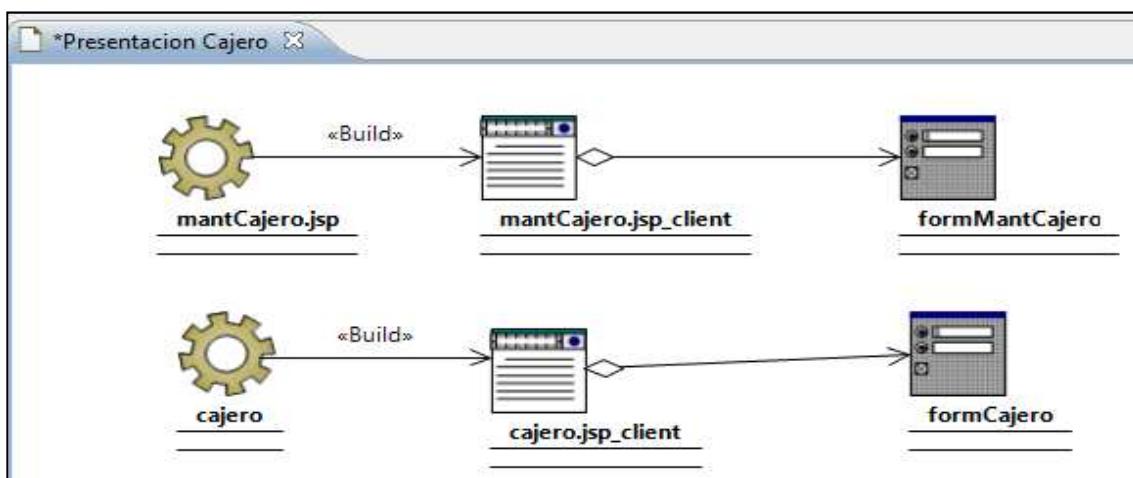
9.2. Luego, crearemos los JSP del CU Mantener Cajero los cuales son **mantCajero.jsp** y **cajero.jsp**. Por cada JSP que tenga formularios, se crea tres clases estereotipadas, tal como se muestra a continuación:



Debe quedar así:

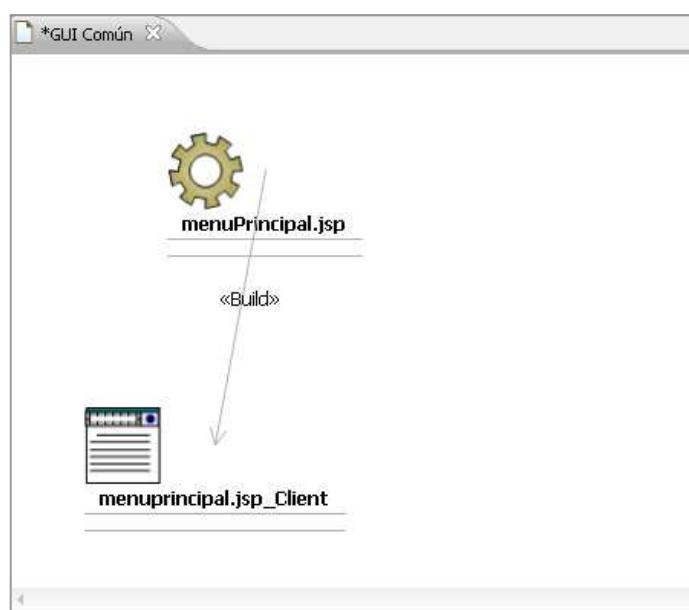


- 9.3. Ahora agregue las relaciones entre las clases. Para la asociación entre <<Server Page>> y <<Client Page>> indique el estereotipo **Build** tal como se hizo con las capas y para <<Client Page>> a <<HTML Form>> agregue una agregación por composición directa.

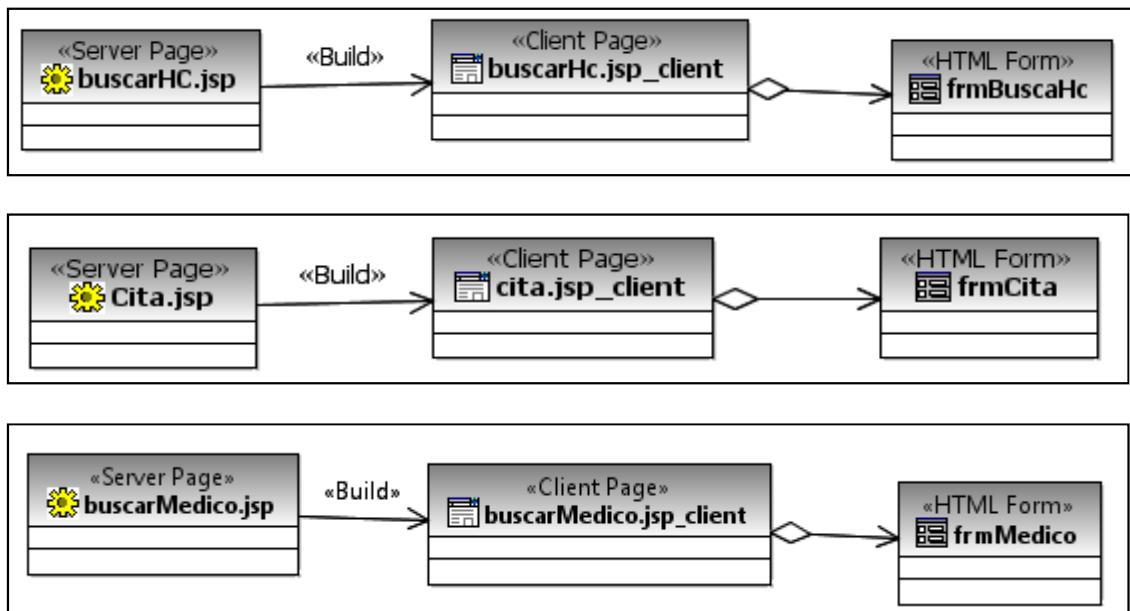


- 9.4. Ahora cree **menuPrincipal.jsp** en GUI Común

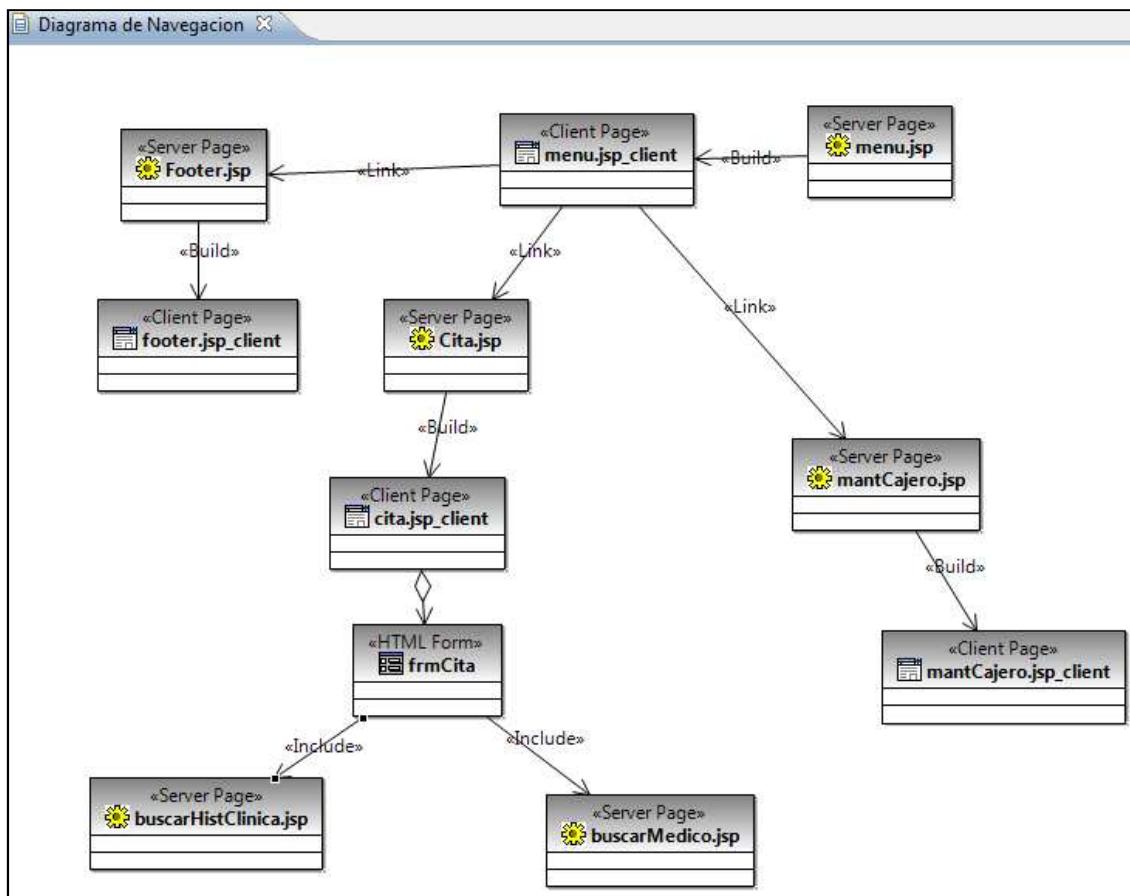
NOTA: Como esta página no contiene formularios no se le creará un *HTML Form*.



9.5. Lo mismo se trabajará para los paquetes Historia Clínica, Cita y médico.

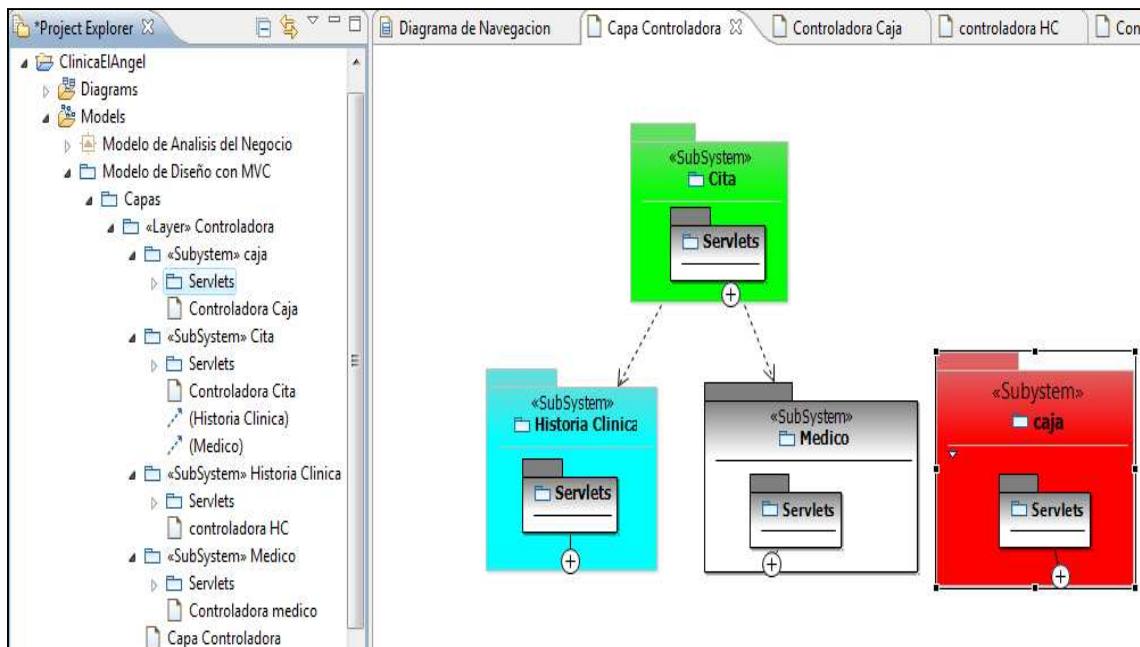


9.6. Ahora, en la capa de Presentación, se agrega un diagrama de clases el cual llamamos Diagrama de Navegación y ahí se coloca la navegación de la capa de presentación

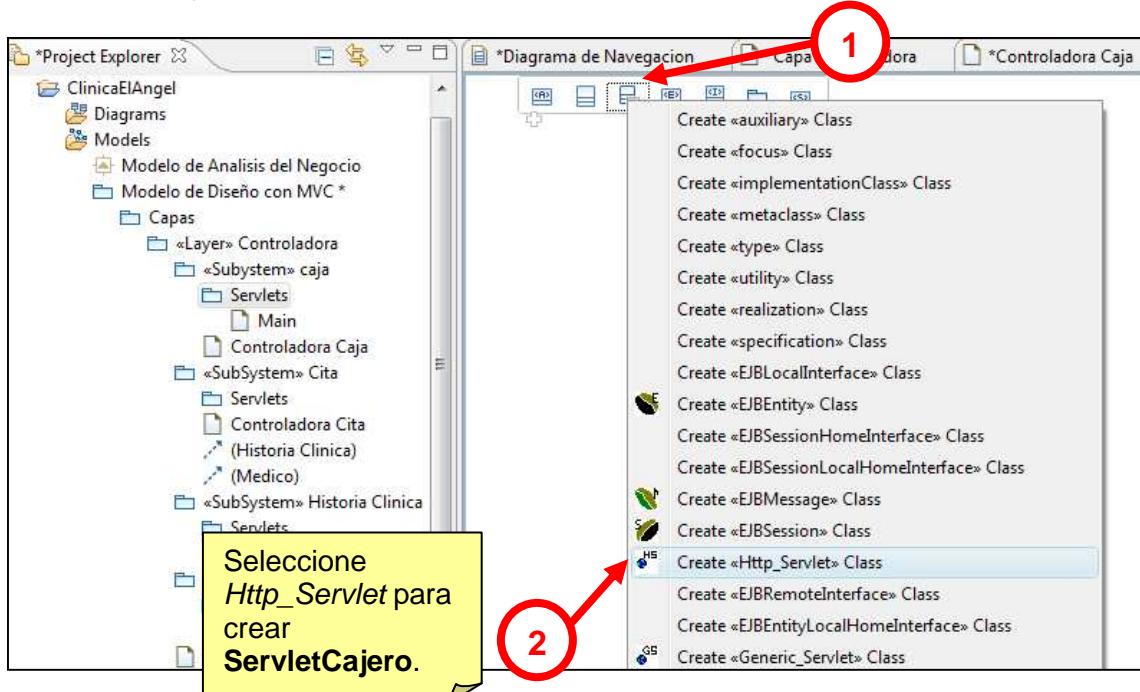


10. Luego, agregue los servlets en la **Capa Controladora** para realizar los CU Mantener Cajero. CU buscar historia Clínica y Generar cita.

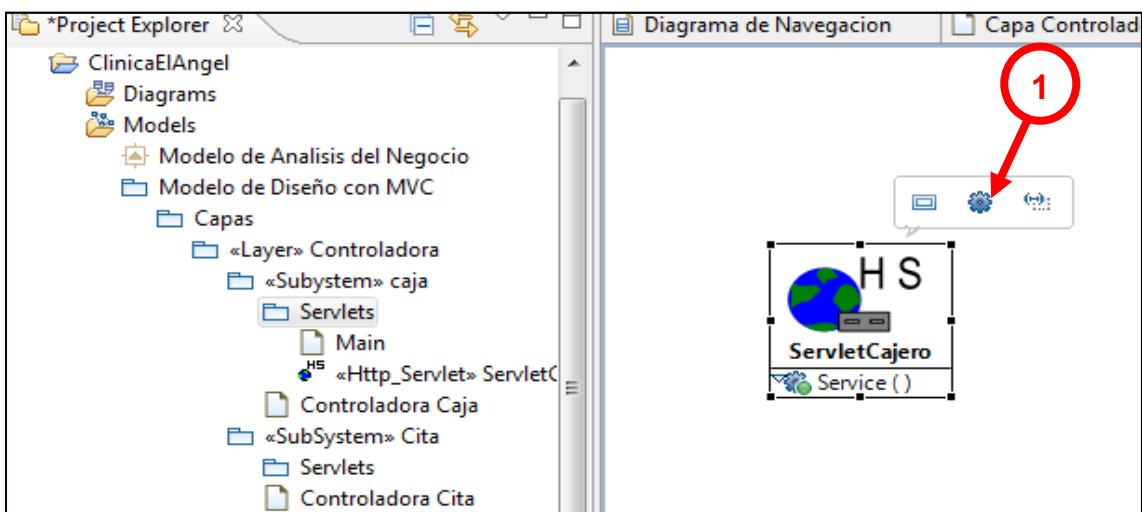
10.1. Primero, cambie los nombre del diagrama *Main* de cada paquete en el cual trabajará y agregue el paquete **servlets**:



10.2. Ahora crearemos el servlet para este subsistema a partir de una clase estereotipada, tal como se muestra a continuación:

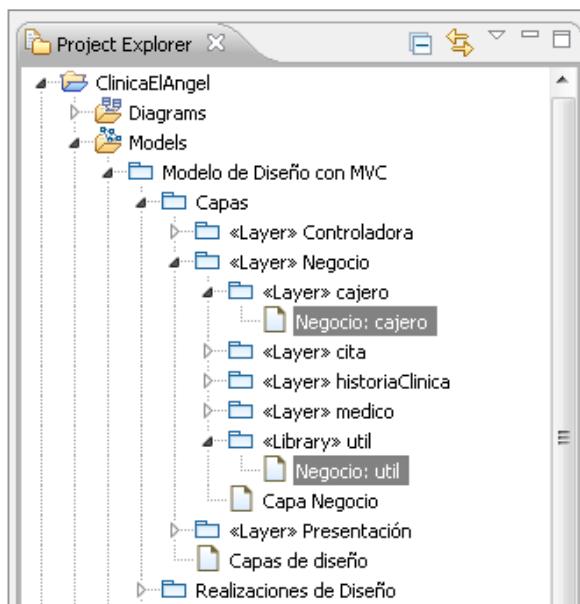


10.3. Por último, agregue la operación **Service()**.

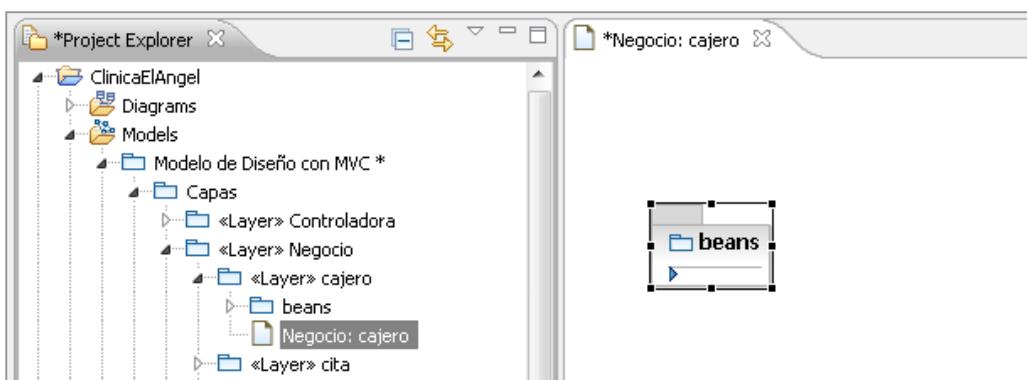


11. Luego, agregue las clases de diseño en la **Capa Negocio** para realizar el CU Mantener Cajero.

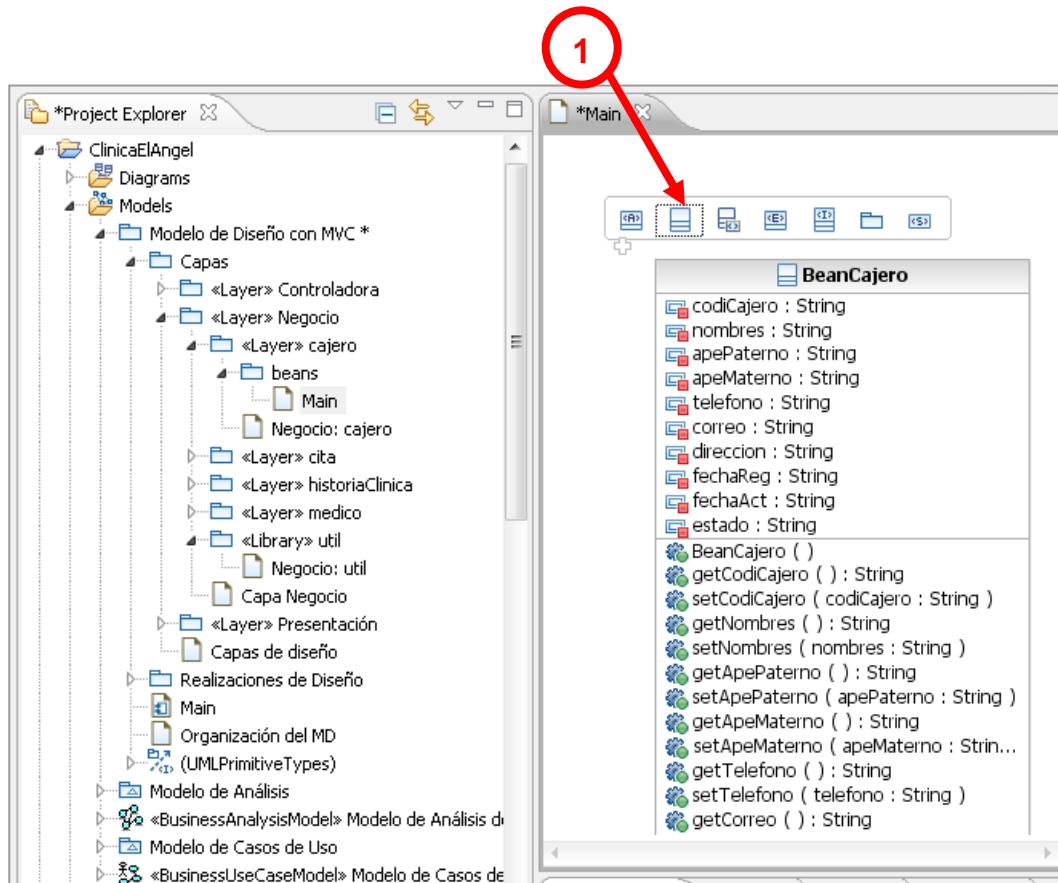
11.1. Primero, cambie el nombre del diagrama *Main* de los paquetes en los cuales trabajará:



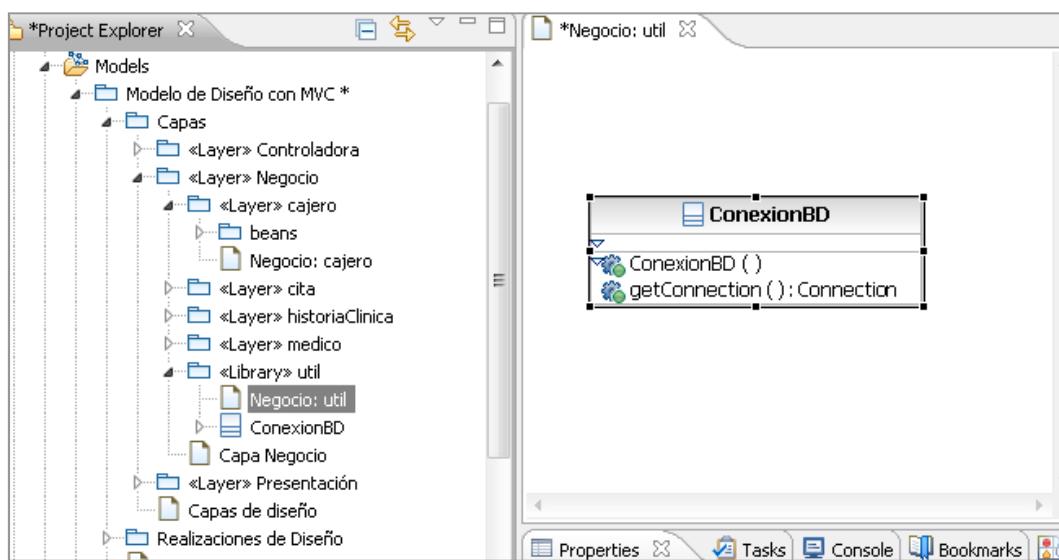
11.2. En el subsistema cajero agregue el paquete **beans**



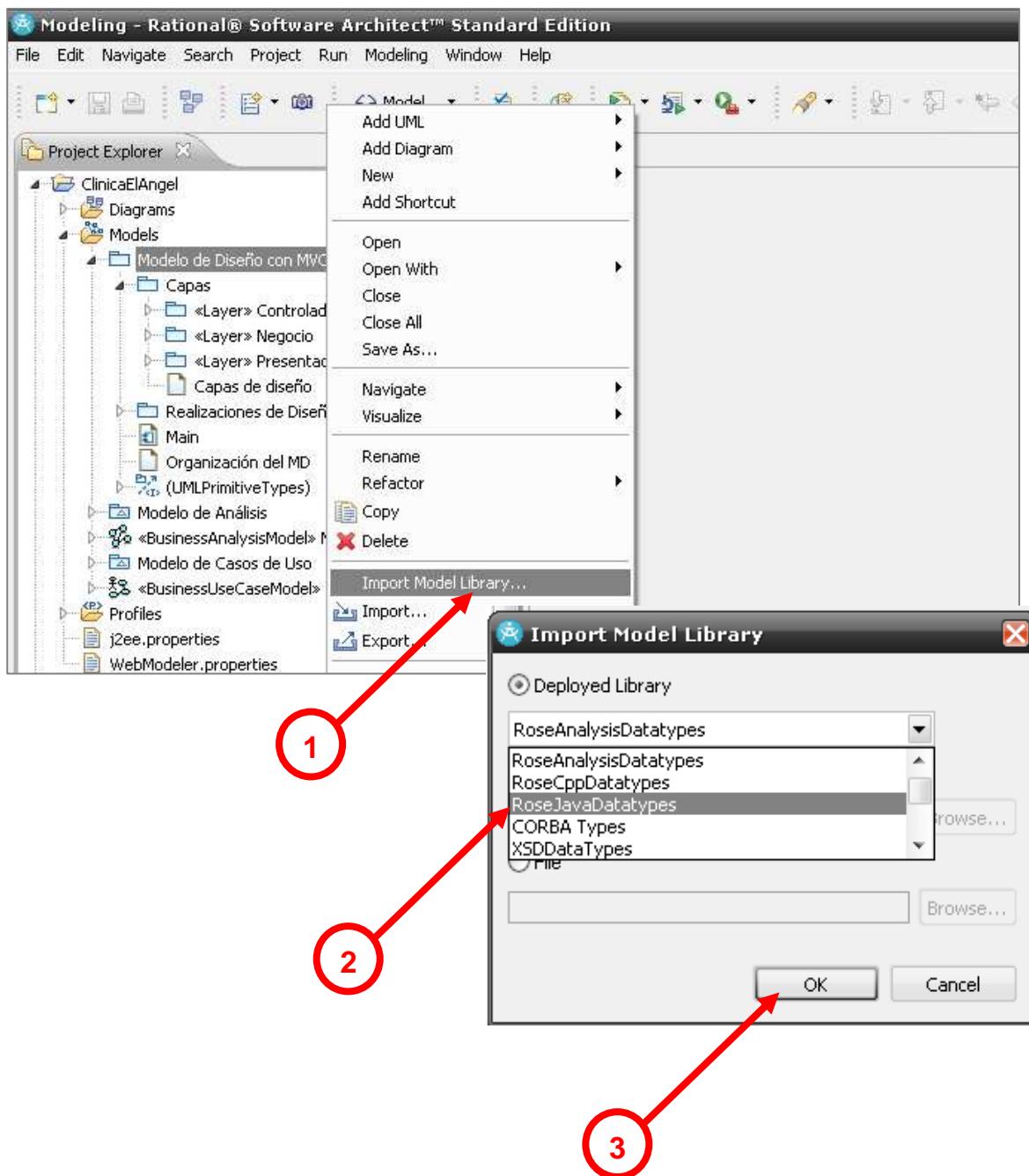
- 11.3. A continuación, en el paquete beans, agregue el **BeanCajero** con sus atributos y operaciones de acceso (get/set).



- 11.4. Por último, en el paquete util agregue la clase ConectaBD con sus operaciones.



12. Finalmente, agregue las librerías de JAVA.

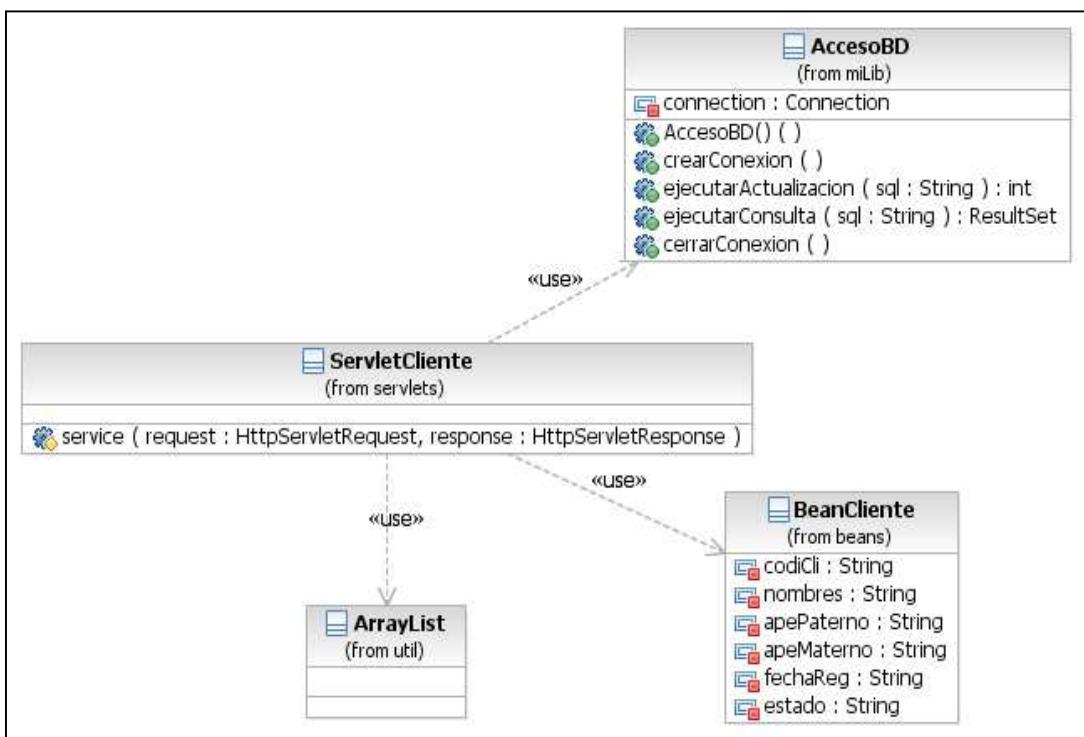


## 2. DISEÑO DE CASOS DE USO CON MVC

A continuación, se explica la definición de algunos elementos que se utilizarán en las realizaciones de diseño de un caso de uso.

### 2.1. Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.



**Figura 1. Diagrama de clases de diseño.**

En las siguientes tablas se muestran las relaciones que pueden existir entre clases. La descripción de cada una, permitirá entender la estructura de clases diseñada para una funcionalidad que será implementada en JAVA:

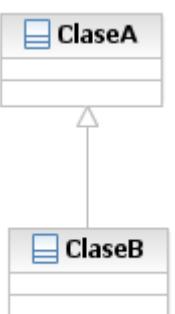
Tipo de relación	UML	Java
Herencia		<pre>public class ClaseA {     //Más código }  public class ClaseB extends ClaseA {     //Más código }</pre>
Implementación		<pre>public interface InterfazX {     //Más código }  public class ClaseY implements InterfazX {     //Más código }</pre>

Tabla 1. Relación de herencia e implementación.

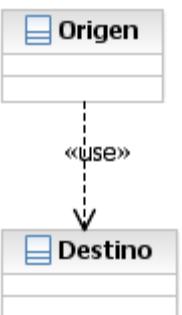
Tipo de dependencia	UML	Descripción
<<use>> (De uso)		El funcionamiento del origen depende del funcionamiento del destino.

Tabla 2. Relaciones de dependencia.

Tipo de dependencia	UML	Descripción
<<instance>> (De instancia)	<pre> classDiagram     class Origen     class Destino     Origen "2" --&gt; "1" Destino : &lt;&lt;instance&gt;&gt;   </pre>	El origen solo crea instancias del destino.

Tabla 2. Relaciones de dependencia. (Continuación)

## 2.2. Diagrama de secuencia

El diagrama de secuencia describe la dinámica del sistema, describiendo las interacciones entre un grupo de objetos mostrando de forma secuencial los envíos de mensajes entre objetos. El diagrama puede asimismo mostrar los flujos de datos intercambiados durante el envío de mensajes.

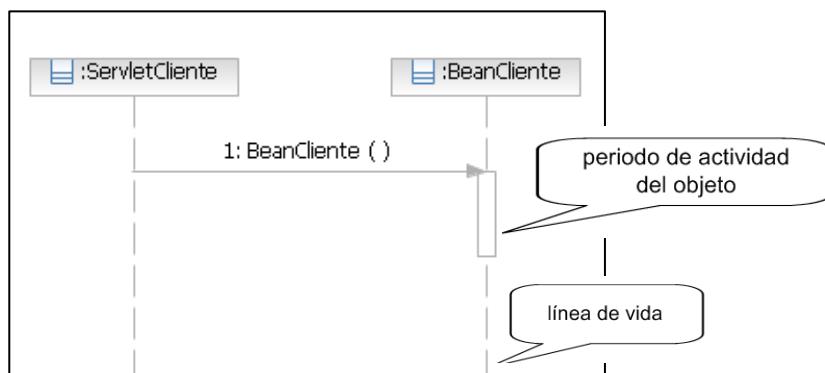


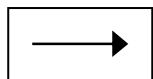
Figura 2. Diagrama de Secuencia con mensaje sincrónico.

## 2.3. Línea de vida de un objeto

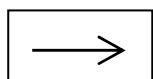
Dado que representa la dinámica del sistema, el diagrama de secuencia hace entrar en acción las instancias de clases que intervienen en la realización de la subfunción a la que está vinculado. A cada instancia se asocia una línea de vida que muestra las acciones y reacciones de la misma, así como los períodos durante los cuales ésta está activa, es decir, durante los que ejecuta uno de sus métodos.

## 2.4. Mensajes

Para interactuar entre sí, los objetos se envían mensajes. Durante la recepción de un mensaje, los objetos se vuelven activos y ejecutan el método del mismo nombre. Un envío de mensaje es, por tanto, una llamada a un método y se representan mediante flechas horizontales que unen la línea de vida del objeto emisor con la línea de vida del objeto destinatario. Existen diferentes tipos de mensajes:



El **mensaje sincrónico** es el utilizado con mayor frecuencia. Su uso significa que el expedidor del mensaje espera que la activación del método mencionado por el destinatario finalice antes de continuar su actividad.



En los **mensajes asincrónicos**, el expedidor no espera el término de la activación invocada por el destinatario. Esto se produce al modelar sistemas en los que los objetos pueden funcionar en paralelo (es el caso de los sistemas multi-thread, donde los tratamientos se efectúan en paralelo).

## 2.5. E) Fragmentos combinados

Para un diagrama de secuencia que representa procedimientos complejos hay un número de mecanismos que permiten agregar un grado de lógicas de procedimientos a los diagramas y que a la vez vienen bajo el encabezado de fragmentos combinados. Un fragmento combinado es una o más secuencias de procesos incluidas en un marco y ejecutadas bajo circunstancias nombradas específicas. Los fragmentos disponibles son los siguientes:

1. El fragmento **Alternative** (denotado “alt”) modela estructuras if...else.
2. El fragmento **Option** (denotado “opt”) modela estructuras switch.
3. El fragmento **Break** modela una secuencia alternativa de eventos que se procesa en lugar de todo del resto del diagrama.
4. El fragmento **Parallel** (denotado “par”) modela procesos concurrentes.
5. El fragmento de secuenciado **Weak** (denotado “seq”) incluye un número de secuencias para las cuales todos los mensajes se deben procesar en un segmento anterior, antes de que el siguiente segmento pueda comenzar, pero que no impone ningún secuenciado en los mensajes que no comparten una línea de vida.
6. El fragmento de secuenciado **Strict** (denotado “strict”) incluye una serie de mensajes que se deben procesar en el orden proporcionado.
7. El fragmento **Negative** (denotado “neg”) incluye una serie de mensajes inválidos.
8. El fragmento **Critical** incluye una sección crítica.
9. El fragmento **Ignore** declara un mensaje o mensajes que no son de ningún interés si este aparece en el contexto actual.
10. El fragmento **Consider** es el opuesto del fragmento **Ignore**: cualquier mensaje que no se incluya en el fragmento Consider se debería ignorar.
11. El fragmento **Assertion** (denotado “assert”) designa que cualquier secuencia que no se muestra como un operando de la aserción es inválida.
12. El fragmento **Loop** incluye una serie de mensajes que están repetidos.

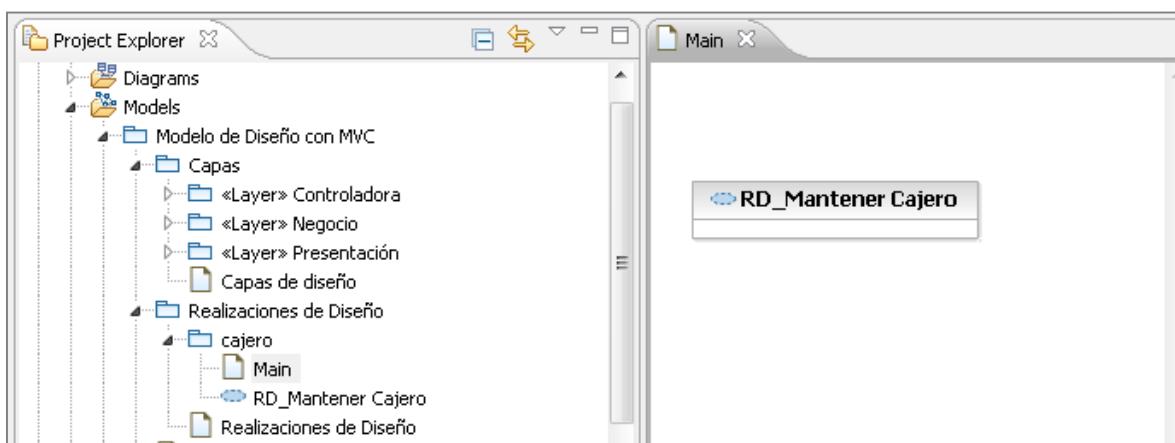
En la siguiente figura se muestra un ejemplo de los fragmentos combinados más utilizados:

<b>Fragmento OPT: Estructura de control alternativa 'if' que no tiene secuencia 'else'.</b>	
Condición	Resultado
if ( saldoInicial == 0 ) System.out.println("Sin fondos");	opt [ saldoInicial==0 ]
<b>Fragmento ALT: Estructura de control alternativa 'if-else'.</b>	
Condición	Resultado
if ( saldoInicial == 0 ) System.out.println("Sin fondos"); else System.out.println("Con fondos");	alt [ saldoInicial==0 ] [else] -----
<b>Fragmento LOOP: Estructuras de repetición o bucles 'for' y 'while'.</b>	
Condición	Resultado
for( int i = 0; i < 10; i++ ) { System.out.println("contador:"+i); }	loop [ int i = 0; i < 10; i++ ]
while( i < 10 ) { System.out.println("contador:"+i); i++; }	loop [ i < 10 ]

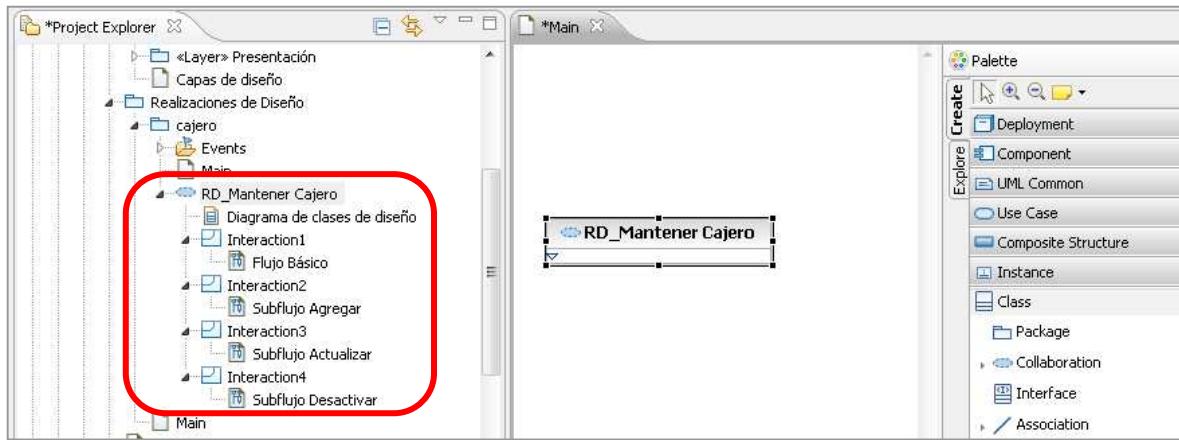
**Figura 3. Fragmentos combinados más utilizados.**

## Realizaciones de diseño de un caso de uso

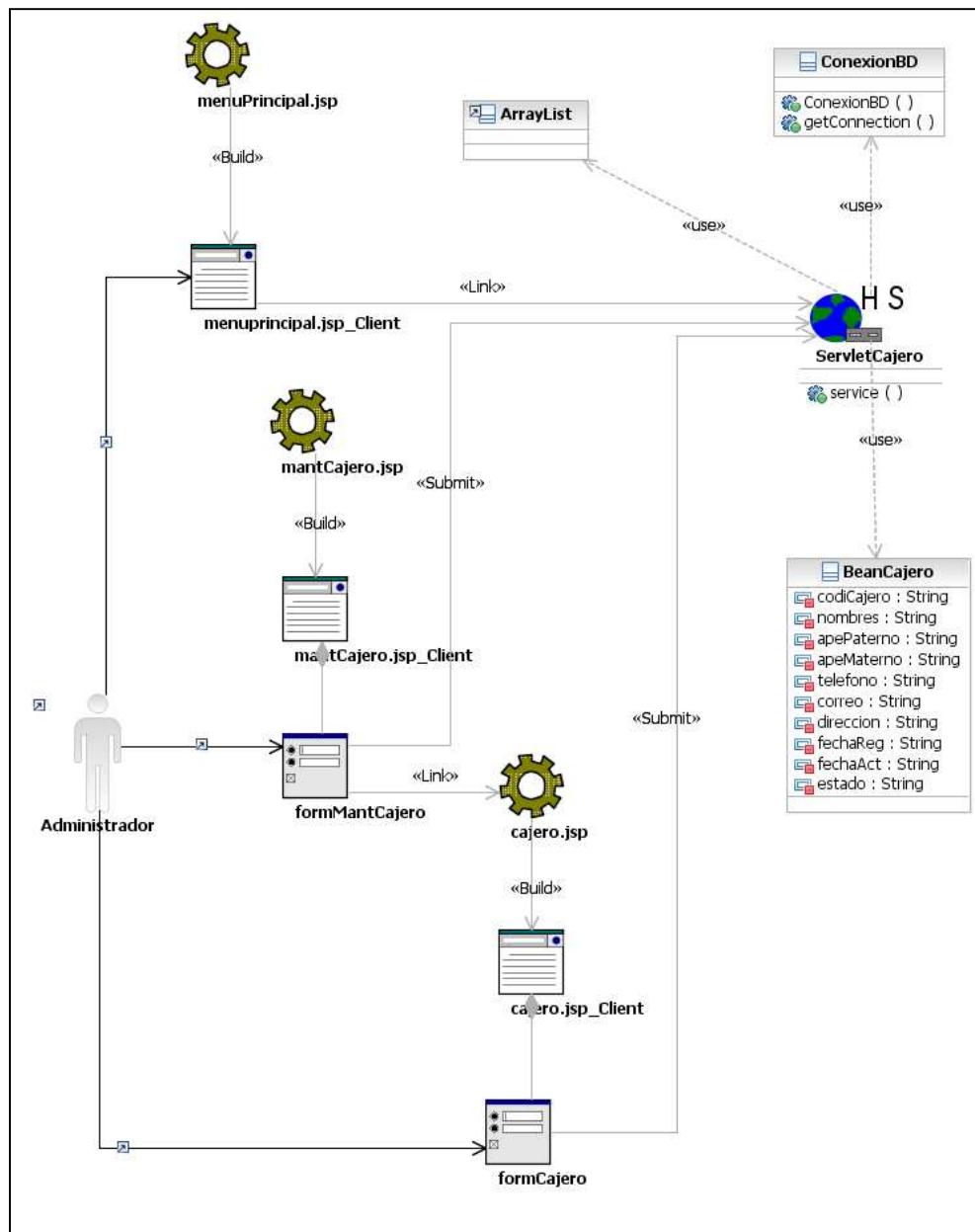
1. En el paquete Realizaciones de Diseño cree el paquete **cajero** con una colaboración para elaborar la Realización de Diseño del CU Mantener Cajero.



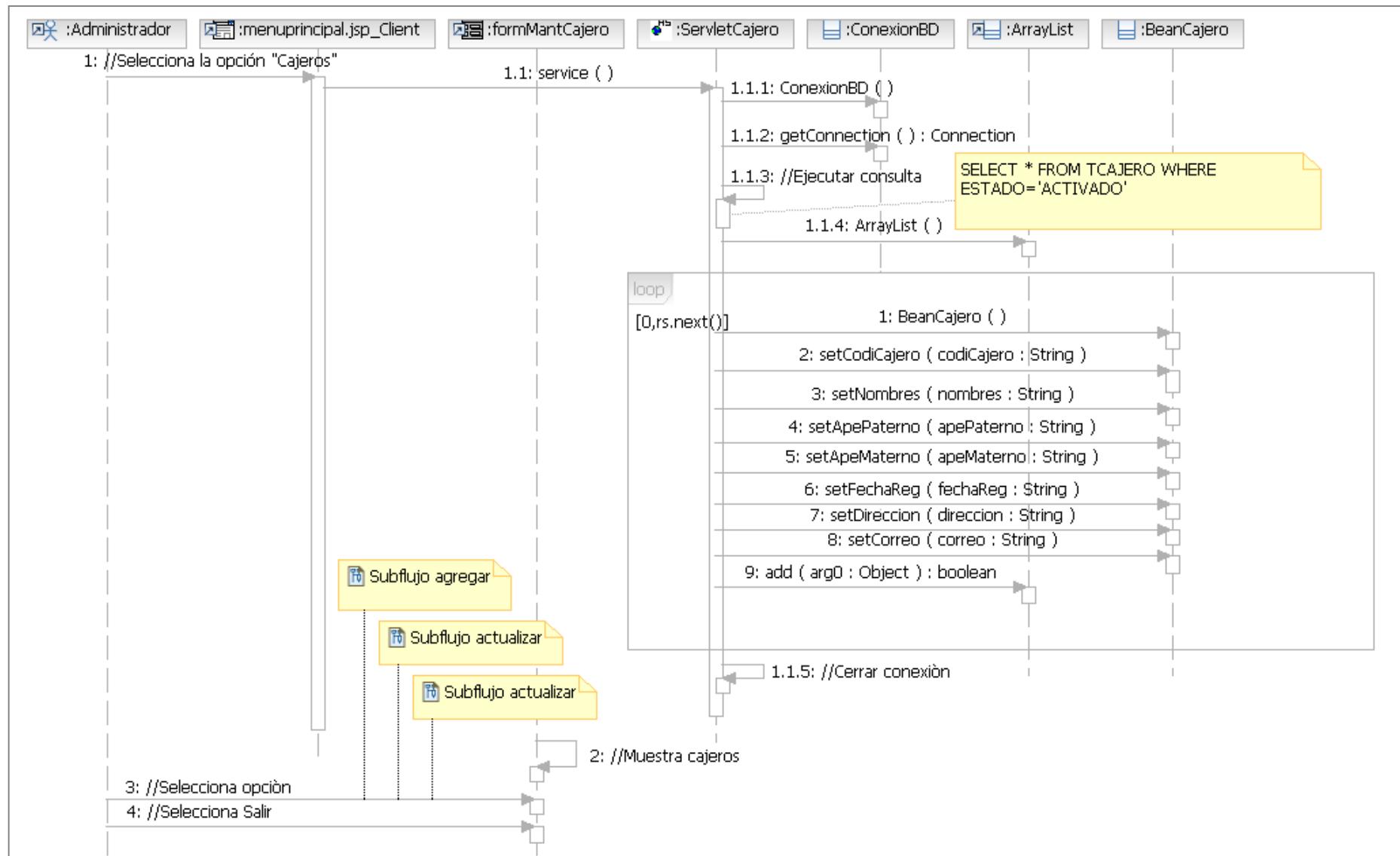
2. Luego, agregue un diagrama de clases y diagramas de secuencia para el flujo básico y subflujos.



3. Ahora realice el Diagrama de Clases de Diseño. Para ello: arrastre los elementos de diseño de las diferentes capas y agregue sus relaciones. Debe quedar así:



4. A continuación, se presenta el Diagramas de Secuencia del flujo básico.



## ACTIVIDADES PROPUESTAS

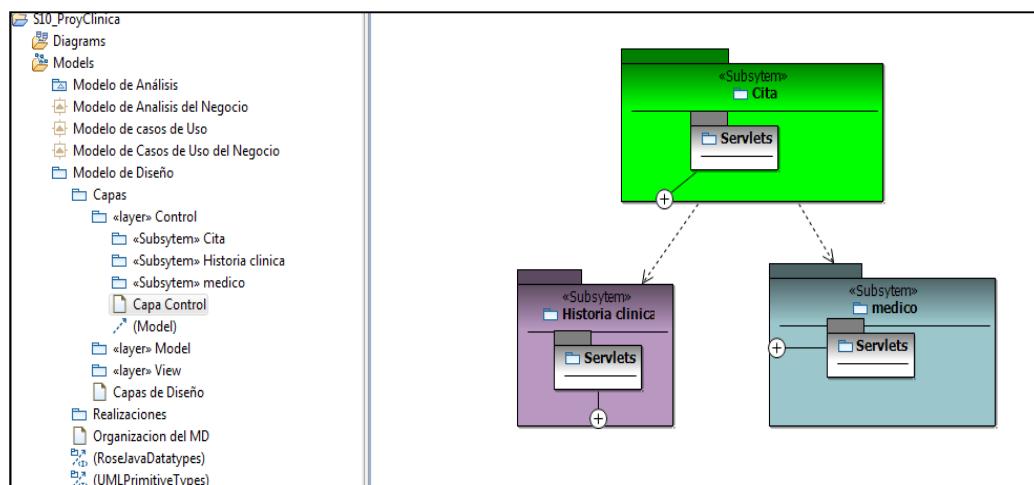
Elabore el diagrama de secuencia para los subflujos agregar, actualizar y desactivar cajeros del caso de uso Mantener Cajeros.

### 3. DISEÑO DE CASOS DE USO CON MVC

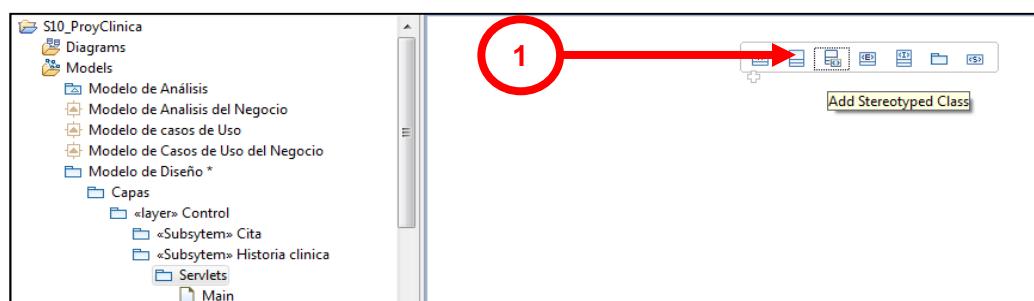
A continuación, se indica los pasos para las realizaciones de diseño de un caso de uso aplicando patrón de diseño DAO. Para el ejemplo se utilizará la ECU del caso buscar Historia clínica y Generar Cita.

#### PASO 1 Crear la clase control de la capa control del Subsystem Historia Clinica

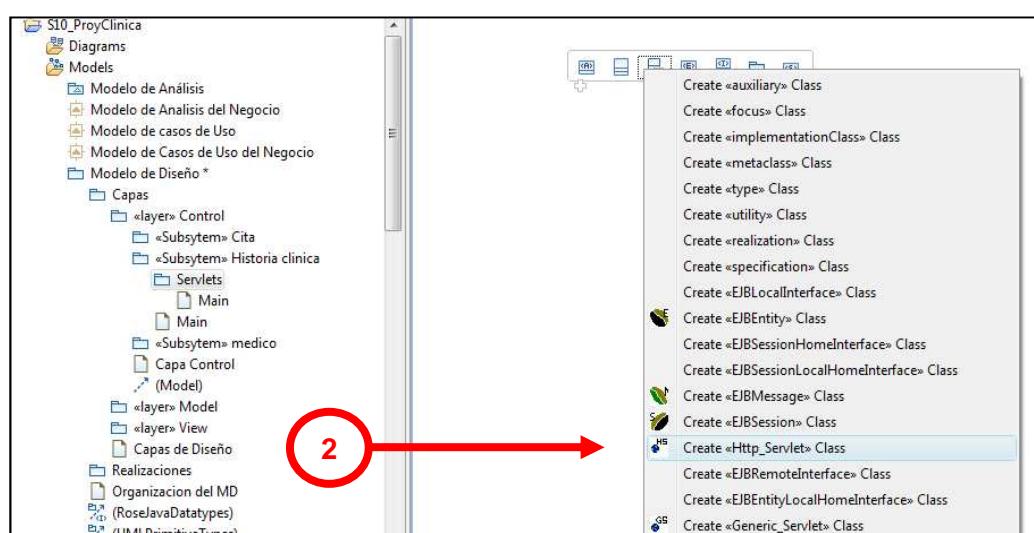
Ir a la capa control a el Subsystem Historia Clinica y crear el servlet “SevletHc”



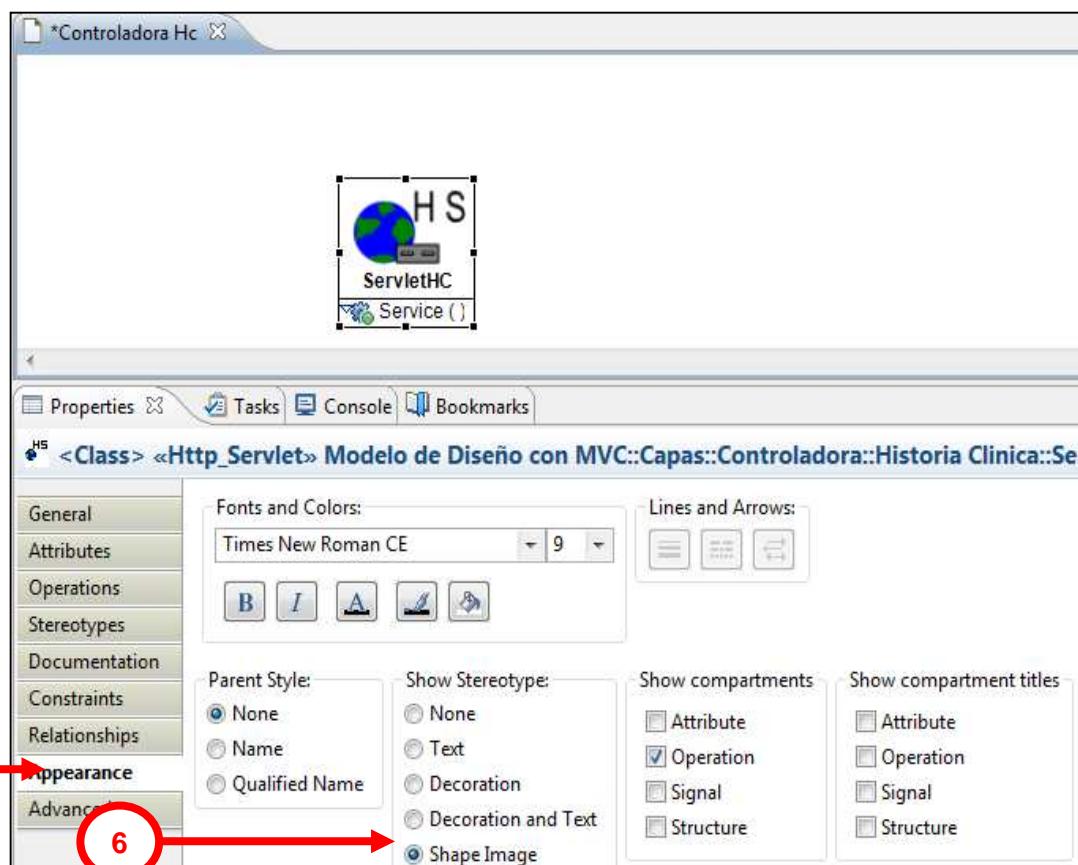
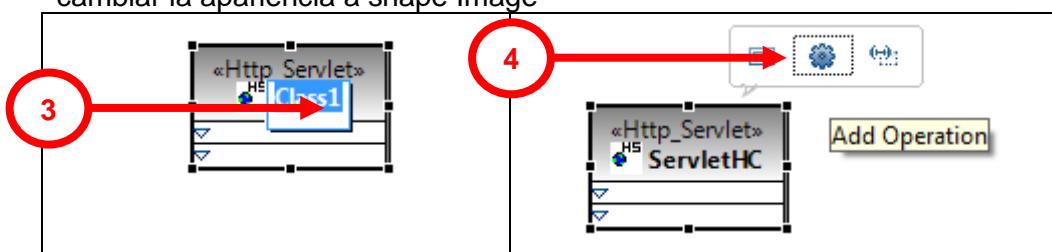
Seleccionar del menú contextual la clase estereotipada



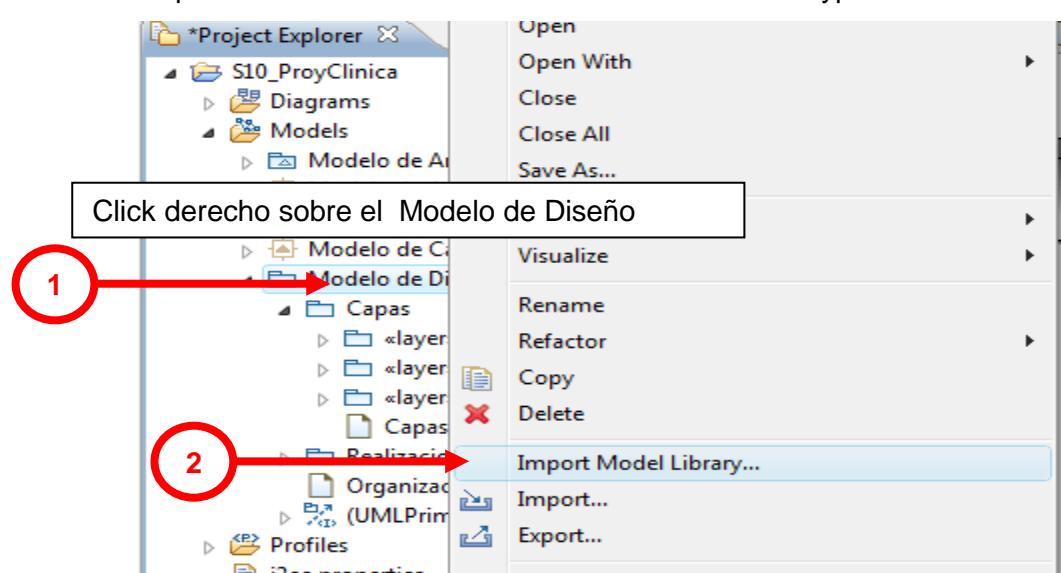
Seleccionamos la clase http Servlet



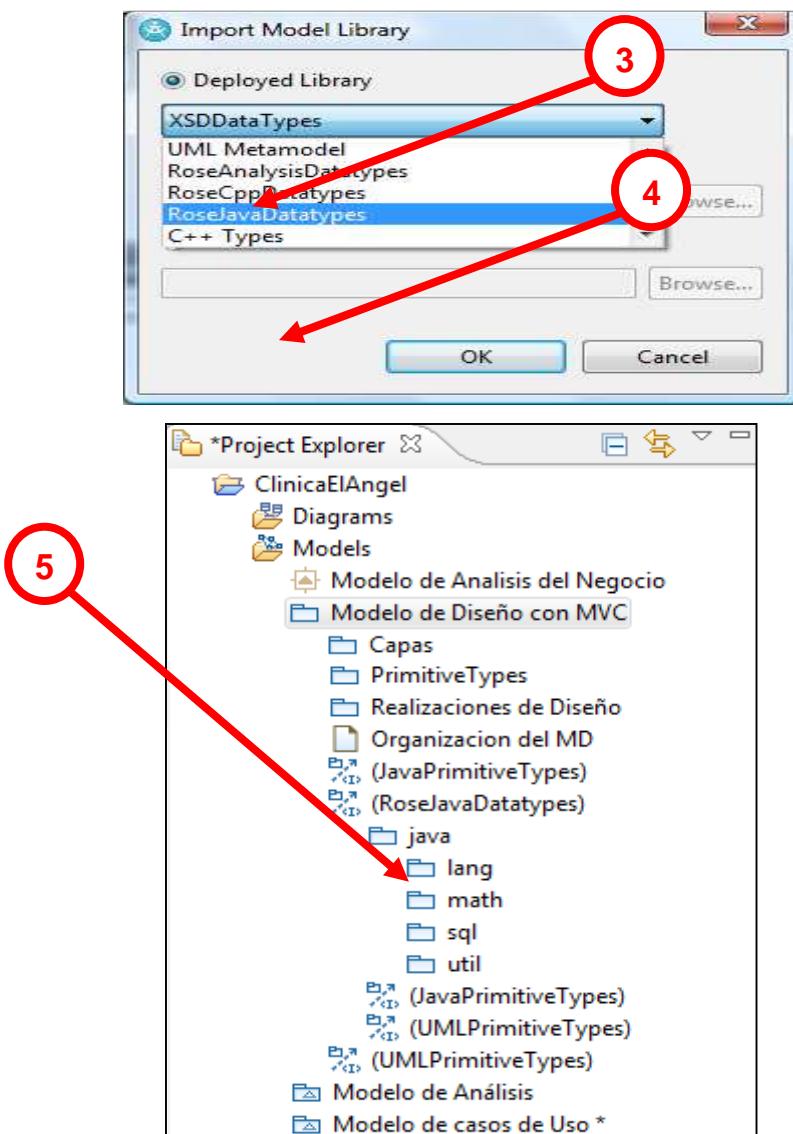
Poner como nombre a la clase ServletHC , crear la Operación Service y cambiar la apariencia a shape Image



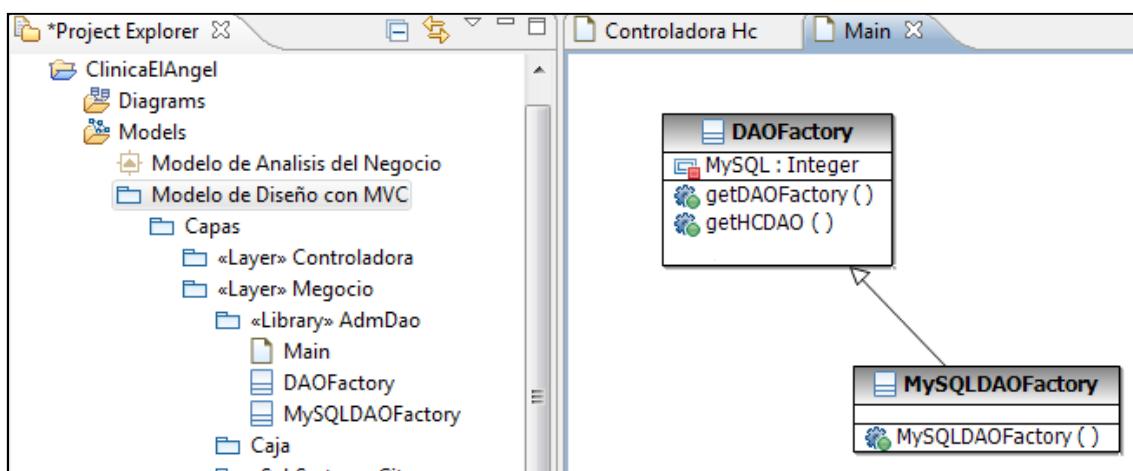
## Paso 2 Importar la librería de datos de Java RoseJavaDatatypes



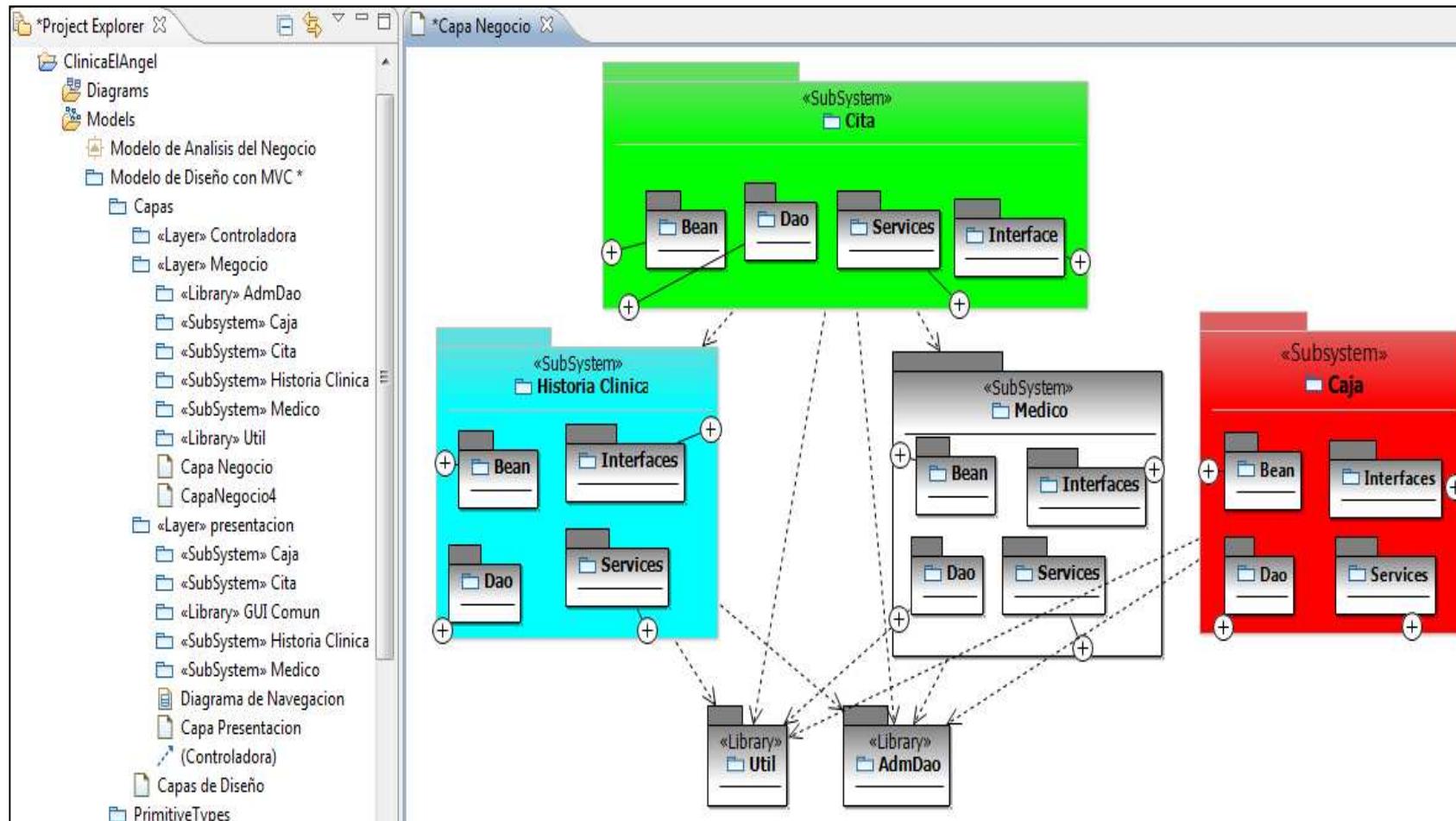
### Seleccionamos la librería RoseJavaDatatypes



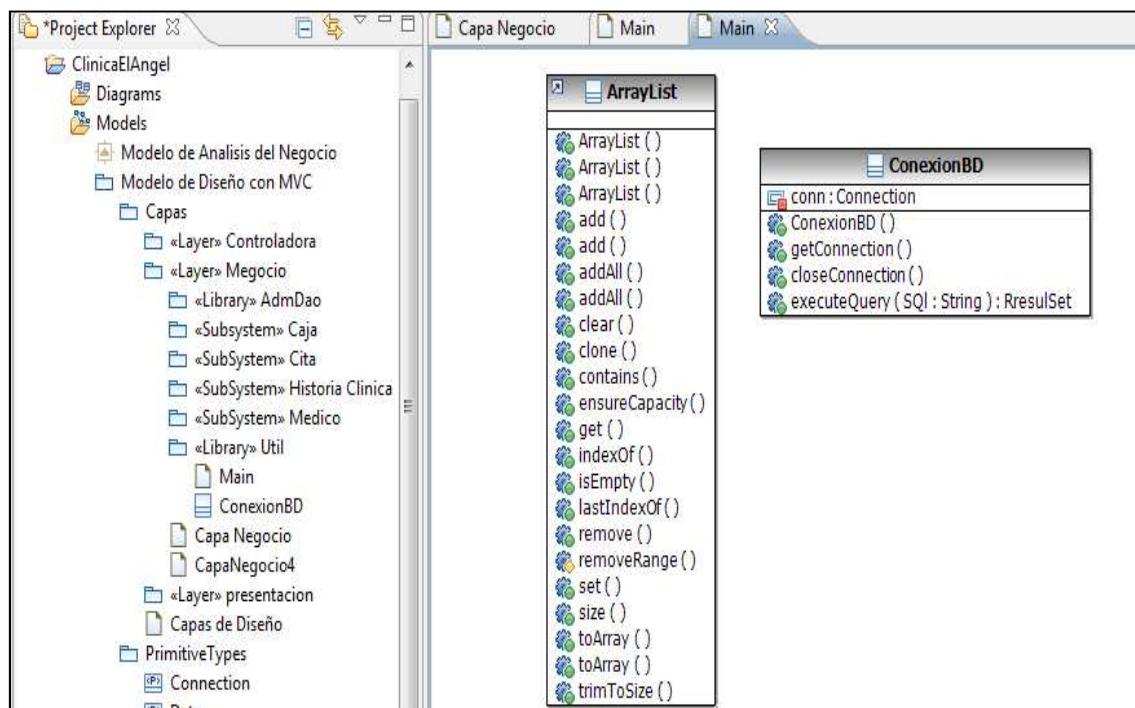
**PASO 3:** Crear las clases de la capa de Negocio, del paquete ADMDAO DAOFactory y MySQLDAOFactory en el Main del paquete AdmDAO:



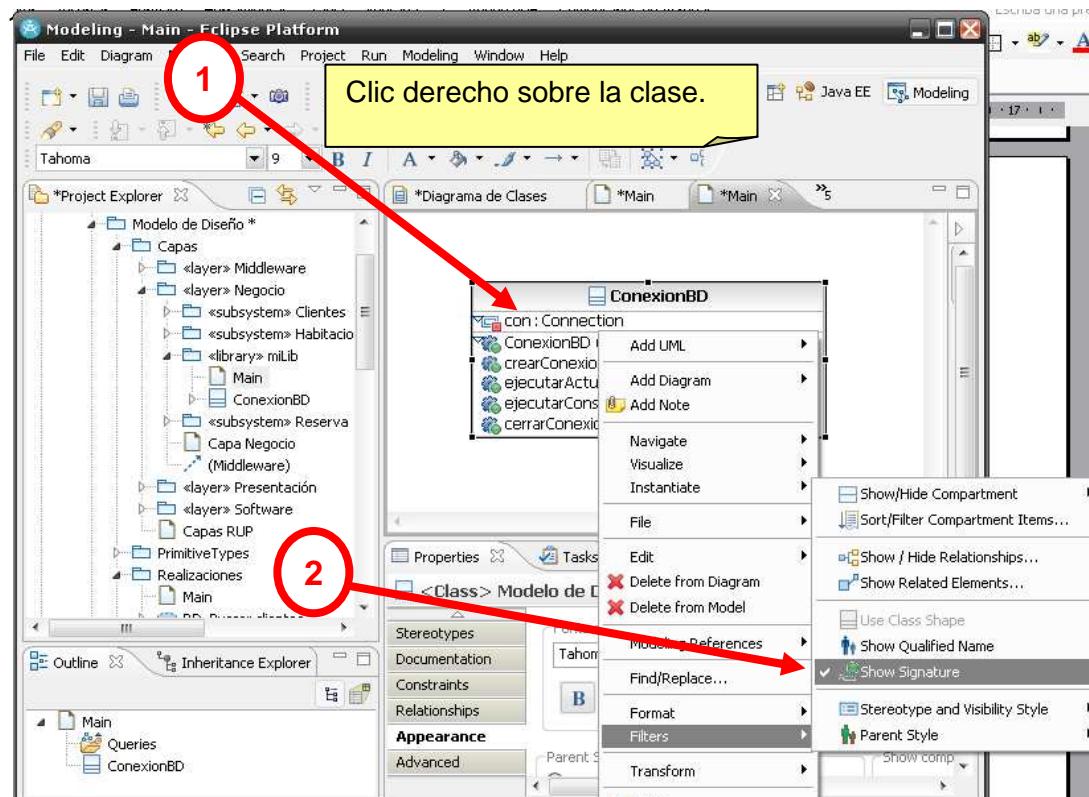
Paso 4 Crearemos en la capa de Negocio para cada SubSystem los paquetes **Bean**, Dao, Interface y Services



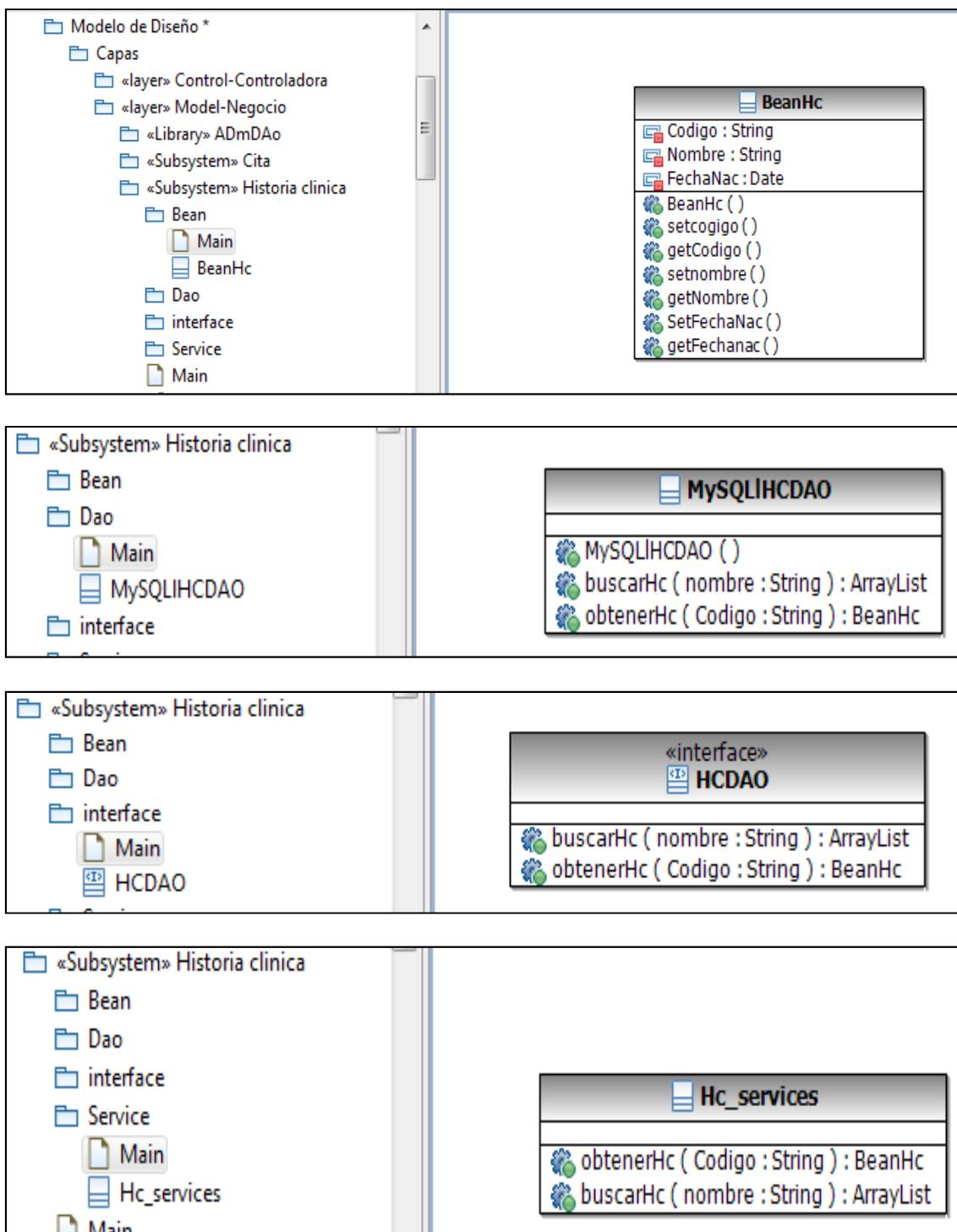
**PASO 5:** Cree la clase ConexionBD en el Main del paquete Util y arrastrar el arraylist



**NOTA:** Para mostrar la firma completa de cada operación realice lo siguiente

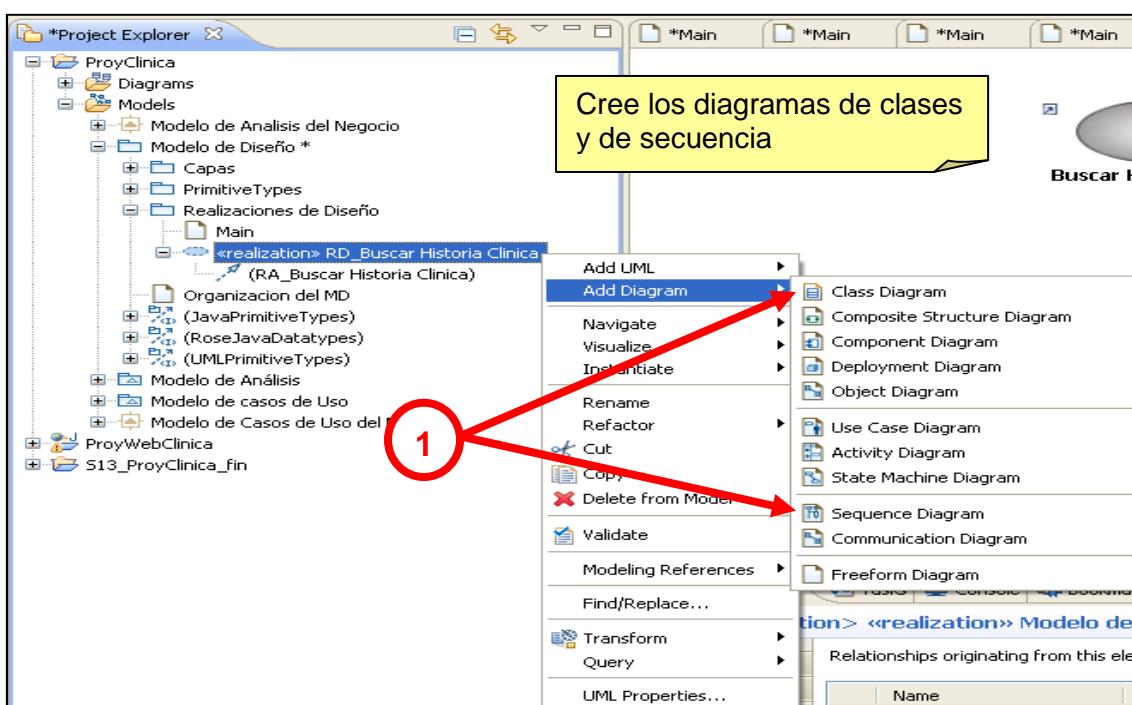
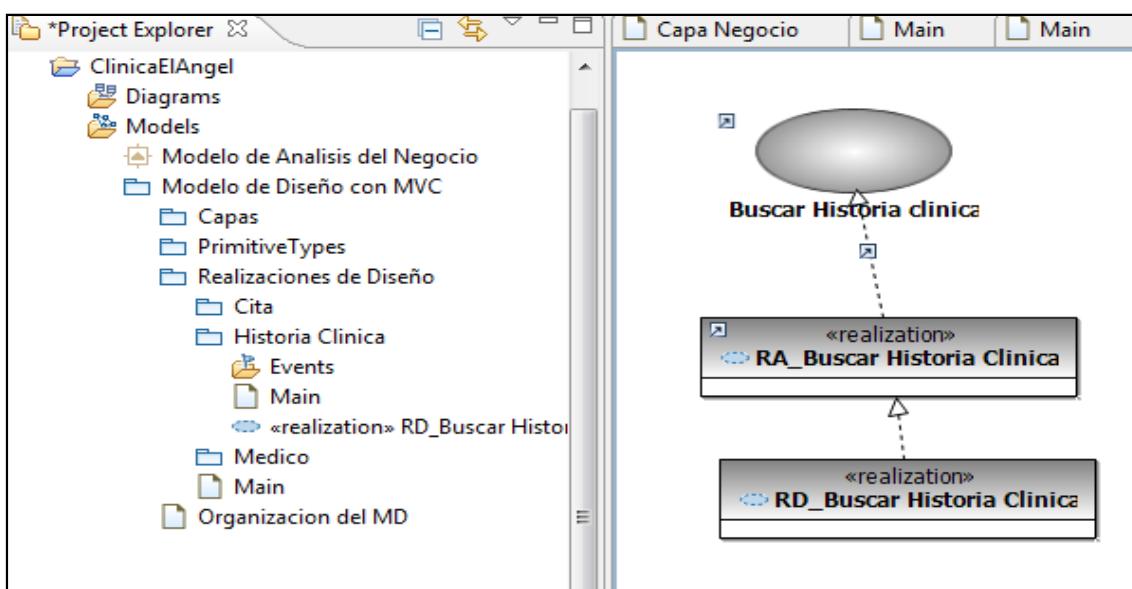


## PASO 6: Tenemos que crear el Bean, Dao, Interface y service de la clase Historia Clínica de la capa Model



Hasta este momento hemos creado los elementos de diseño que van a participar en el caso de uso “Buscar Historia Clínica”, ahora veremos las Realizaciones de diseño.

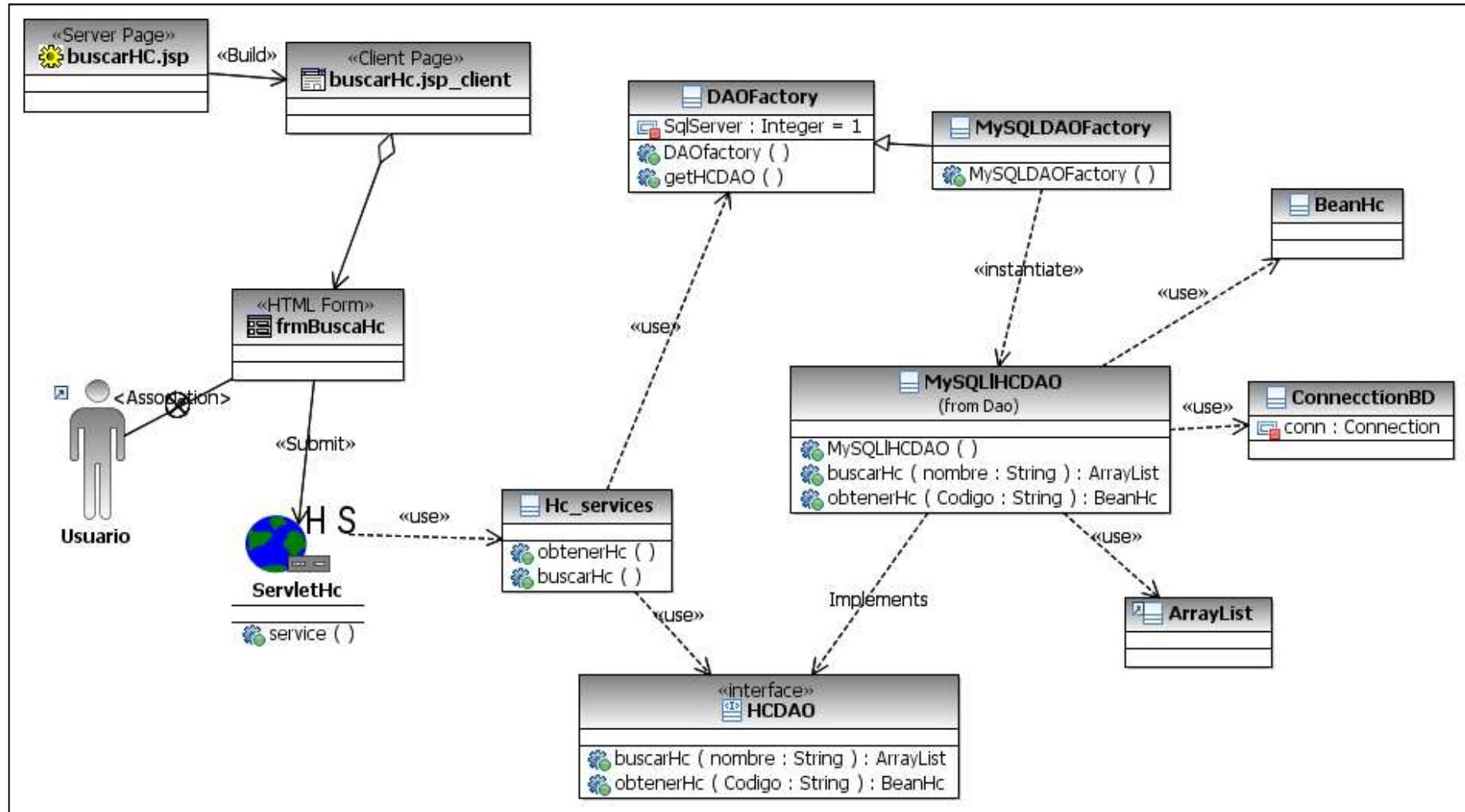
**PASO 7:** Ahora crearemos en el paquete de Realizaciones de Diseño, los diagramas de clases y diagrama de secuencias del flujo básico a partir de la realización de diseño buscar historia clínica del paquete Historia Clínica:



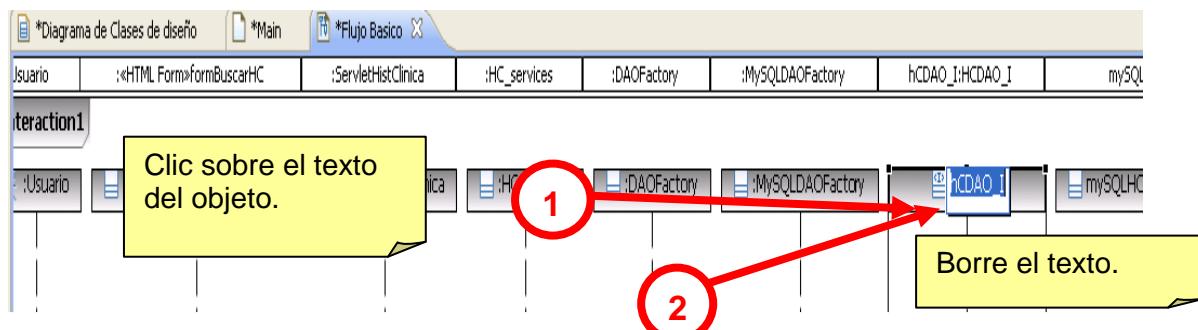
**PASO 8:** Arrastre el actor y las clases de las capas Presentación, negocio y Controladora.

- Es recomendable que las tres clases del JSP sean copiadas del Main de la capa Presentación a este Diagrama de clases.
- La clase ArrayList arrástrelo del Modelo de librería importado que se encuentra en la (RosejavaDataTypes) en / Java / Util.

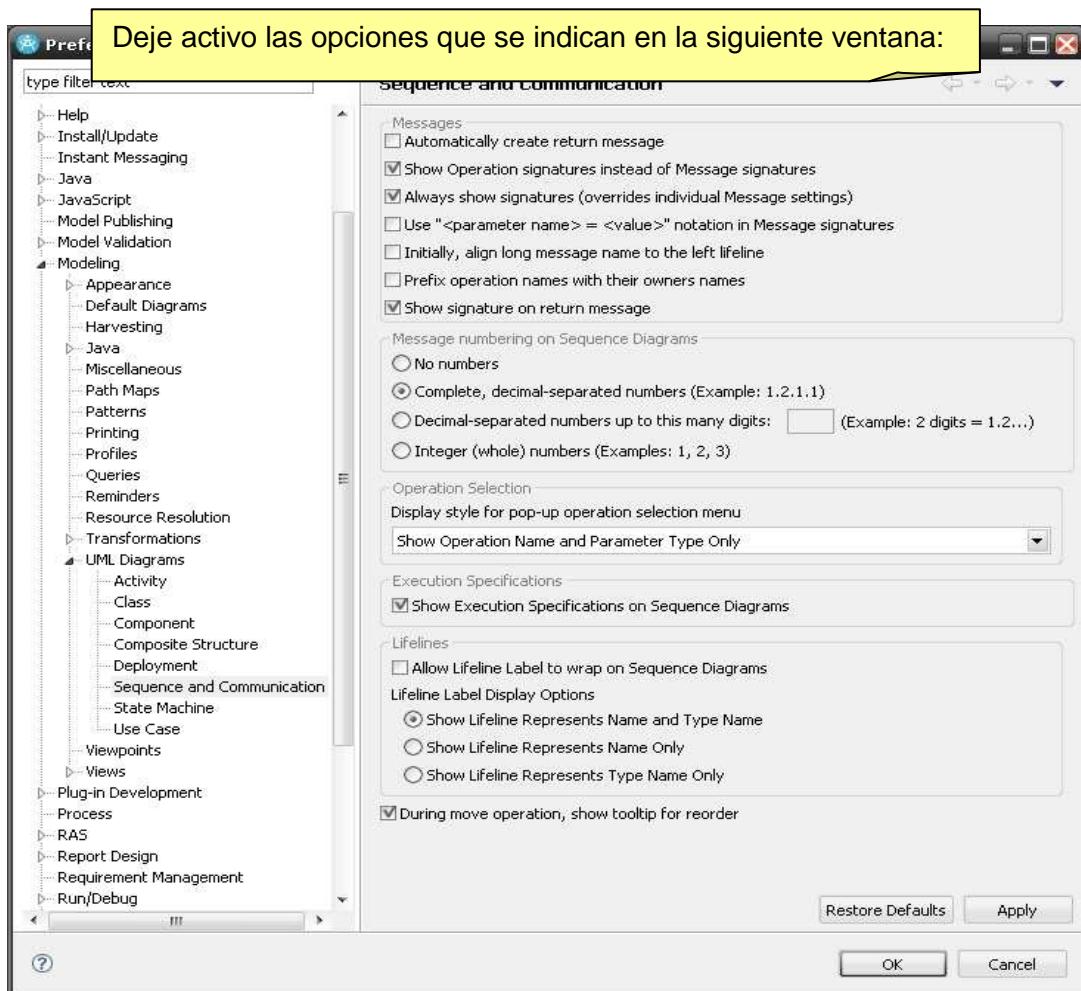
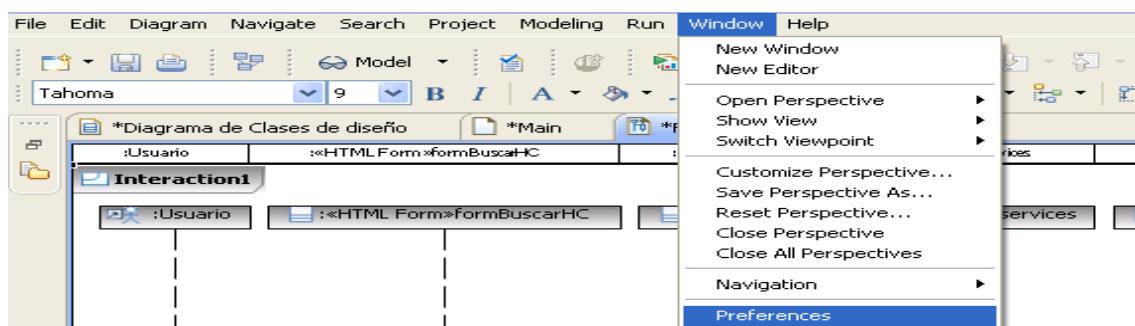
## Diagrama de clases de diseño



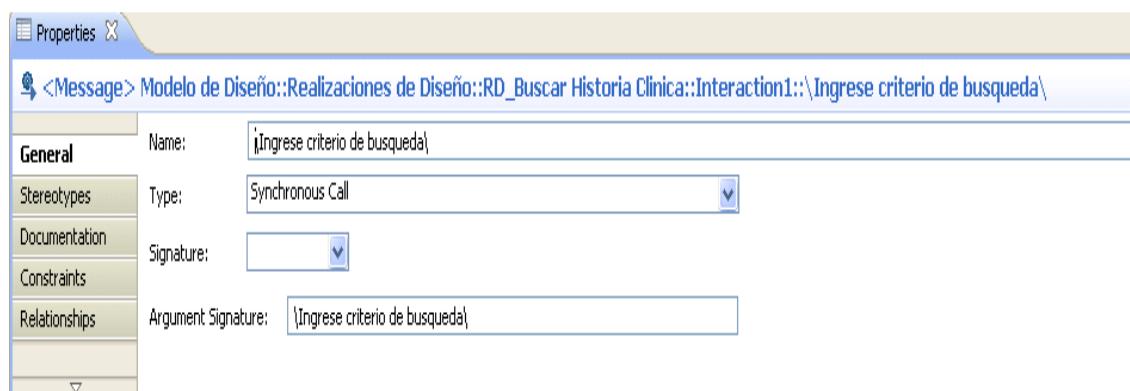
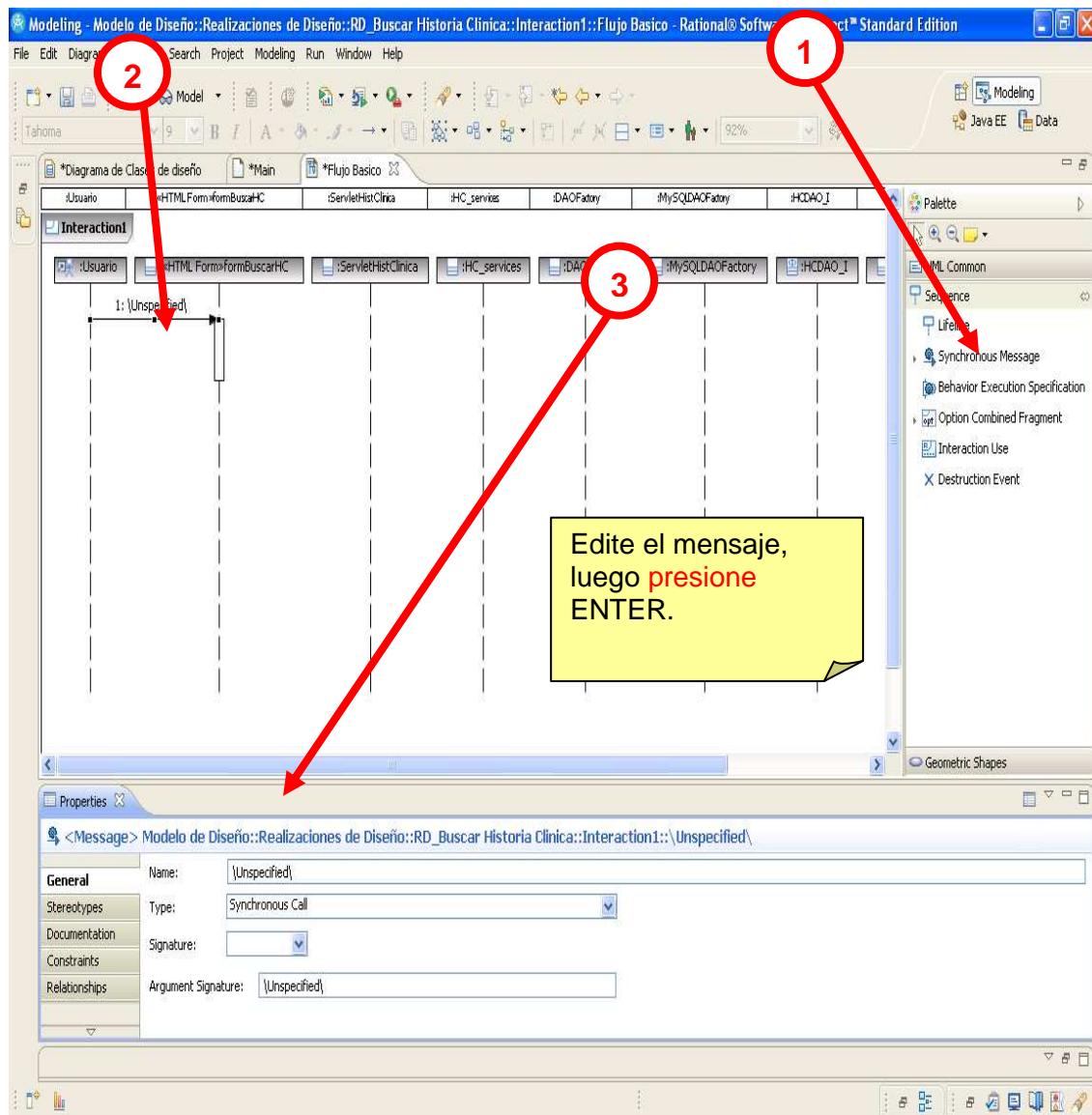
**PASO 9:** A continuación, en el diagrama de secuencia, arrastre las clases al flujo básico y, por cada elemento, dé clic sobre el texto para eliminar el nombre de instancia. Por cada método, se crea otro diagrama de secuencia



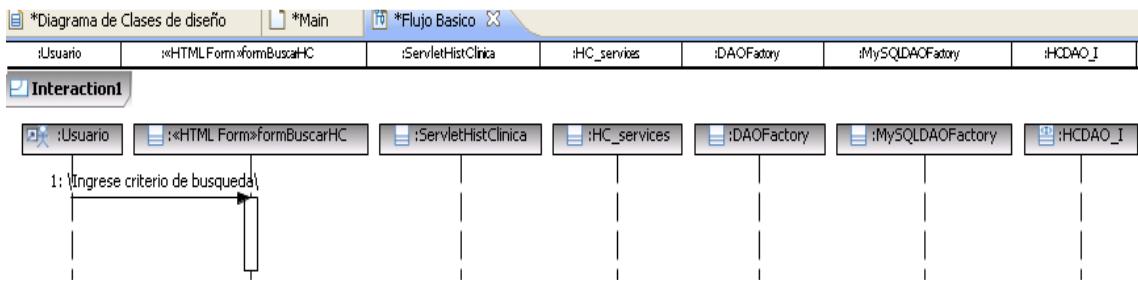
**PASO 10:** Configure el diagrama de secuencia para no mostrar mensajes de retorno:



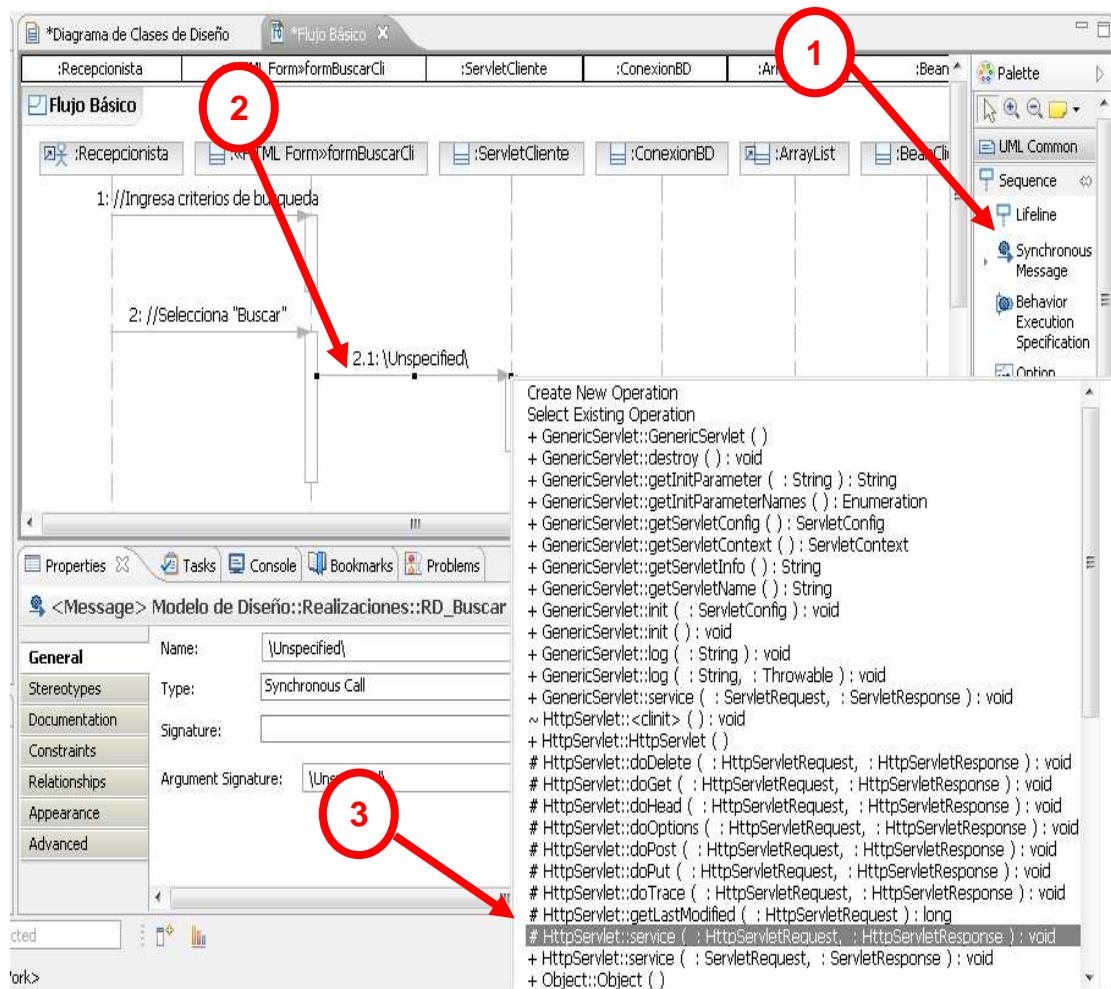
**PASO 11** Agregue los mensajes síncronos. En caso no sea un método, edite el mensaje utilizando la caja de **Propiedades/General**:



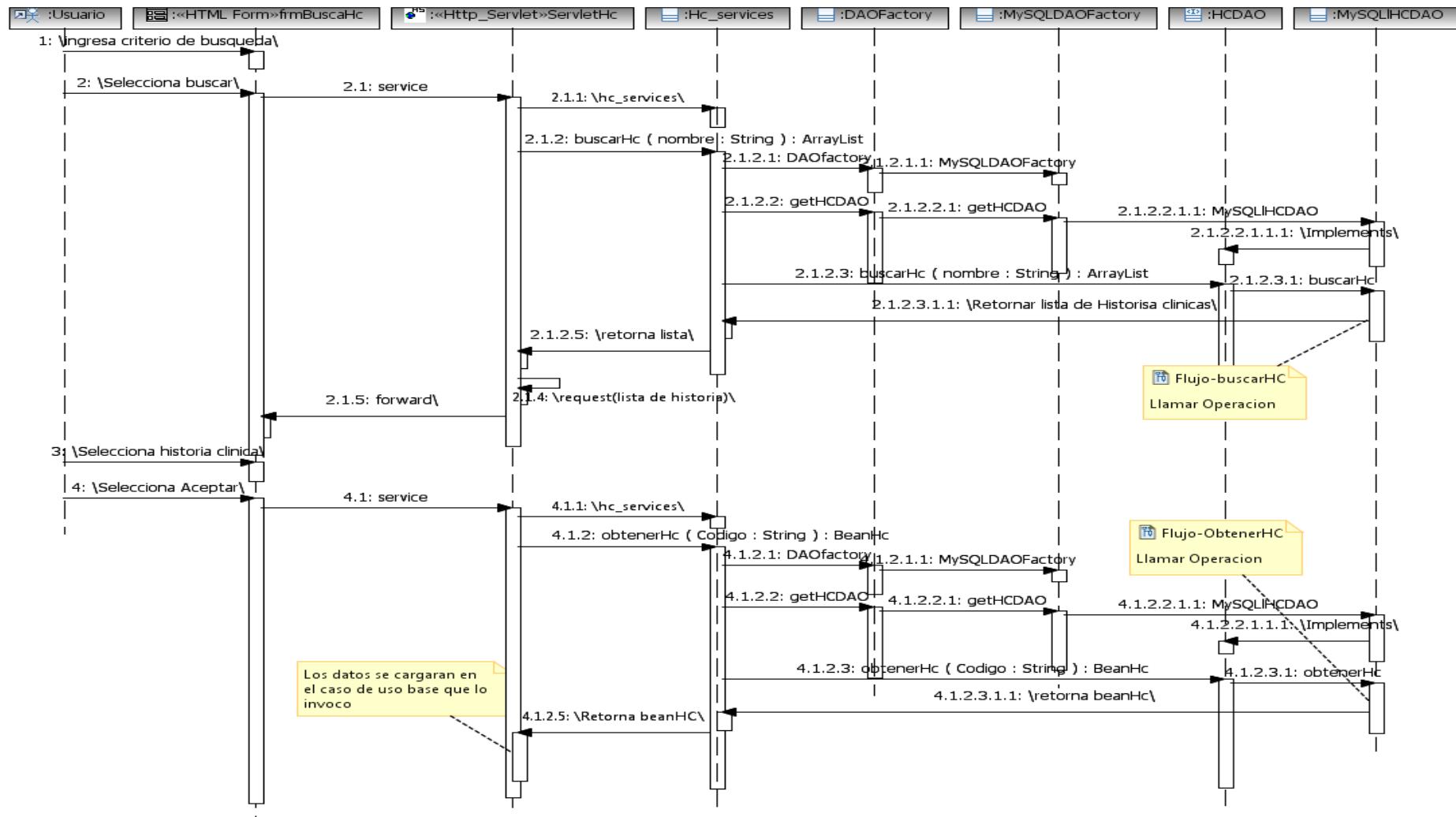
Se verá así:



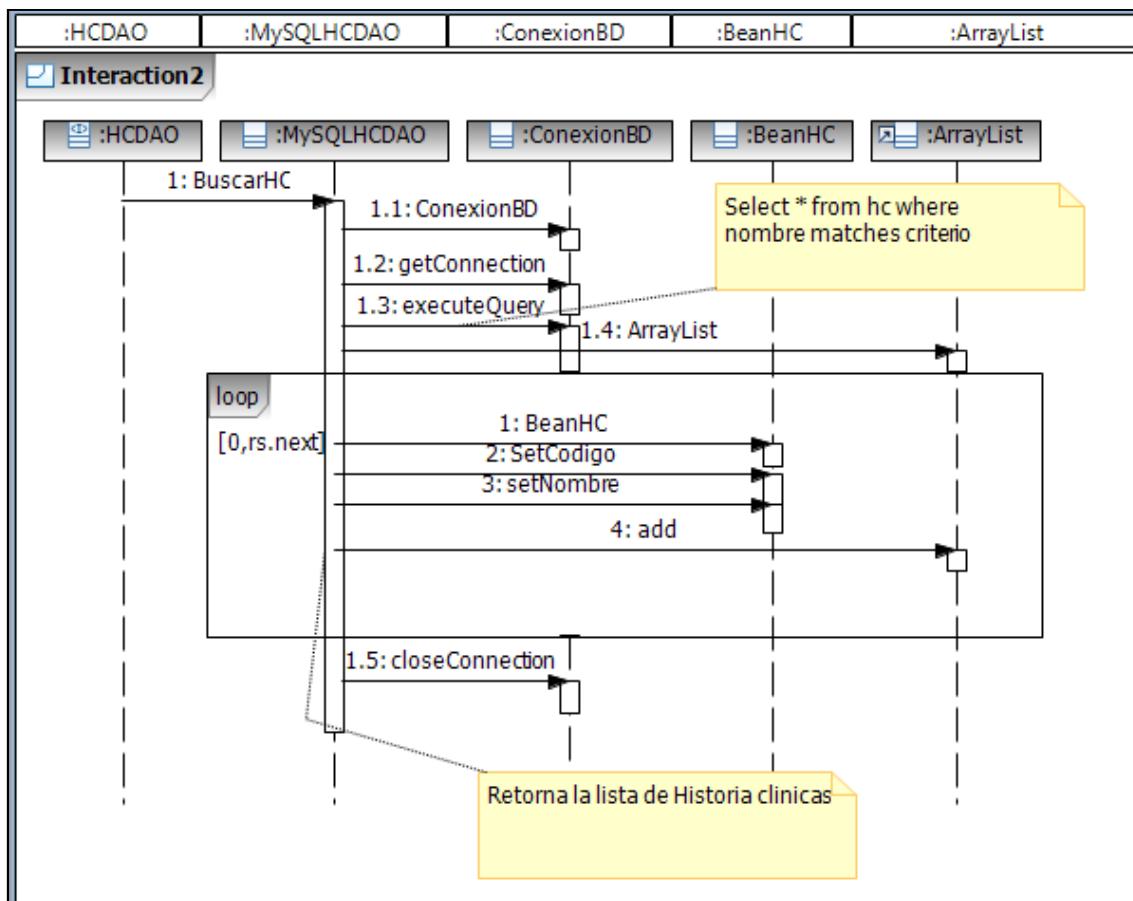
**PASO 12:** Si el mensaje es un método del objeto destino, realice lo siguiente:



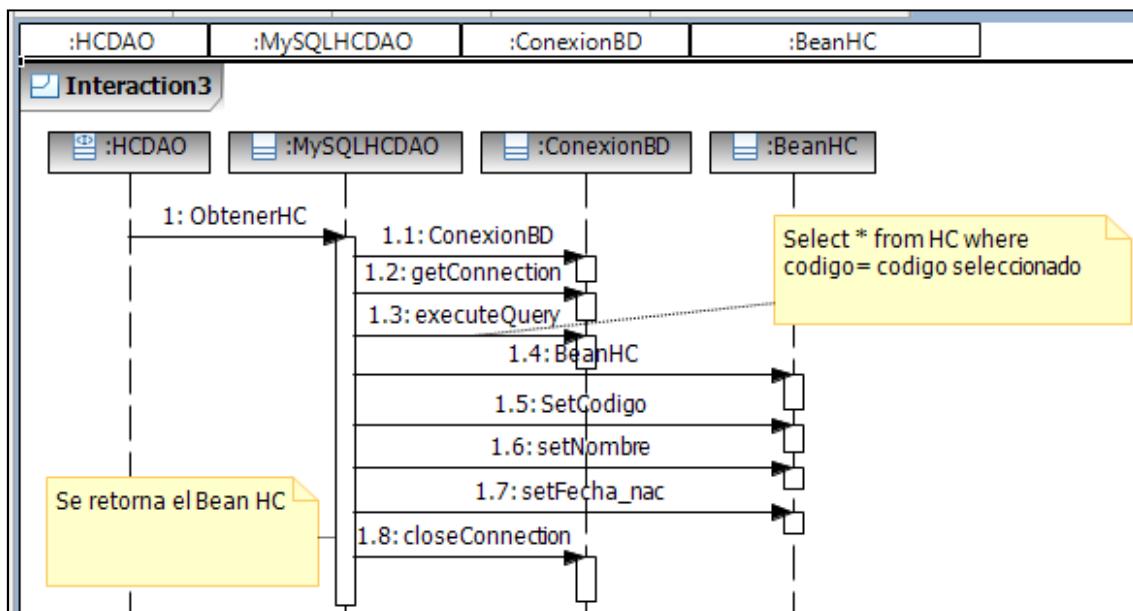
**PASO 13:** El diagrama de secuencia final para el flujo básico del CU Buscar Historia Clínica, es el que se muestra a continuación:



### :Paso 14 Flujo Buscar Buscar\_HC



### :Paso 14 Flujo Buscar Obtener\_HC



## CASO PRÁCTICO Nº 1

A partir de la Especificación de Caso de Uso, realice los siguientes artefactos:

1. Realización de Diseño del caso de uso
2. Diagrama de clases de diseño
3. Diagrama de secuencia del flujo básico
4. Diagrama de secuencia de las Operaciones

### Especificación de caso de uso: Generar Cita

#### 9. Descripción:

El caso de uso permite a la recepcionista de la clínica, registrar una cita médica para consultas externas por especialidad.

#### 10. Actor(es)

Recepcionista

#### 11. Flujo de Eventos

##### 3.1. Flujo Básico

13. El caso de uso comienza cuando la recepcionista selecciona la opción “Generar Citas” de la interfaz del menú principal.
14. El sistema muestra la interfaz “GENERAR CITAS” con la fecha y hora de registro cargado y los siguientes campos:
  - Datos de la HC: número, nombre y apellidos del paciente.
  - Datos del médico: nombres, apellidos y especialidad.
  - Datos de la cita : fecha y hora de la cita y consultorio.
  - Además, presenta las opciones: Buscar HC, Buscar Horarios de Médico y Grabar Cita.
15. La recepcionista selecciona “Buscar HC”.
16. El sistema **incluye el caso de uso Buscar Historia Clínica**.
17. El sistema muestra los datos de la historia clínica del paciente.
18. La recepcionista selecciona “Buscar Médico”.
19. El sistema **incluye el caso de uso Buscar Horarios de Médico**.
20. El sistema muestra los datos del médico y de la cita.
21. La recepcionista selecciona “Grabar Cita”.
22. El sistema valida los datos.
23. El sistema genera el número de cita y registra la cita con estado pendiente.
24. El sistema muestra el mensaje “Cita generada”. El caso de uso termina.

##### 3.2. Flujos Alternativos

###### 1. No existe HC

Si en el paso 6 el sistema detecta que no existe la HC del paciente, muestra el MSG “No existe HC” y ofrece la posibilidad de registrar la HC del paciente.

###### 2. No hay médicos disponibles

Si en el paso 8 el sistema detecta que no hay médicos disponibles, muestra el MSG “No hay médicos disponibles” y el caso de uso finaliza.

**3. Campos vacíos**

Si en el paso 10 el sistema detecta que alguno de los campos está vacío, muestra el MSG “Alguno de los campos está vacío” y el caso de uso continúa.

**12. Precondiciones**

4. La recepcionista está identificada en el sistema.
5. Lista disponible de historias clínica.
6. Lista disponible de médicos.

**13. Poscondiciones**

2. En el sistema quedará registrada la cita en estado pendiente.

**14. Puntos de Extensión**

En el paso 6, el sistema extiende al caso de uso Mantener HC – subflujo “Registrar HC”.

**15. Requisitos Especiales**

Ninguno.

**16. Prototipos**

### Generar Citas

---

Fecha: 08/03/2010  
Hora: 15:32

**Datos de la HC del paciente**

Nº HC:

Paciente:

**Datos del médico**

Nombre y apellidos:

Especialidad:

**Datos de la cita**

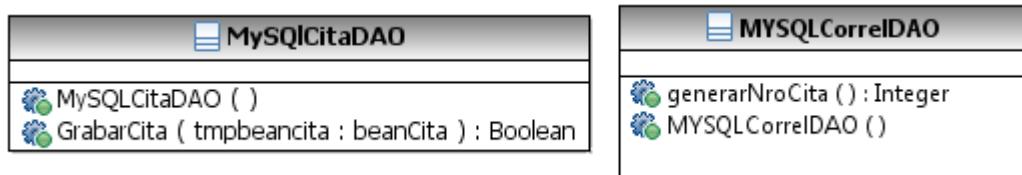
Fecha:

Hora:

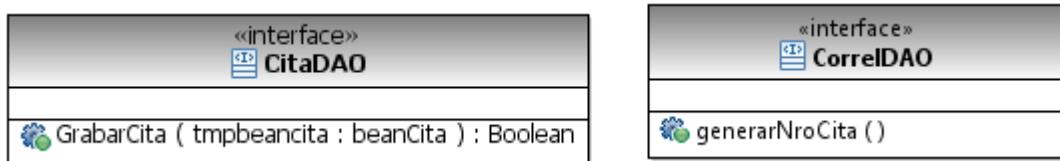
Consultorio:

## **Paso 1:** Crear las clases de Negocio en el paquete Cita

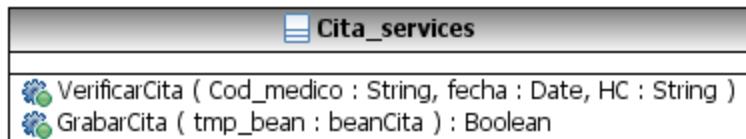
Crear las clases DAO MySQLCitaDAO y MYSQLCorrelDAO y BeanCita



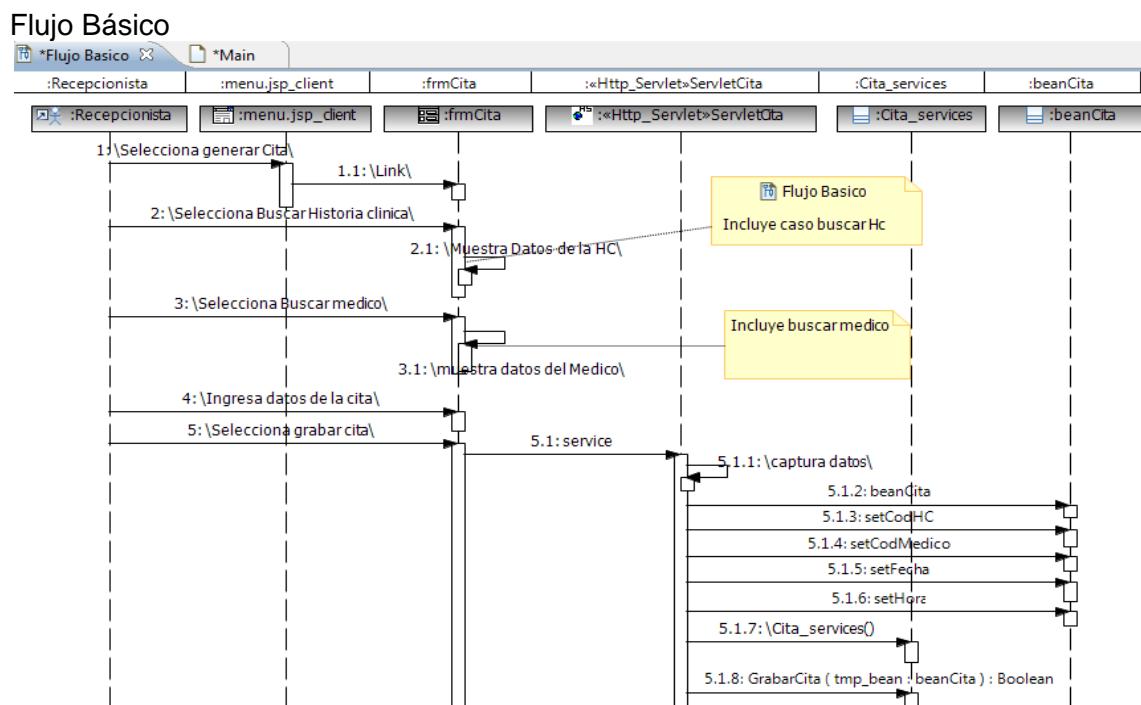
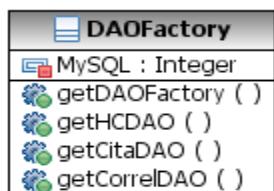
## Crear las clases Interfaces CitaDAO y CorreIDAO

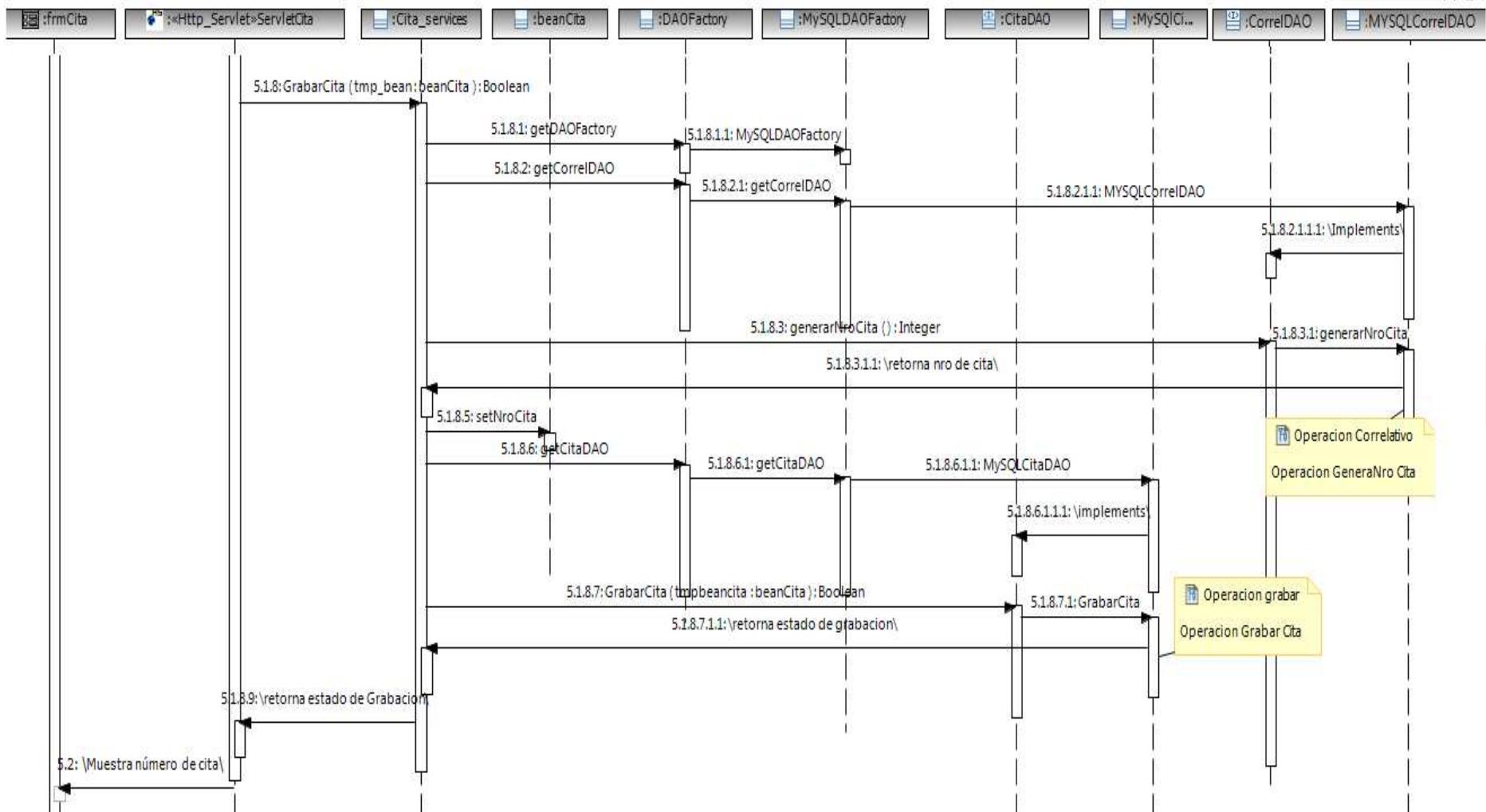


## Crear la clase cita\_services

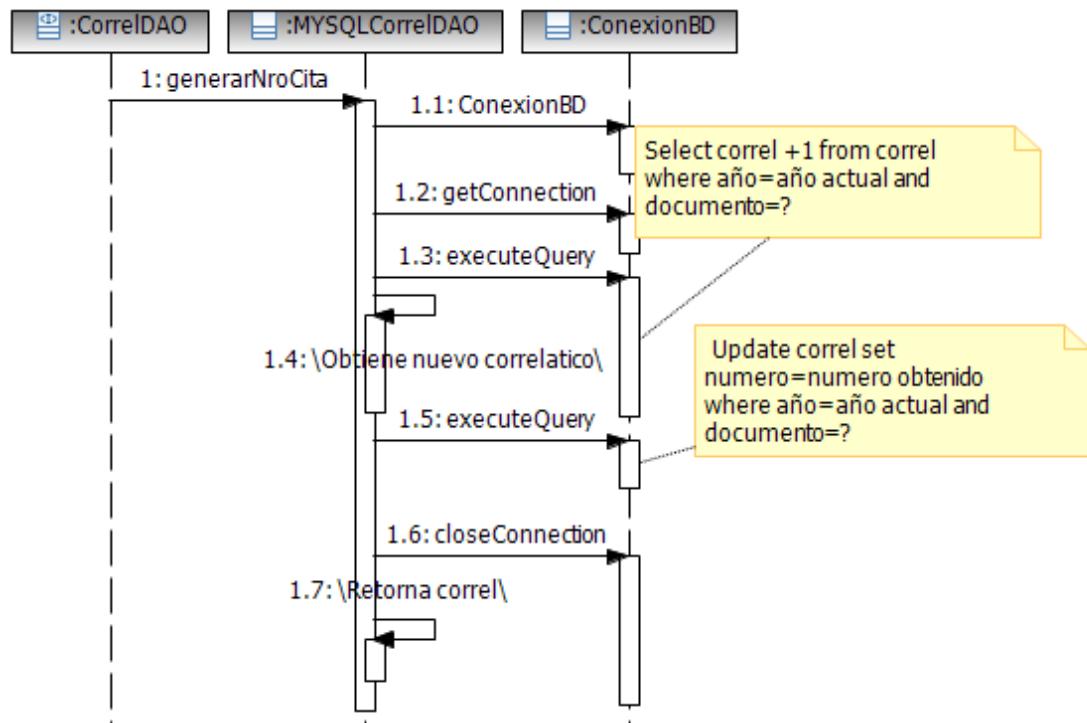


## Agregar los métodos al DAOFactory

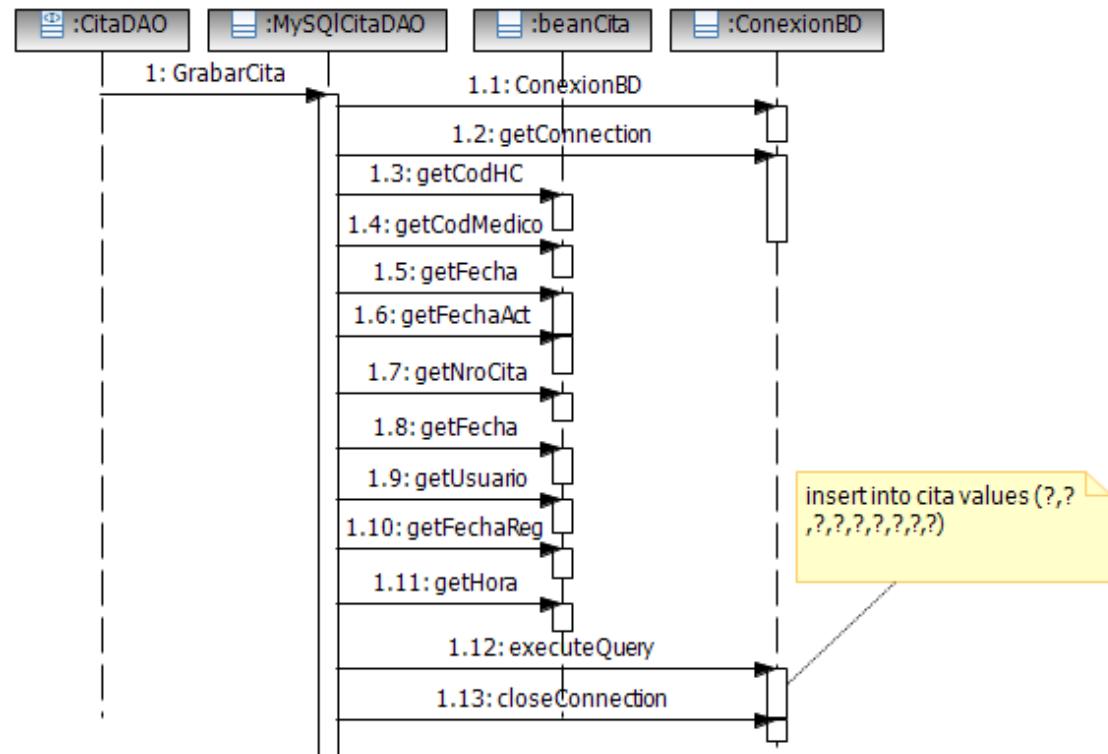




## Operación generaNroCita



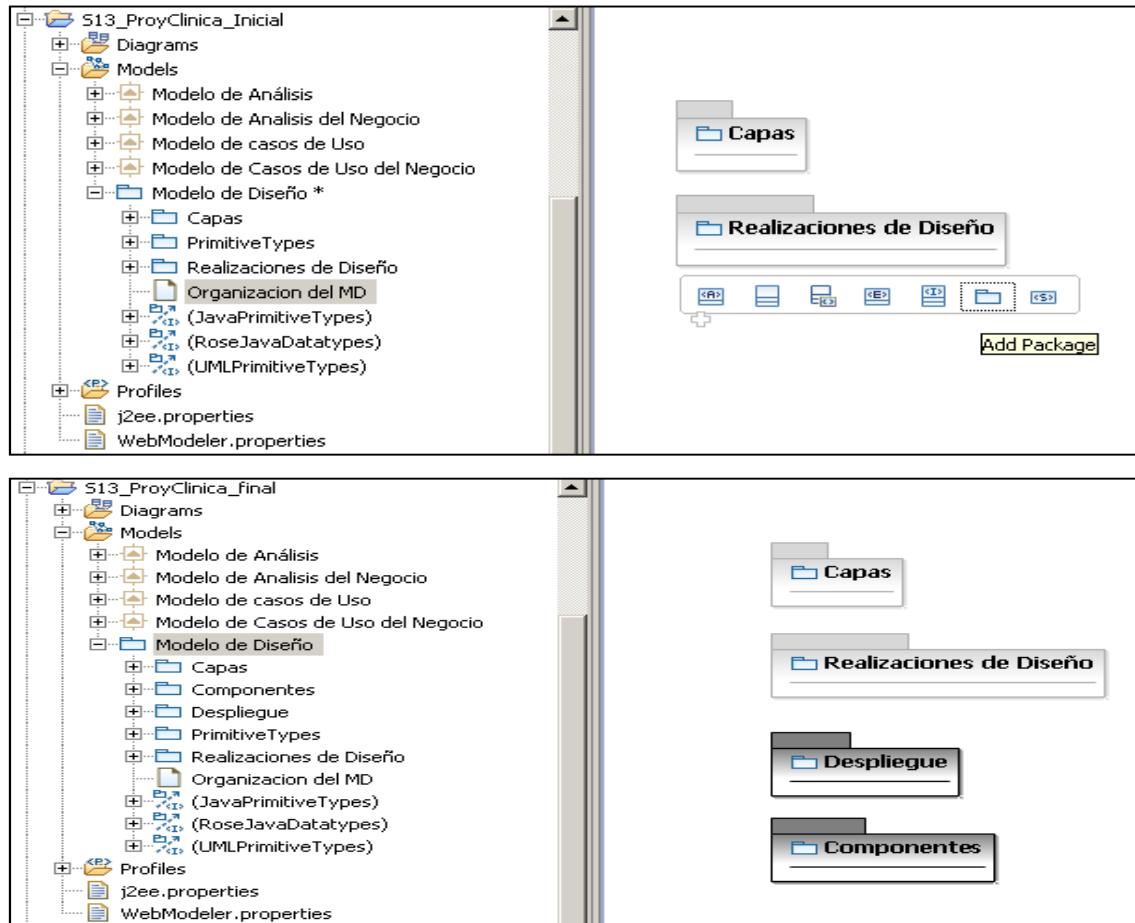
## Operación grabar



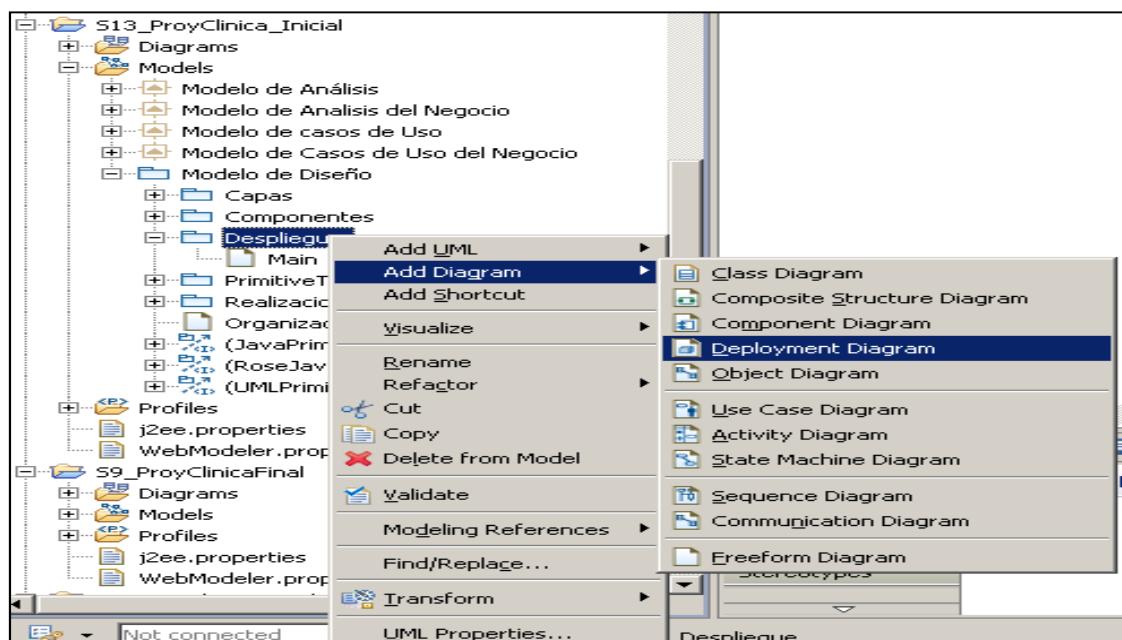
## 5 Modelo de Diseño

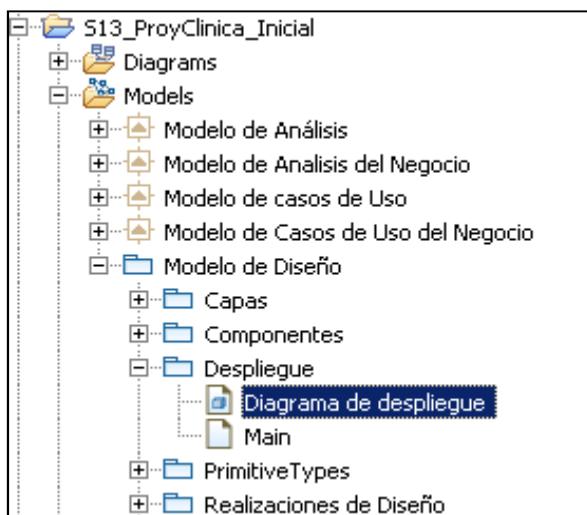
### Diagrama de componentes y despliegue

**PASO 1:** Crear 2 paquetes, uno para Despliegue y otro para Componentes en el modelo de diseño en la organización del MD.

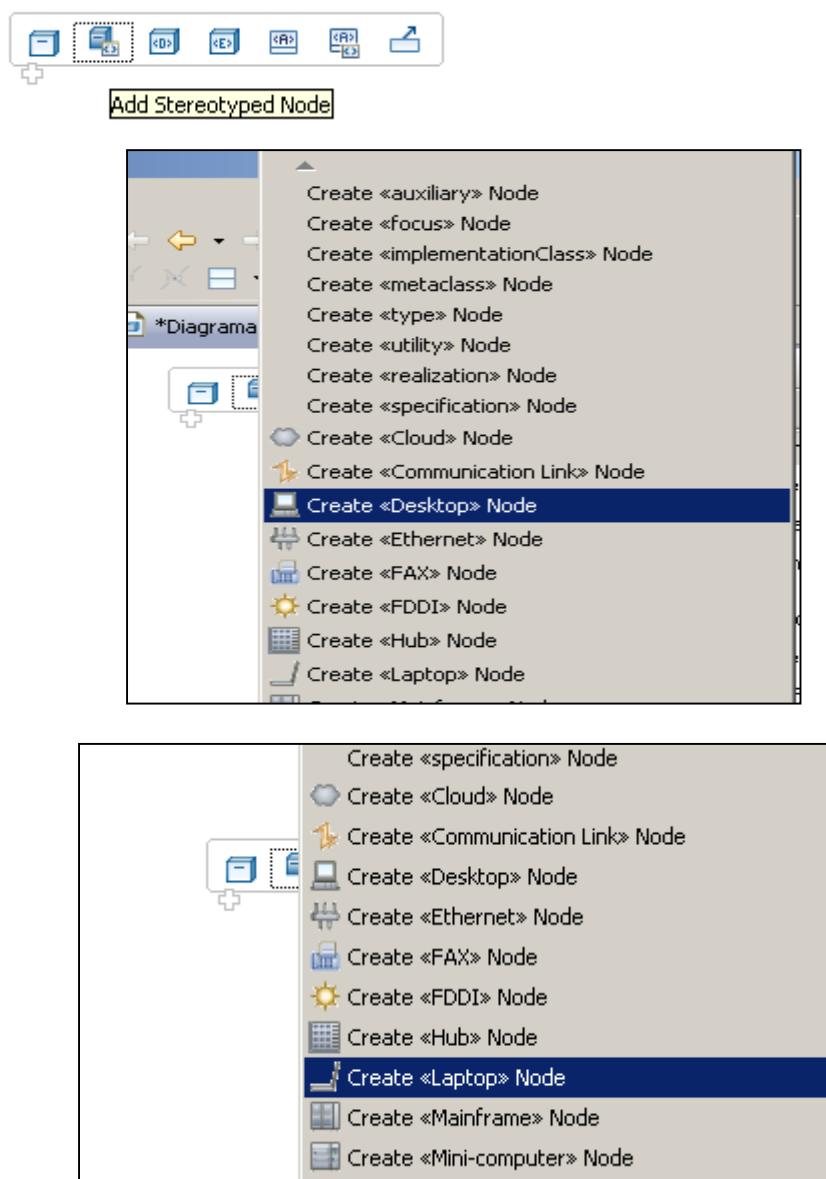


**Paso 2: Agregar un diagrama de despliegue en el paquete Despliegue**

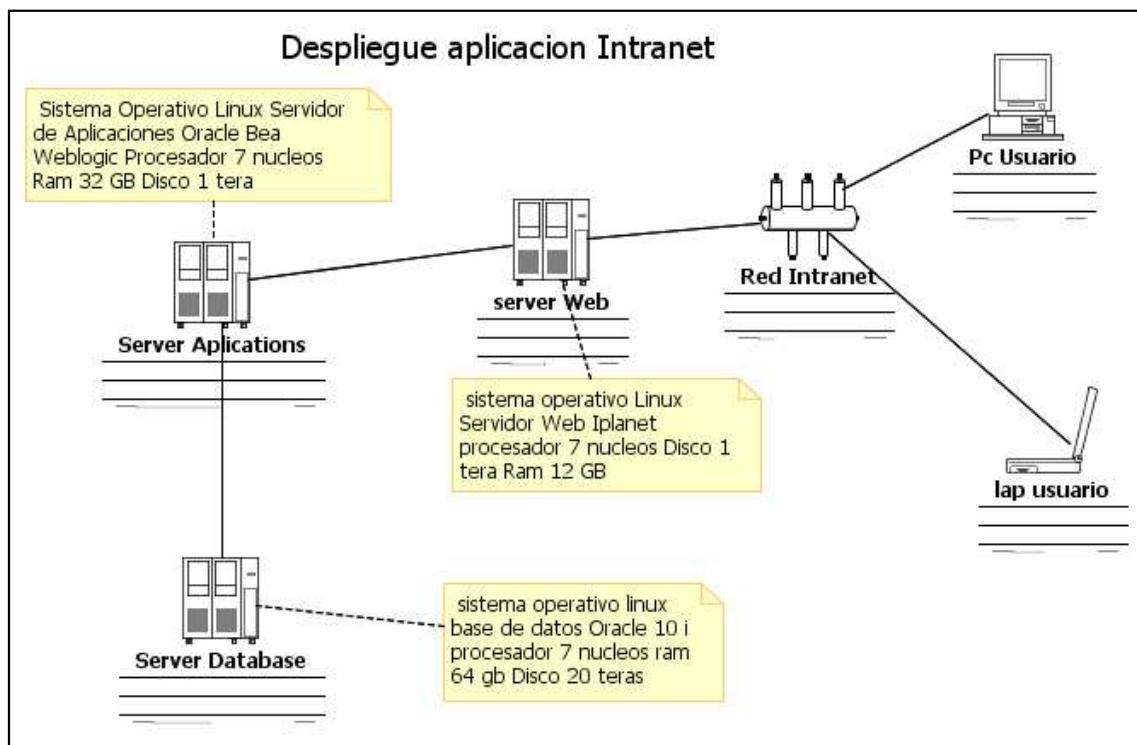




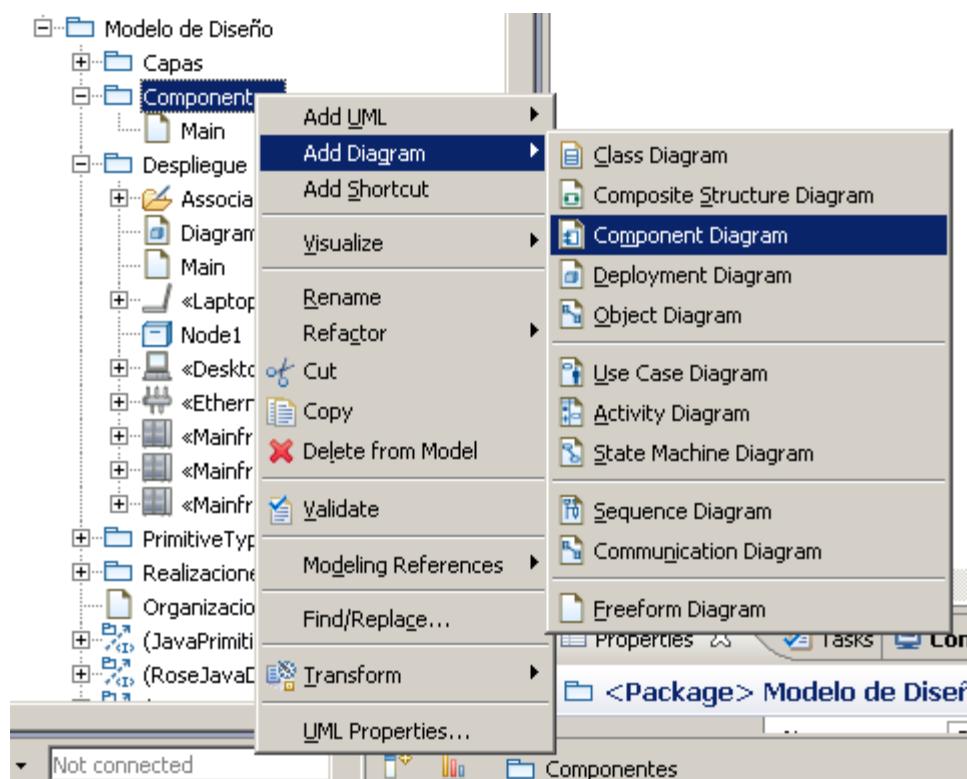
PASO 3: Agregar los nodos necesarios para armar la arquitectura de una intranet y colocamos en mensajes la especificación de cada nodo.



**PASO 4:** Relacionar los Nodos con el Communication PATH, quedando así:

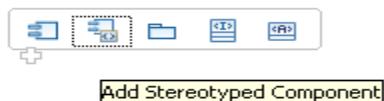


**PASO 5:** Agregar en el paquete de componentes un diagrama de componentes, el cual llamamos diagrama de componentes.



**PASO 6:** Agregamos los componentes necesarios. A continuación, detallamos cómo agregar componentes:

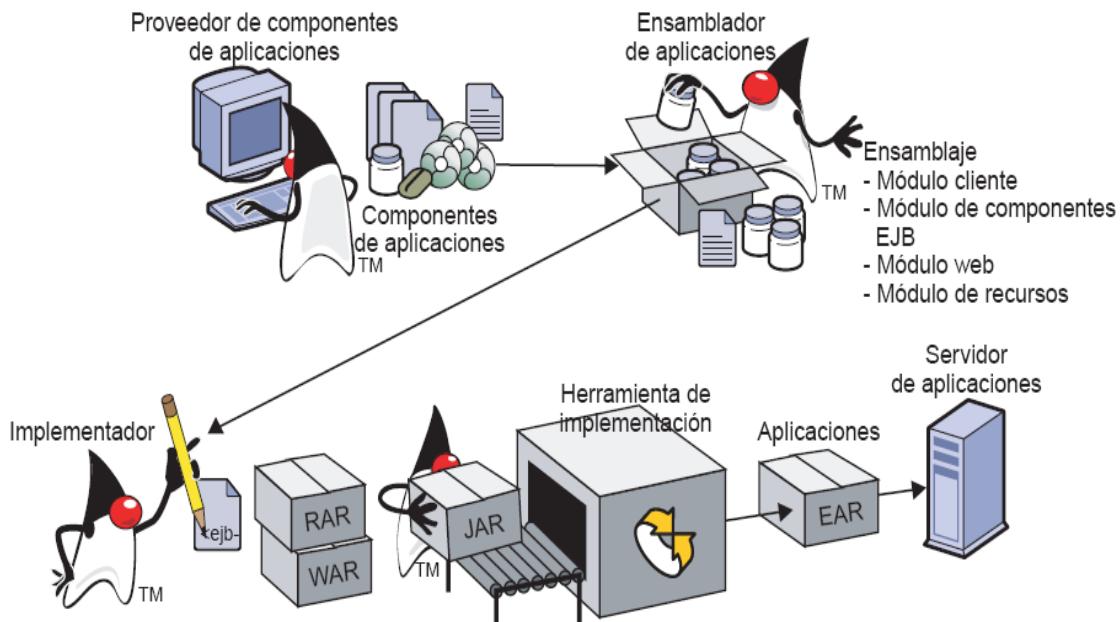
Seleccionamos el estereotipo de componente



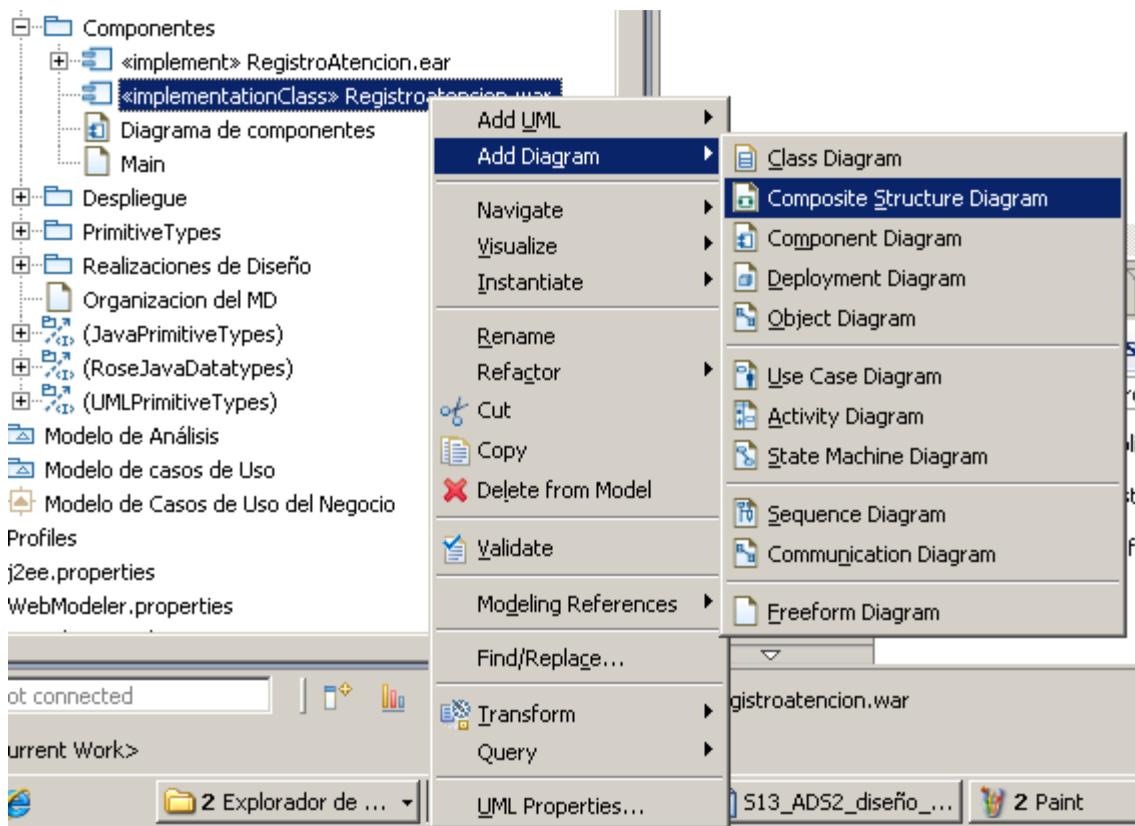
**Paso 7:** Agregamos 2 componentes 1 componente RegistroAtencion.war RegistroAtencion.ear.



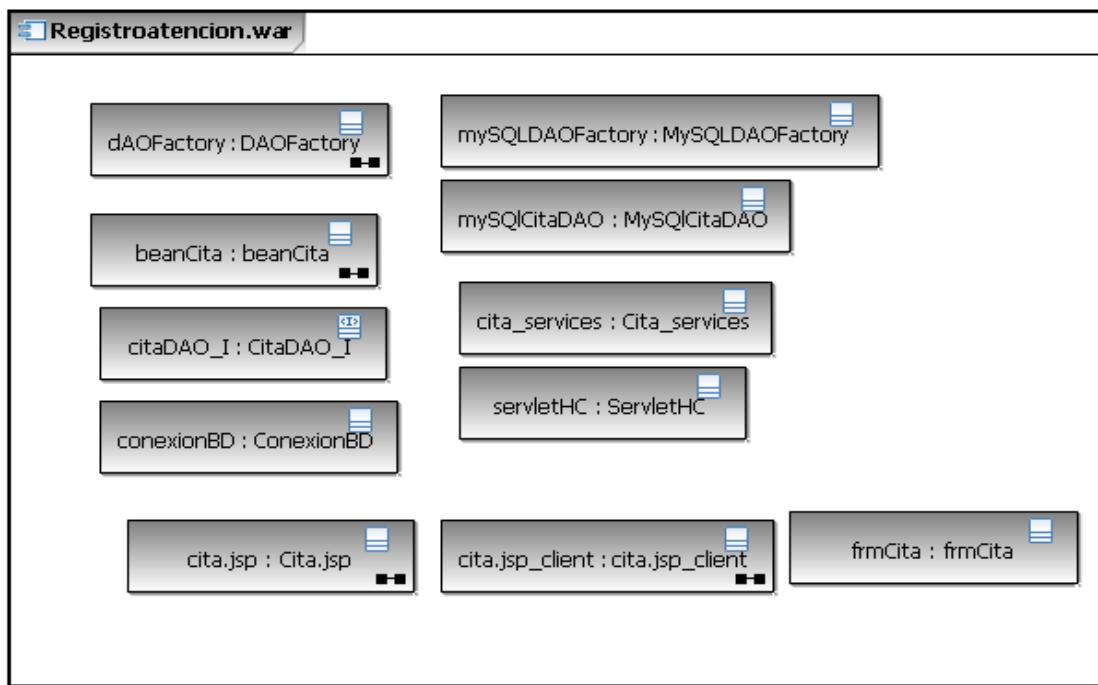
Basados en el concepto de instalación de JEE el War contiene todas las clases y dentro del ear colocamos solo al war



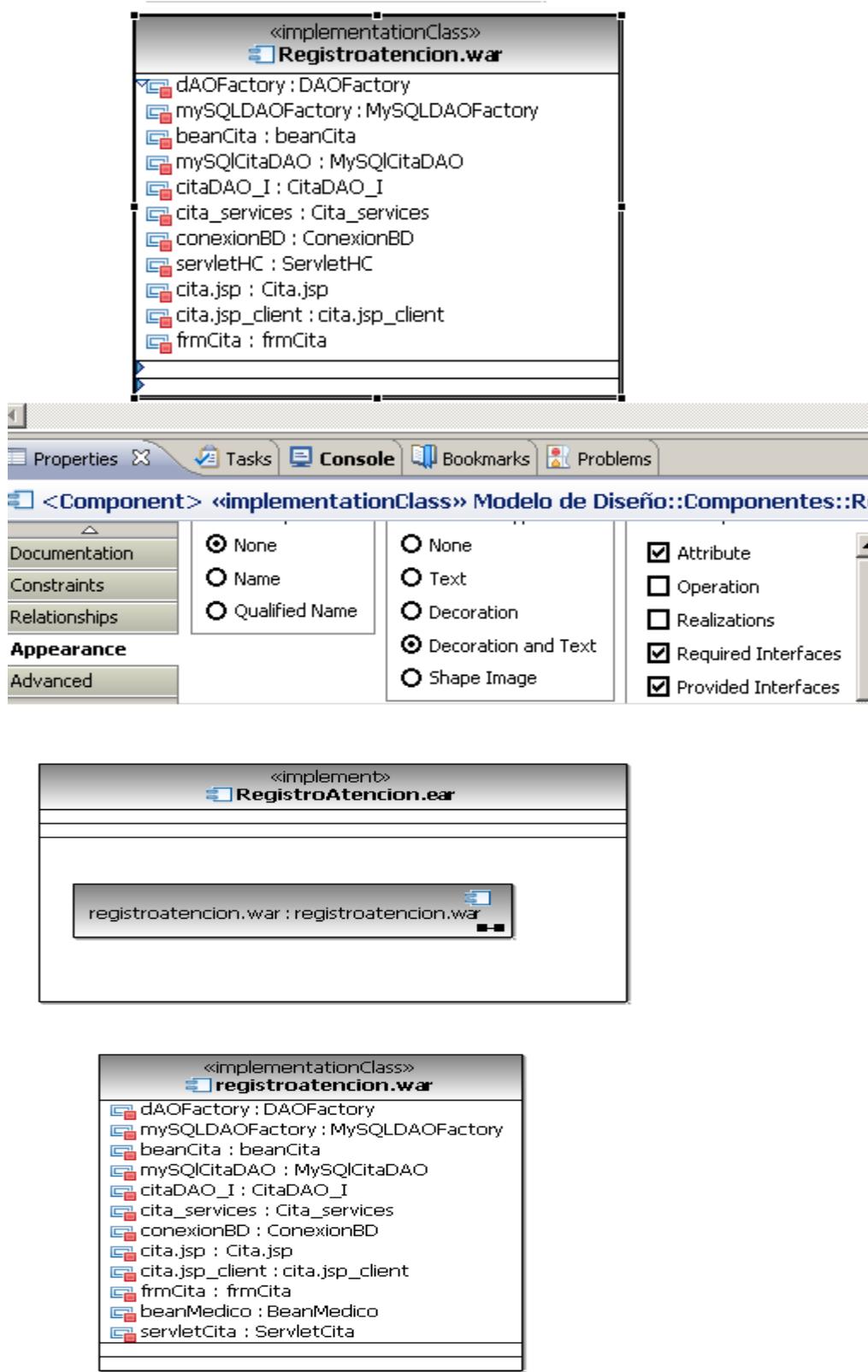
**PASO 8:** Dentro del componente RegistroAtencion.war creamos el diagrama de estructura compuesta y ahí colocamos las clases del proyecto.



Paso 10: Arrastramos todas las clases al diagrama de implementación creado.



Paso 11: Mostramos el RegistroAtencion.war con los elementos, seleccionando apariencia y el *check attribute*, y en el ear creamos lo mismo y agregamos el war.



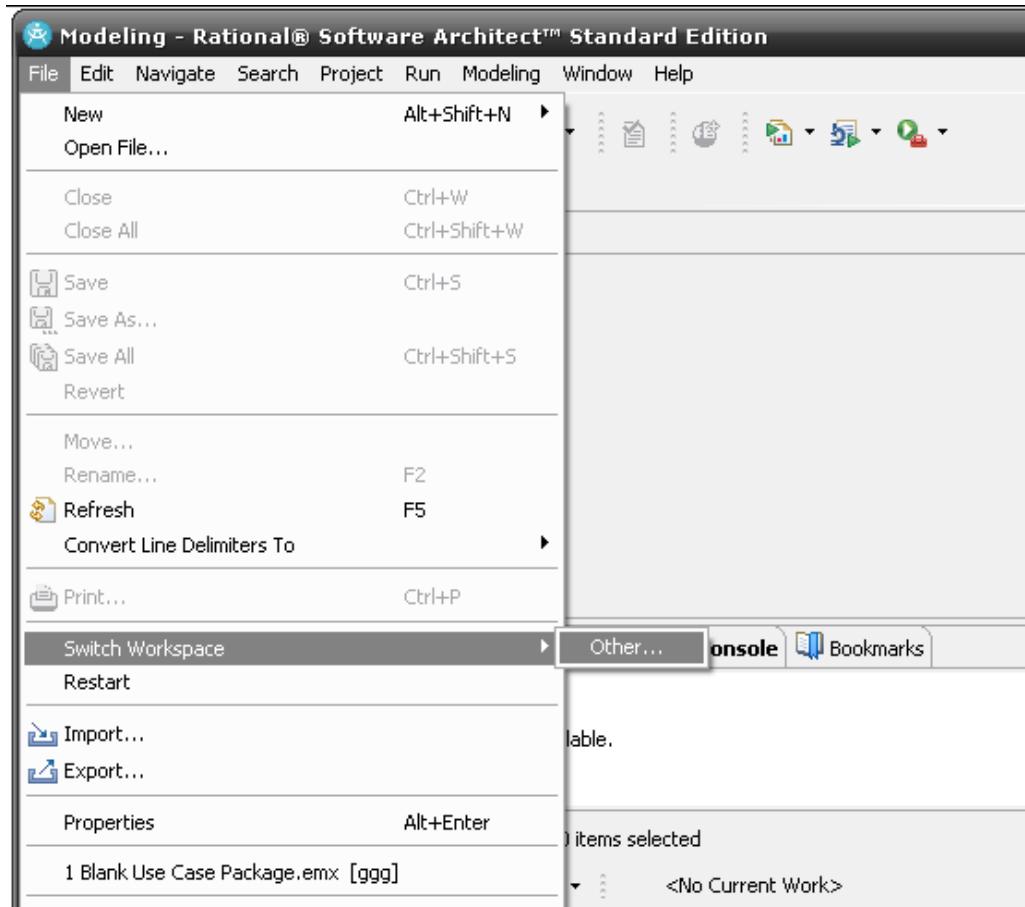
# ANEXOS

## CONTENIDO

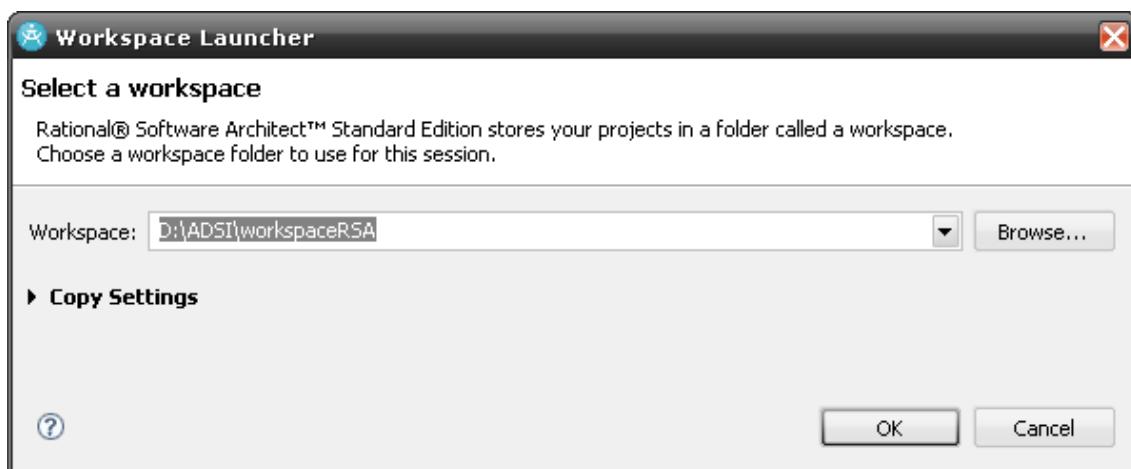
- Cambio de *workspace*
- Importación de proyectos
- Publicación de modelos
- Ingeniería reversa de código fuente a clases de diseño

## CAMBIO DE WORKSPACE

- Para cambiar el workspace actual, seleccione **File/Switch Workspace/Other...**



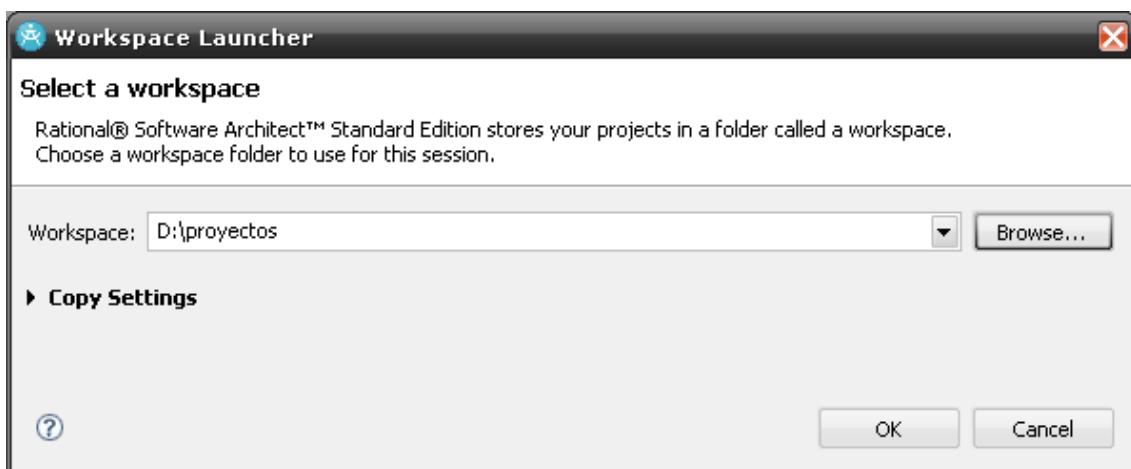
- A continuación, se mostrará en Workspace la ruta del espacio de trabajo actual. Debe dar clic a **Browse...** para ubicar la ruta del nuevo workspace.



3. Desde este explorador, ubique el directorio del nuevo workspace. Además, tiene la opción de crear otro directorio con el botón **Crear nueva carpeta**. Luego, dé clic en **Aceptar**.

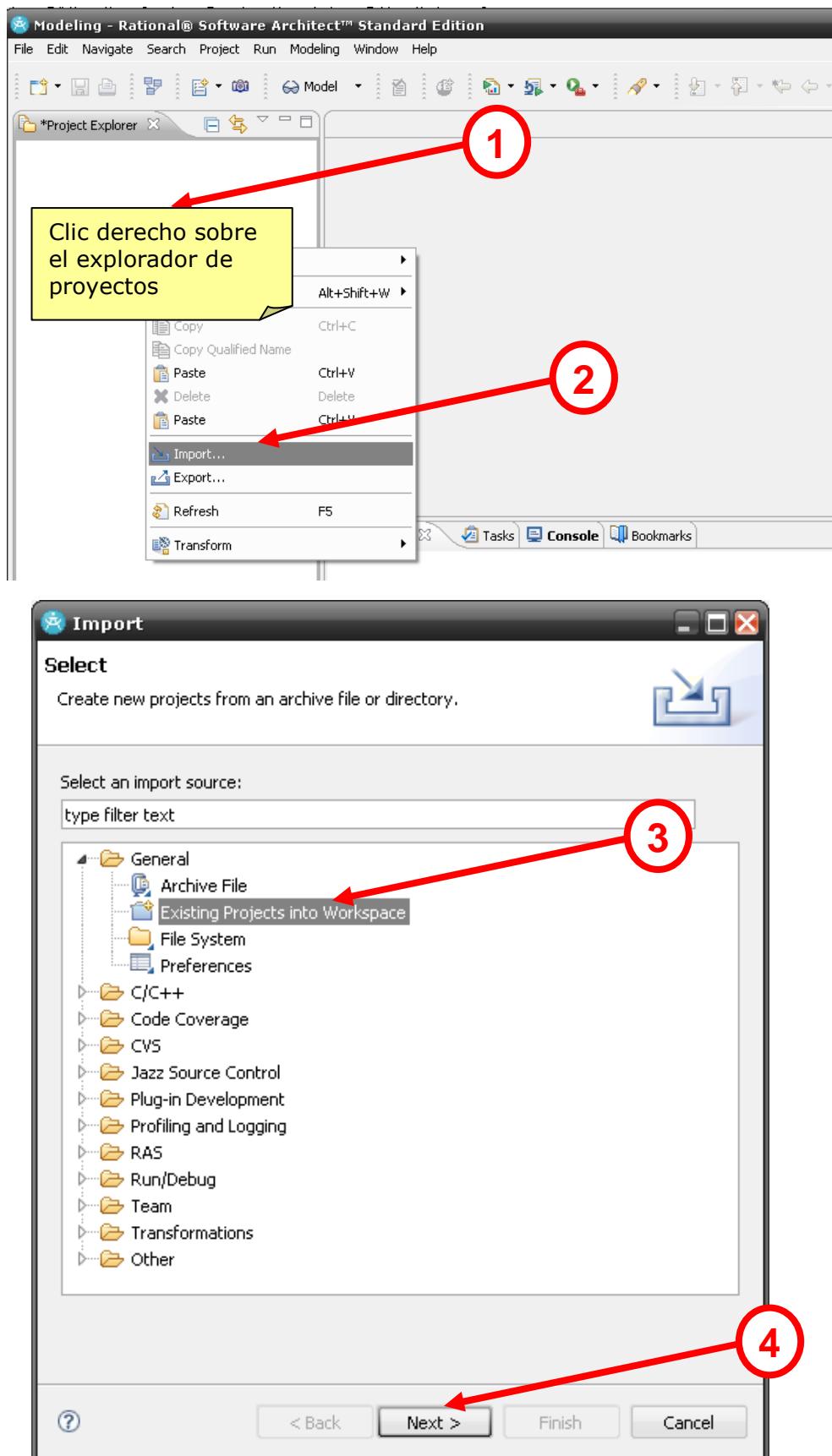


4. A continuación, se mostrará la ruta del nuevo workspace. Para finalizar dé clic en **OK** para que el IBM RSA se reinicie con el nuevo espacio de trabajo.

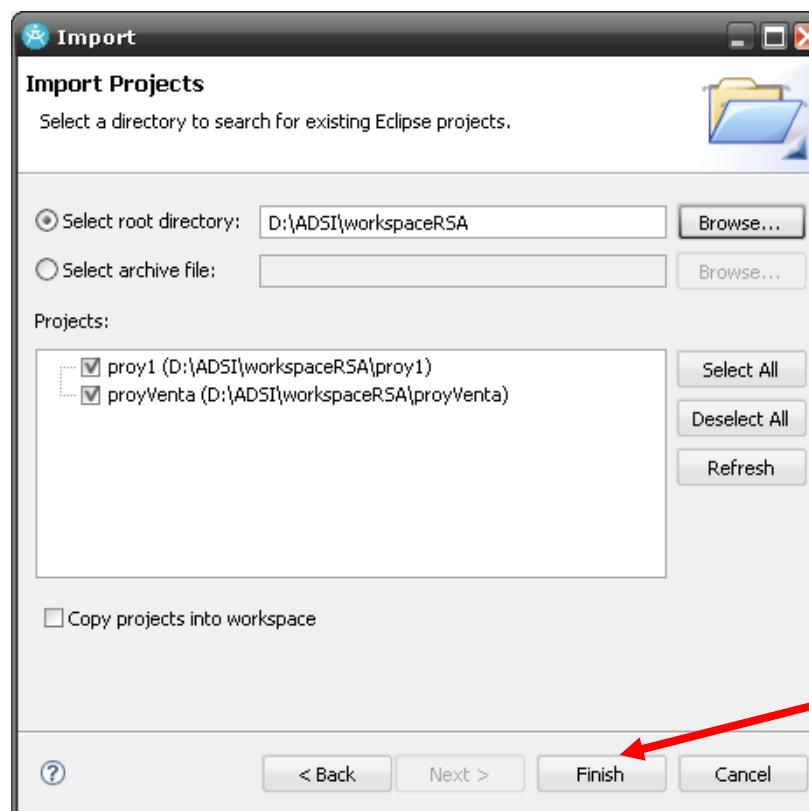
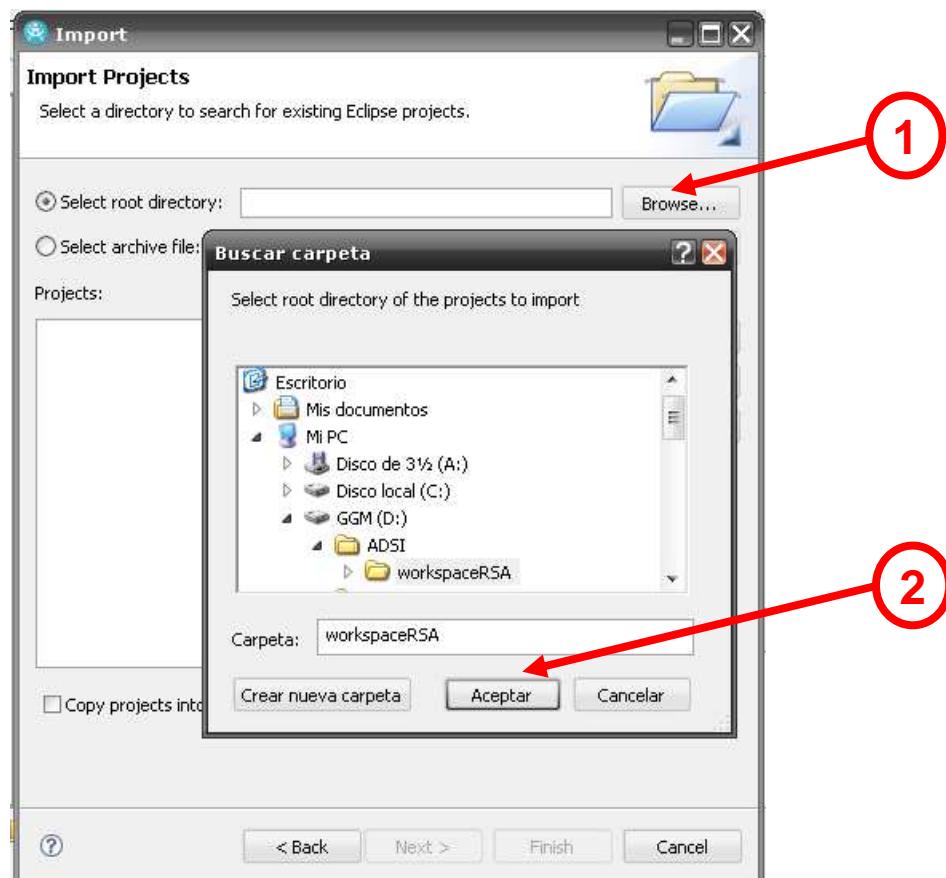


## IMPORTACIÓN DE PROYECTOS

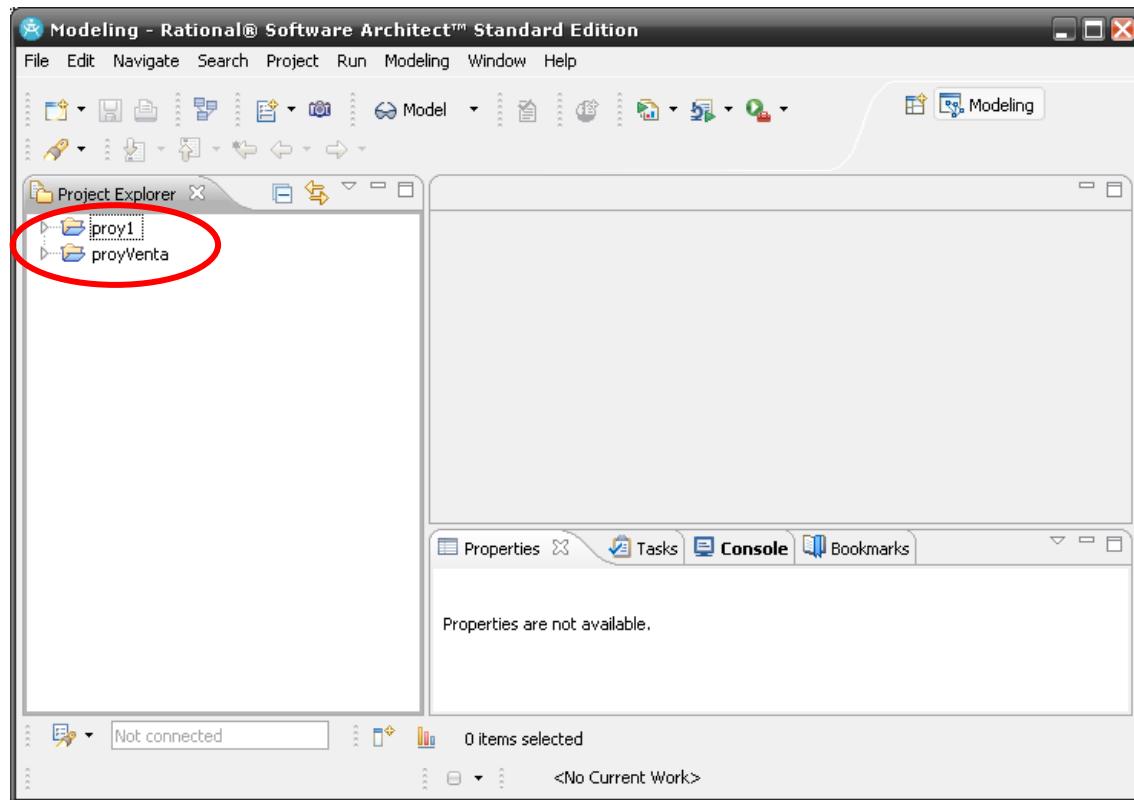
1. Seleccione la fuente de importación.



2. A continuación, seleccione el workspace configurado, el cual contiene proyectos a importar.

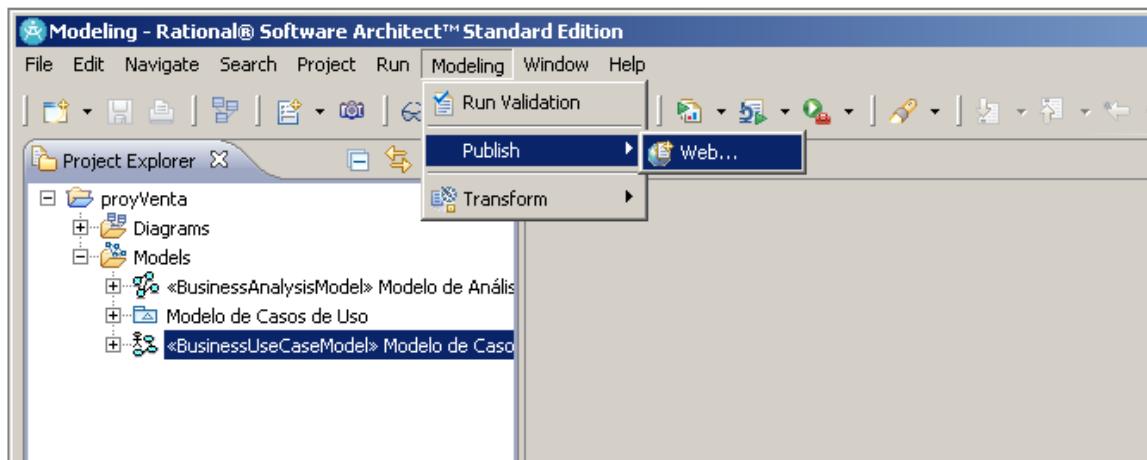


3. Por último, en el explorador de proyectos, se mostrará la lista de proyectos importados.

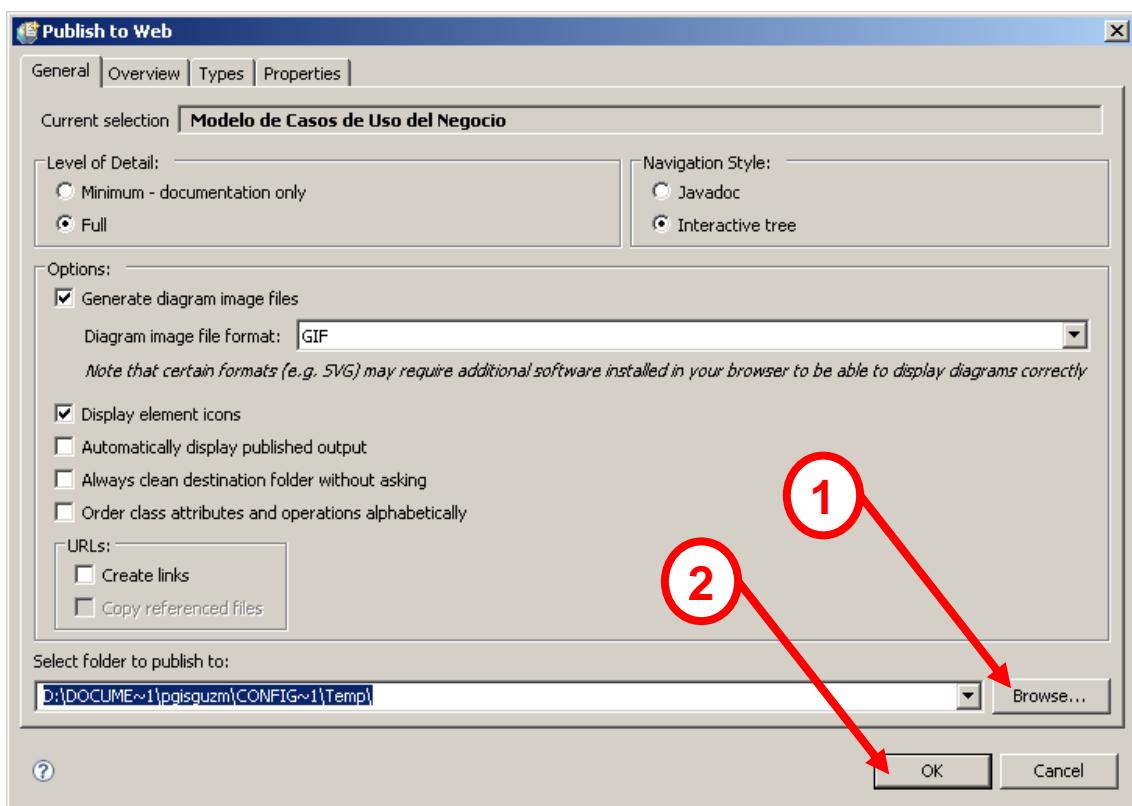


## PUBLICACIÓN DE MODELOS

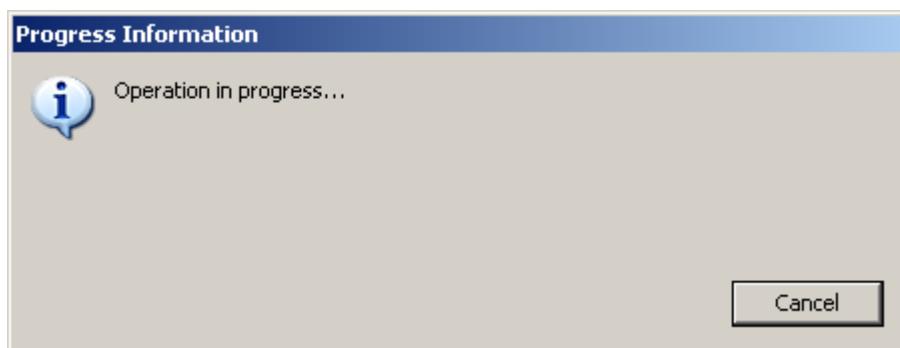
- Para publicar los modelos de un proyecto, seleccione el modelo y luego, en la barra de menú, seleccione **Modeling / Publish / Web...**



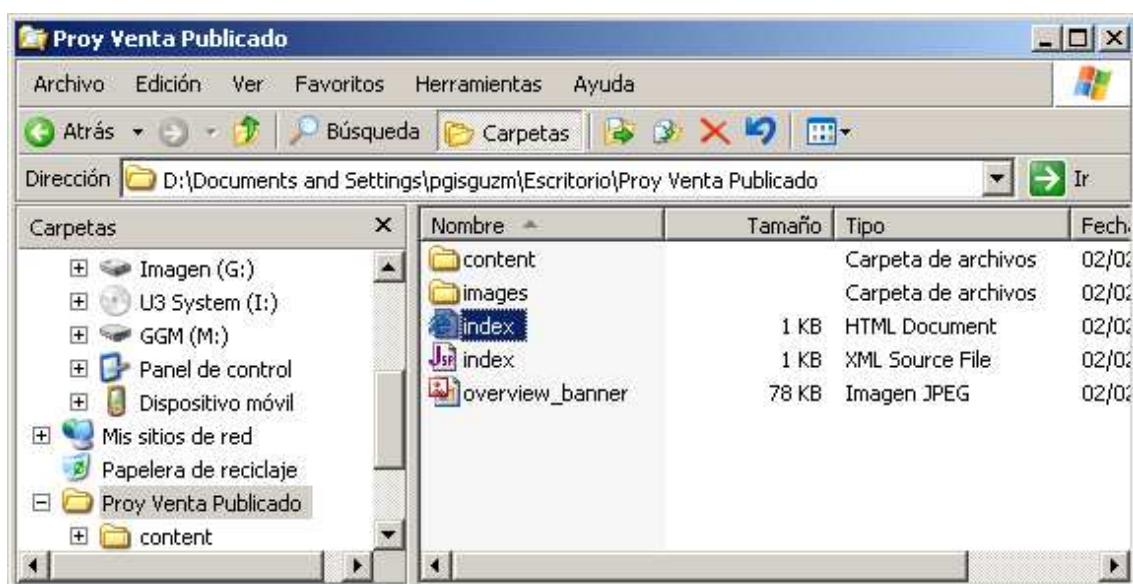
- Especifique folder a publicar.



3. Espere unos breves minutos.

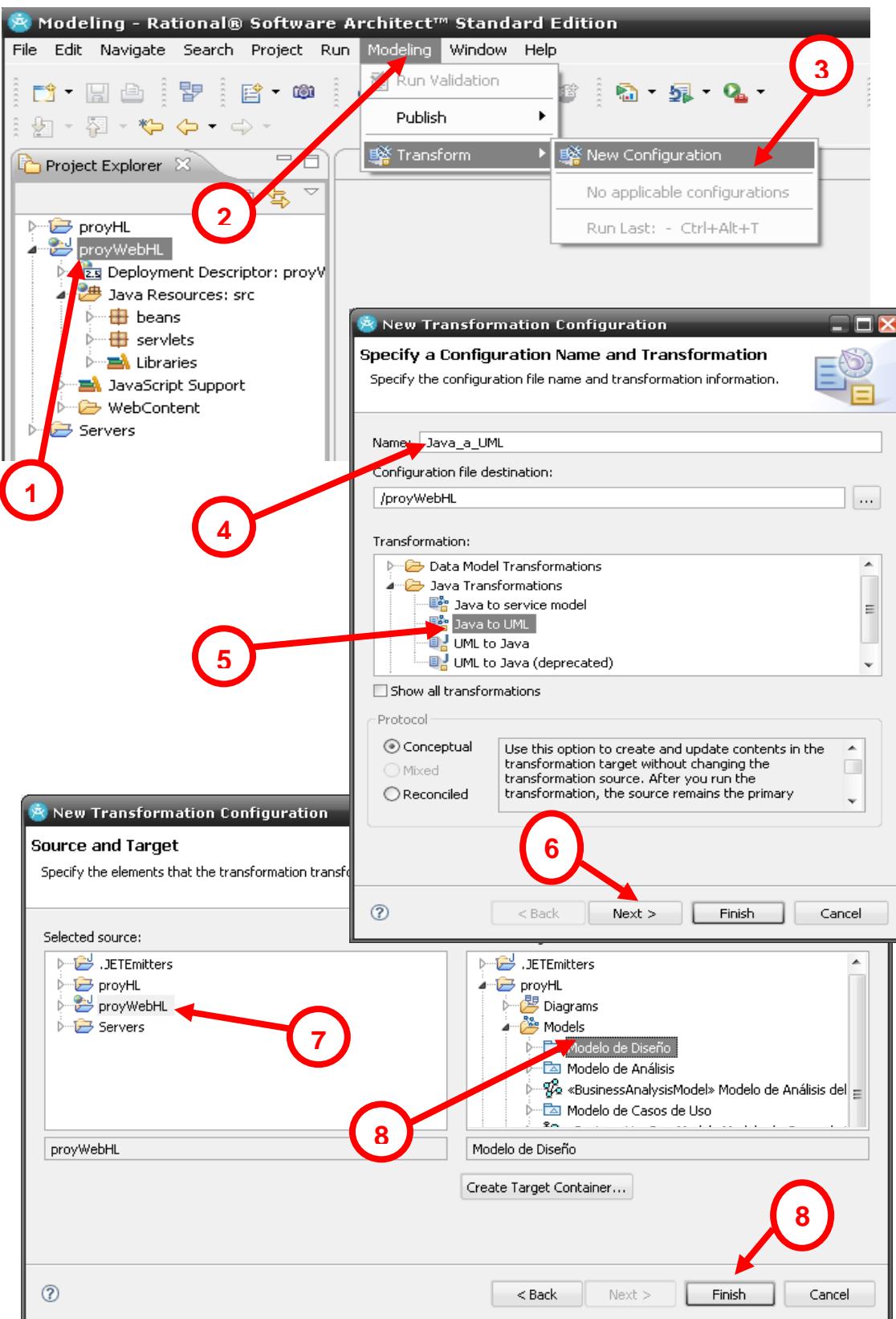


4. Por último, podrá visualizar el modelo publicado desde la página **index.html**

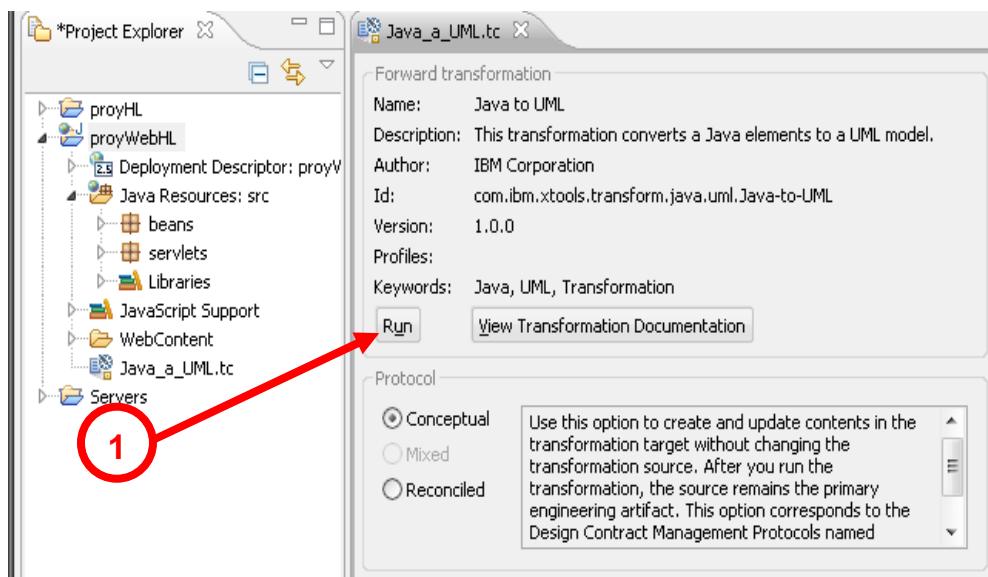


## INGENIERÍA REVERSA DE CÓDIGO FUENTE A CLASES DE DISEÑO

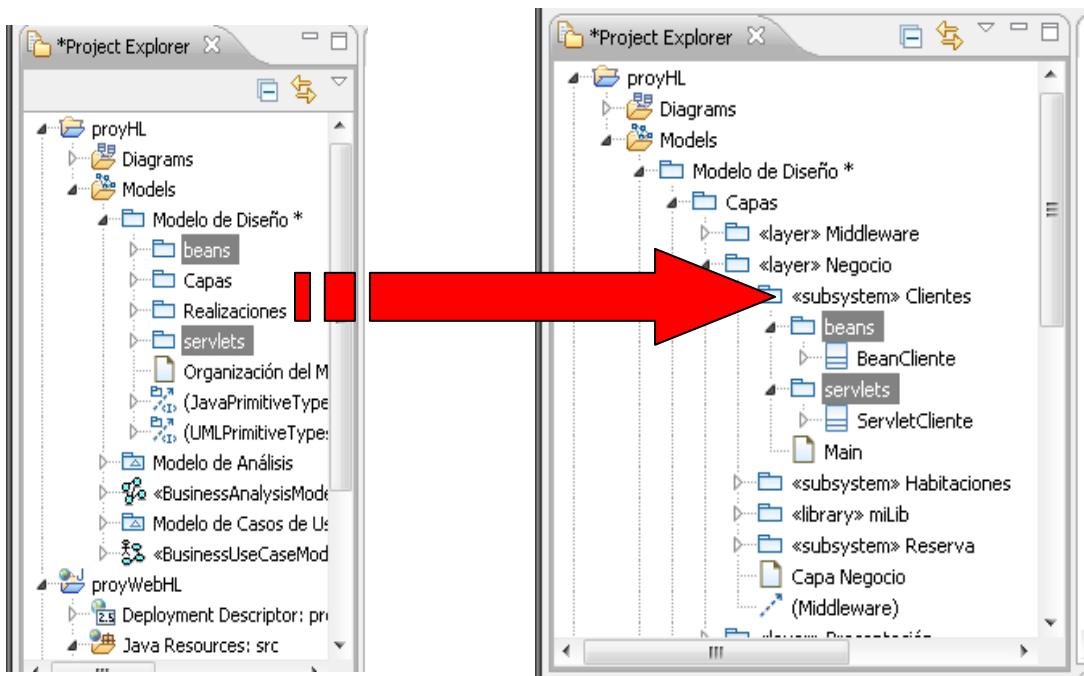
**PASO 1:** Crear la configuración para transformar el código de las clase JAVA a UML.  
Asegúrese de que la perspectiva a **Modeling** esté activada.



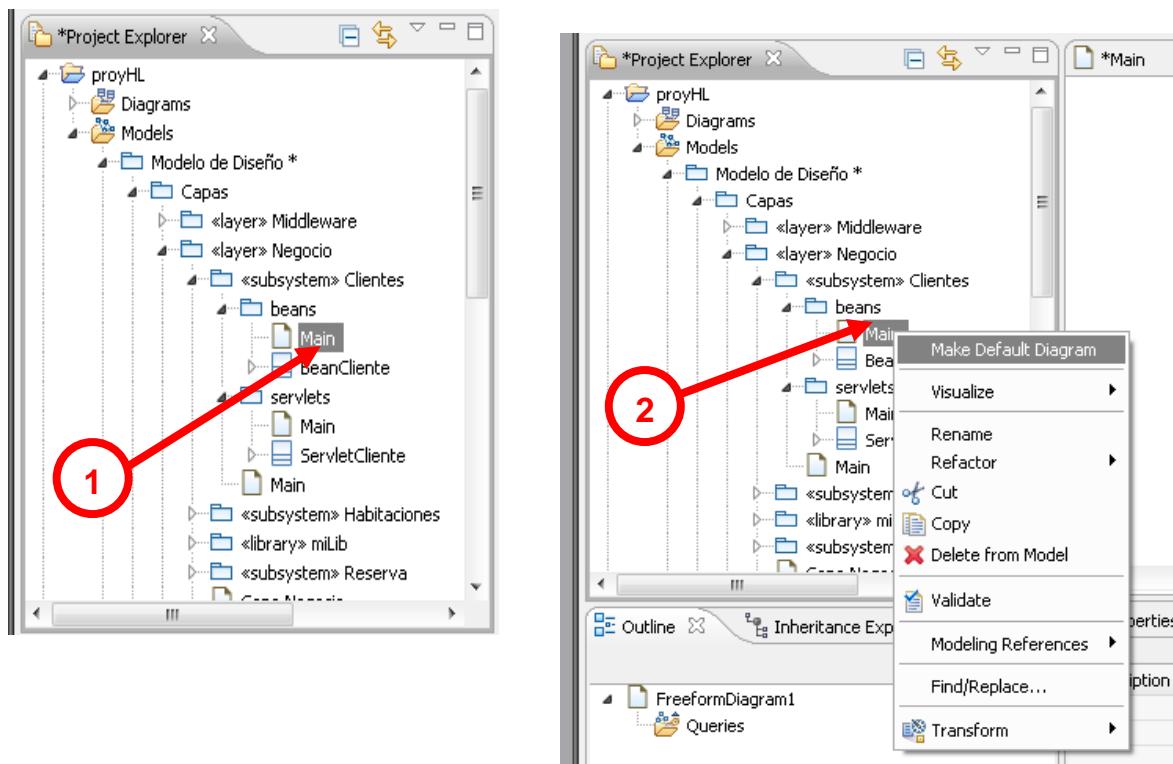
**PASO 2:** Ejecute la transformación.



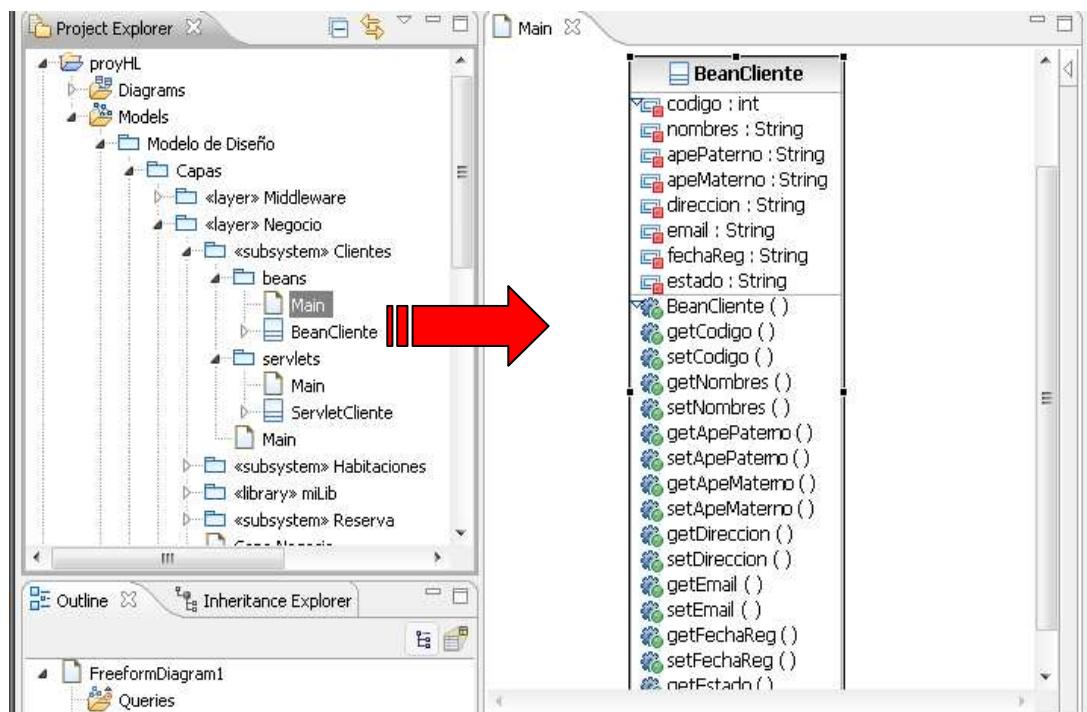
**PASO 3:** Mueva los paquetes **beans** y **servlets** del Modelo de Diseño al **subsistema Clientes de la Capa Negocio**.

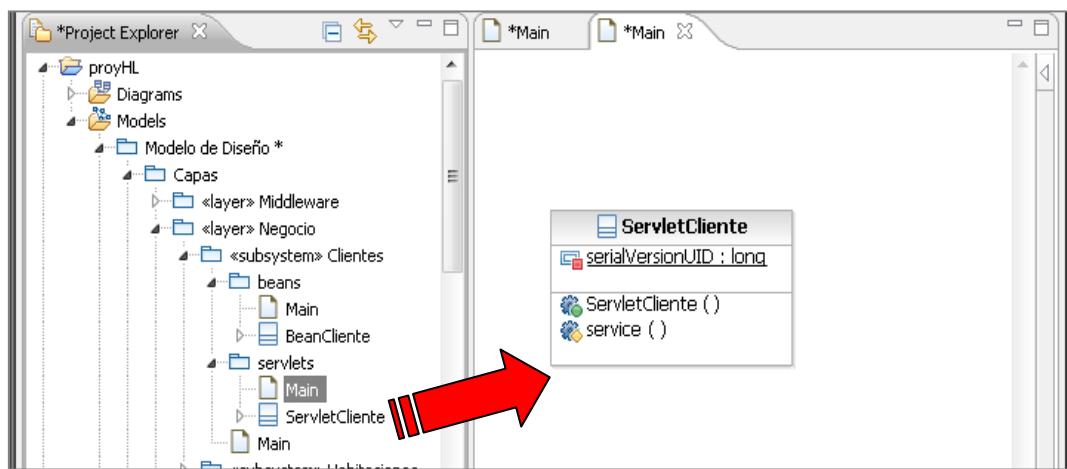


**PASO 4:** Cree un diagrama **freeform** de nombre **Main** en los paquetes beans y servlets. Luego, actívelos como diagramas por defecto a cada paquete:



**PASO 5:** Arrastre las clases generadas en cada Main del paquete:





# Glosario

## Abstracción

Características esenciales de una entidad que la distingue de otros tipos de entidades. Define una frontera desde la perspectiva del observador.

## AORE Aspect-Oriented Software Requirement

Ingeniería de requisitos orientada a aspectos, la cual provee un conjunto de enfoques para gestionar intereses y requisitos transversales que podrían modularizarse para luego, componerlos con otros intereses.

## API

Una API representa una interfaz de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software.

## Artefacto

Pieza discreta de información que es utilizada o producida por un proceso de desarrollo de software.

## Aspecto

Módulo software que no puede ser encapsulado en un procedimiento. Los aspectos no son unidades funcionales en las que se pueda dividir un sistema, sino propiedades que afectan a la ejecución o semántica de los componentes. Son conocidos también como intereses transversales.

## Elemento

Constituyente atómico de un modelo.

## Especificación

Descripción textual de la sintaxis y la semántica de un bloque de construcción específico; descripción declarativa de lo que algo es o hace.

## Estereotipo

Extensión del vocabulario de UML que permite crear nuevos bloques de construcción derivados a partir de los existentes, pero específicos a un problema concreto.

## Framework

En el desarrollo de software es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

## Gestión de Requisitos

Actividad para gestionar los cambios en los requisitos del sistema. La gestión implica el control de cambios y el impacto de los cambios.

#### **Heurística**

Capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como el arte y la ciencia del descubrimiento y de la invención o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente.

#### **Ingeniería de Requisitos**

Es un área de investigación que procura atacar un punto fundamental en el proceso, que es la definición de lo que se quiere producir.

#### **Intereses (*concerns*)**

Todo aquello que resulta importante para una aplicación (requisitos, infraestructura, código, etc.).

#### **Ingeniería de Software**

Rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas a los proyectos de desarrollo o mantenimiento de software de calidad.

#### **Notación**

Sistema de signos convencionales que se adoptan para expresar un conjunto de conceptos sobre el sistema de software por desarrollar.

#### **OMG Object Management Group**

Consorcio del cual forman parte las empresas más importantes que se dedican al desarrollo de software.

#### **Refinamiento**

Relación que representa una especificación más completa de algo que ya ha sido especificado a cierto nivel de detalle.

#### **Requisito**

Característica, propiedad o comportamiento deseado de un sistema.

#### **RUP Rational Unified Process**

Proceso Unificado de Rational, metodología del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo.

#### **Stakeholder**

Persona, grupo u organización que tenga directa o indirecta participación en una organización, ya que puede afectar o ser afectados por la organización de acciones, objetivos y políticas. Actores claves en una organización de negocios incluyen los acreedores, clientes, directores, empleados, gobierno (y sus organismos), los propietarios (accionistas), los proveedores, los sindicatos y la comunidad en la que se basa el negocio de sus recursos.

**UML *Unified Modeling Language***

Lenguaje Unificado de Modelado, notación estándar para el modelado de sistemas Software.

**Validación de los requisitos**

Proceso de confirmación, por parte de los usuarios o del cliente, de que los requisitos especificados son válidos, consistentes, completos, etc.

**Verificación de los requisitos**

Proceso de comprobación de que los requisitos realmente cubren las necesidades del cliente.

**Vista**

Proyección de un modelo, que se ve desde una perspectiva o un punto de vista dado, y que omite entidades que no son relevantes desde esa perspectiva.

**Workspace**

Es un directorio que representa el espacio de trabajo y el cual contendrá los proyectos que se crean en la herramienta RSA.