

Apuntes – Introducción a Rational Rose.

Introducción

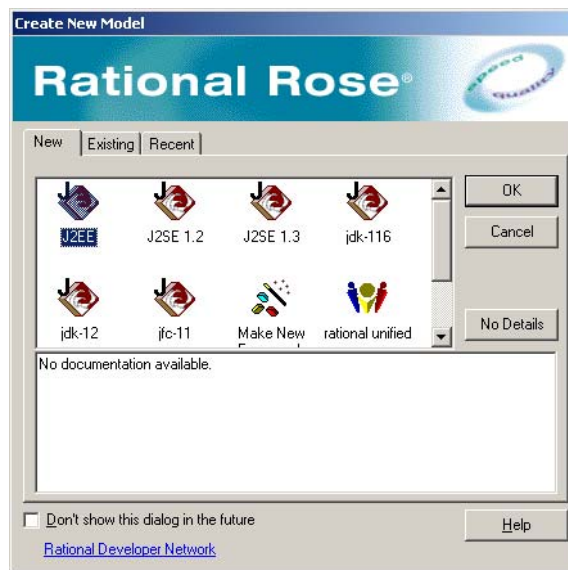
Rational Rose es una herramienta para “**modelado visual**”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software.

Rational Rose permite completar una gran parte de las **disciplinas** (flujos fundamentales) del proceso unificado de Rational (RUP), en concreto:

- Modelado del negocio
- Captura de requisitos (parcial)
- Análisis y diseño (completo)
- Implementación (como ayuda)
- Control de cambios y gestión de configuración (parte)

La ventana principal

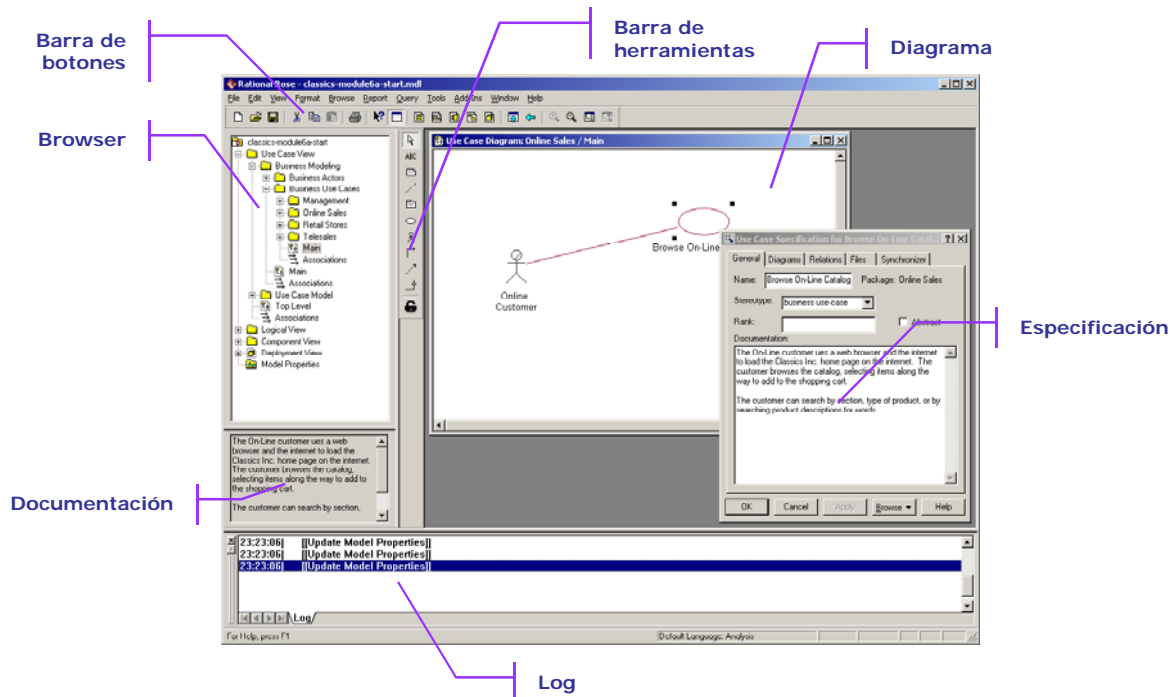
La primera pantalla que aparece es un **selector de “Framework”** (esqueleto). Este cuadro de diálogo permite elegir modelos que contienen elementos predefinidos para distintos tipos de proyectos. Existen esqueletos para Java, Visual C++, RUP, ...



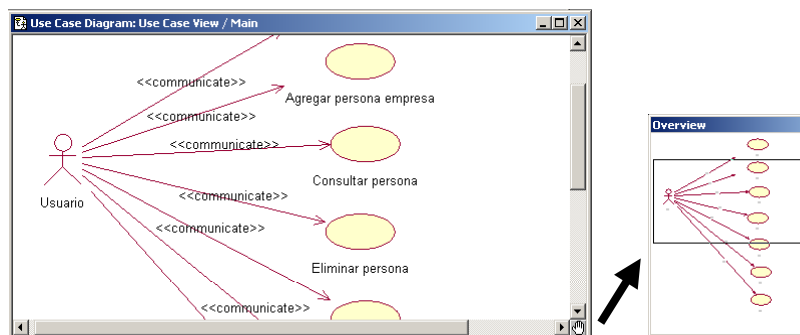
Al cancelar este cuadro de diálogo aparece la **ventana principal** de Rose.

Esta ventana tiene los siguientes componentes:

- “*Browser*”: muestra de forma jerárquica todos los elementos de los modelos de un proyecto.
- “*Documentación*”: muestra texto asociado al elemento seleccionado. Permite también modificar ese texto.
- “*Log*”: muestra mensajes sobre errores, progreso de tareas, etc.



- **Diagramas**: cada diagrama se muestra con una ventana diferente. Las ventanas de diagrama cuentan con un botón "overview", que permite desplazarse rápidamente por el contenido de diagramas grandes.



- **Barra de botones**: es independiente del diagrama activo. Contiene los botones típicos de cualquier aplicación Windows (nuevo, abrir, guardar, copiar, cortar, pegar, imprimir, ayuda contextual) y **botones propios de Rose**:



- Ver / ocultar documentación
- Ir a diagramas de clase, de interacción, de componentes, de estado, de despliegue, de caso de uso. Al activar estos botones se muestra una lista con los diagramas del tipo correspondiente, para seleccionar cuál se quiere visualizar.
- Ir al diagrama padre
- Ir al diagrama anterior
- Aumentar zoom, disminuir zoom
- Ajustar a ventana, deshacer ajustar
- Ayuda general

Los botones "ir a diagramas de caso de uso" y "ayuda general" no aparecen la primera vez que se arranca Rose, pero pueden activarse editando esa barra de botones.

- **Barra de herramientas:** dependerá del diagrama que se encuentre seleccionado. Contiene:
 - Icono para *seleccionar elementos*
 - Conjunto de *iconos propios* del tipo de diagrama
 - *Icono de bloqueo*. Si está bloqueado, cualquier icono de la barra permanece activo hasta que no se elija otro icono o el diagrama pierda el foco. Esto facilita insertar rápidamente elementos del mismo tipo en un diagrama. Este icono no se muestra generalmente, pero puede añadirse a la barra. Se puede conseguir el mismo resultado pulsando MAYS al insertar elementos en el diagrama.
- **Especificación:** permite completar la especificación de un elemento. La especificación puede abrirse de tres formas:
 - Pulsando con el botón derecho del ratón en un elemento y eligiendo la opción "*Open specification*".
 - Seleccionado el elemento y activando la opción "*Specification*", del menú "*Browse*".
 - Haciendo doble clic en el elemento. Esto funciona así sólo en algunos tipos de elementos.

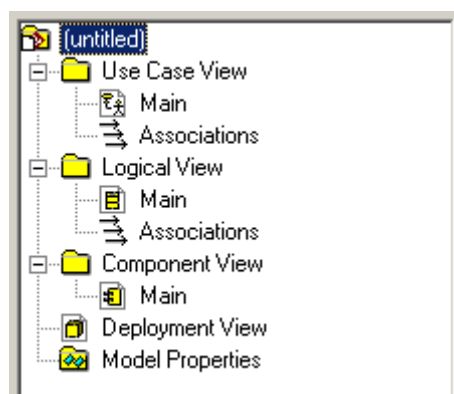


El "Browser"

Esta ventana **proporciona**:

- Una vista jerárquica de los elementos de un proyecto.
- Capacidades de "arrastrar y soltar" para editar el modelo.
- Actualización automática del modelo ante cambios en el árbol.

Esta ventana aparece por defecto en la parte izquierda de la ventana principal. Puede ocultarse si se desea y también puede cambiarse su posición (fija o flotante).



Uso del browser para **navegación**:

- Al hacer doble clic sobre un diagrama, se muestra la ventana de ese diagrama.
- Al hacer doble clic sobre un elemento de otro tipo (una clase, por ejemplo), se muestra la especificación de ese elemento.

- En un diagrama, al hacer clic con el botón derecho en un elemento y elegir la opción "select in browser", se selecciona ese elemento en el árbol.

Uso de **arrastrar y soltar** (introducción):

- De browser a browser: en general se mueven elementos de una parte a otra.
- De browser a diagrama: se insertan elementos en el diagrama. Si el elemento pertenece a un paquete distinto que el del diagrama se muestra un texto "from..." para indicar dónde está definido ese elemento.

La funcionalidad de "arrastrar y soltar" del browser es más compleja. En el manual de usuario está totalmente descrita (páginas 27, 28 y 29) y se irán viendo ejemplos según se profundice en las siguientes sesiones.

Creación de elementos. Hay varias opciones:

- Desde un diagrama, mediante el icono correspondiente de la barra de herramientas.
- Desde el browser, pulsando con el botón derecho en el paquete correspondiente, y eligiendo la opción "new" + el tipo de elemento que se desea crear. Esta opción no inserta el elemento en el diagrama.
- Desde el menú principal, seleccionando "tools / create / <<tipo de elemento>" y luego pinchando en el diagrama correspondiente.

Borrado: si se borra con [SUPR] un elemento de un diagrama no se está borrando ese elemento del modelo, sólo se está quitándolo del diagrama. Para borrar un elemento definitivamente hay que hacerlo:

- Eligiendo la opción "delete" del menú de contexto en el browser.
- Eligiendo la opción "edit / delete from model" ([CTRL+D]) del menú de contexto en el diagrama.

Introducción a los diagramas

Los diagramas son vistas gráficas del modelo. Rose mantiene automáticamente la consistencia entre los diagramas y las especificaciones correspondientes: si se modifica el diagrama se cambia la especificación, y viceversa.

Tipos de diagramas soportados por Rose:

- de Clases
- de Casos de uso
- de Colaboración
- de Secuencia
- de Componentes
- de Estados
- de Actividad
- de Despliegue

En el árbol de todo modelo aparecen cuatro **vistas**, cada una de las cuales admite determinados tipos de diagramas:

- *Caso de uso:* esta vista define la interacción entre actores y casos de uso. Diagramas principales: casos de uso, colaboración, secuencia, actividad.

- *Lógica*: esta vista define fundamentalmente las clases del sistema y sus relaciones. Diagramas principales: clases, estados.
- *Componentes*: esta vista contiene información sobre ficheros, ejecutables y librerías del sistema. Diagrama de componentes.
- *Despliegue*: esta vista muestra la asignación de procesos al hardware. Diagrama de despliegue.

Creación de diagramas:

- Desde el browser, mediante el menú de contexto de un paquete, opción "new / <tipo de diagrama>".
- Desde la barra de botones, mediante los botones "ir a <tipo de diagrama>", seleccionando "new" en el cuadro de diálogo posterior.
- Desde el menú principal, mediante la opción "browse / <tipo de diagrama>" y seleccionando "new" en el cuadro de diálogo.

Enlazado de diagramas. Se pueden enlazar diagramas mediante notas. Forma de hacerlo:

- En el diagrama origen del enlace, insertar una nota.
- Arrastrar el diagrama destino desde el browser hasta esa nota.
- Haciendo doble clic en la nota se irá al diagrama destino.

Borrado de diagramas:

- Desde el browser, mediante el menú de contexto del diagrama, opción "delete".
- Desde la barra de botones, mediante los botones "ir a <tipo de diagrama>", seleccionando el diagrama y luego pulsando el botón "delete" en el cuadro de diálogo posterior.
- Desde el menú principal, mediante la opción "browse / <tipo de diagrama>", seleccionando el diagrama y luego pulsando el botón "delete" en el cuadro de diálogo posterior.

Creación de elementos (ya visto):

- Desde el browser.
- Desde el diagrama (barra de herramientas).
- Desde el menú principal.

Nombrado de elementos:

- Elementos que pueden tener nombre repetido (si están en paquetes distintos): actores, casos de uso, clases, componentes y paquetes. En caso de producirse un nombre repetido, saldrá un mensaje avisándolo.
- Si el nombre se define desde el diagrama, no existe la posibilidad de repetir nombres: se cogerá el elemento cuyo nombre se ha escrito. Si se quieren repetir nombres, habrá que poner el nombre en el browser o en la especificación.

Edición de diagramas:

- Funciona como cualquier editor gráfico.
- Los elementos se pueden mover, cambiar el tamaño, copiar, pegar, ... con el ratón.

Borrado de elementos (ya visto):

- Borrado “superficial” (sólo se borra del diagrama).
- Borrado “profundo” (se borra del modelo).

Relaciones:

- Se crean eligiendo el tipo de relación y arrastrando desde el origen hasta el destino.
- Se pueden definir “codos” en la línea que representa la relación (al crearlos o posteriormente).

Organización automática: Rose puede reorganizar de forma automática un diagrama. Para ello se usa la opción “Format / Layout diagrama” del menú principal.

Workspaces: define un entorno de trabajo, representando qué ventanas de un modelo están abiertas. Pueden grabarse y cargarse y pueden existir varios workspaces asociados a un mismo modelo.

Requisitos y Análisis con Rose

1. Captura de Requisitos

- **Estructura** recomendada en la fase de captura de requisitos para un proyecto de la complejidad del que se pide:
 - Use Case View
 - Paquete "Casos de uso"
 - Casos de uso y actores directamente, sin paquetes intermedios.
 - Dentro de cada caso de uso irá su modelo de estados, con su correspondiente diagrama de transición de estados.
 - Diagrama de casos de uso "Main": mostrará el modelo de casos de uso.
 - Paquete "Interfaz de usuario" para el prototipo de IU
 - Diagrama "Main" (diagrama de clases) mostrará la navegación entre ventanas.
 - Las pantallas (y resúmenes de datos) se representarán como clases.
 - Diagrama "Main": mostrará los dos paquetes.

1.1 Encontrar actores y casos de uso

- En la barra de herramientas de caso de uso incorporar el botón "*Creates an association relation*", que permite crear asociaciones sin dirección.

1.1.1 Identificación de actores

- Los actores se pueden crear directamente en el diagrama.
- Descripción de cada actor: en el campo "*documentation*" de su ventana de especificación.

1.1.2 Identificación y breve descripción de casos de uso

- Los casos de uso se pueden crear directamente en el diagrama.
- Descripción breve de cada caso de uso: en el campo "*documentation*" de su ventana de especificación.

1.1.3 Descripción del Modelo de casos de uso

- El resultado se va reflejando en el diagrama "Main" del paquete "Casos de uso".

1.2 Detallar los casos de uso

- Aquí van diagramas de estados y especificación de los casos.

1.2.1 Diseñar un diagrama de estado para cada caso

- Diagrama de estados de cada caso de uso: se crea dentro del caso de uso.
- Para "pintar" el camino básico sólo se pueden usar colores diferentes.

1.2.2 Descripción textual del caso de uso

- Lo mejor es hacerla en documentos aparte (Word, workpad, HTML, ...) y enlazarla al caso de uso en la pestaña "files" de su especificación.

1.3 Diseñar un prototipo de la interfaz de usuario

- Como Rose no permite pintar flechas entre notas, se utilizarán clases para representar las pantallas.
 - Las **pantallas** llevarán el estereotipo "pantalla"
 - La **navegación** se reflejará mediante asociaciones con dirección.
 - Los **datos** que aparecen en pantalla se reflejarán como atributos públicos de la clase, con los estereotipos "entrada" o "salida" para reflejar su función.
 - Las **acciones** que puede elegir el usuario se reflejarán como operaciones públicas de la clase, con el estereotipo "acción".
 - Los estereotipos se pueden escribir en el momento, o se pueden crear de forma completa, modificando ficheros de configuración de Rose. Esta segunda opción permite reutilizar estereotipos entre proyectos, así como asignarles representaciones gráficas.

2. Análisis

- **Estructura** recomendada en la fase de análisis para un proyecto de la complejidad del que se pide:
 - Logical View
 - Paquete "Modelo de Análisis"
 - Un paquete "Clases de análisis" que contendrá las clases de análisis (así se podrán utilizar en varios casos de uso).
 - Este paquete tendrá las clases de análisis
 - Habrá un diagrama de clases de análisis ("Main").
 - Un paquete para la realización de cada caso de uso, con nombre "<caso de uso> - Análisis".
 - Un diagrama de clases de análisis ("Clases") con aquellas clases que participan en el caso de uso (incluyendo su relación con los actores).
 - Un elemento de tipo "Realización de caso de uso" (caso de uso con estereotipo "use-case realization") con el mismo nombre que el paquete ("<caso de uso> - Análisis").
 - Dentro de cada realización de caso de uso se define un diagrama de colaboración
 - Un diagrama de casos de uso ("Realizaciones – Análisis") donde se muestran las relaciones de realización entre los casos de uso de captura de requisitos y los casos de uso de análisis.
 - Paquete "Modelo de Diseño" (su contenido se verá cuando se trate el tema de diseño, implementación, etc.).
 - Diagrama "Main": mostrará los dos paquetes de la vista lógica.

2.1 Analizar los casos de uso

- Para mayor comodidad, poner los botones de los tres tipos de clases de análisis en la barra de herramientas de diagrama de clase.

- En los diagramas de clases de análisis se usarán asociaciones sin dirección. Poner el botón correspondiente en la barra de herramientas de clases de análisis.

2.1.1 Identificar las clases de análisis

- Las clases de análisis se crearán directamente en el diagrama "Main" del paquete "Clases de Análisis". Luego se arrastrarán desde el browser a los diagramas de clases de la realización de cada caso de uso.
- Los actores se arrastran desde el browser.
- Las relaciones se definen en el diagrama de clases global. Luego, para los diagramas parciales basta con arrastrar las clases y actores y se dibujan automáticamente las relaciones.

2.1.2 Describir interacciones entre objetos de análisis

- Relaciones de realización:
 - Para cada realización de caso de uso se reflejará la relación entre un caso de uso y su realización como una asociación dirigida (de realización a caso de uso) con estereotipo "*realize*" (En el diagrama "Realizaciones - Análisis").
- Diagramas de colaboración:
 - En los diagramas de colaboración los objetos (de clases de análisis o de actores) se crean arrastrando las clases sobre el diagrama.
 - Hay que dibujar los enlaces entre objetos (no se "heredan" de las relaciones entre clases).
 - Conviene recordar el sentido en el que se pintaron los enlaces, ya que afecta a la inserción de mensajes.
 - Los mensajes se numeran de forma automática. No se pueden insertar mensajes "en medio" de la secuencia actual, por lo que conviene preparar un borrador en papel antes de meter el diagrama de colaboración en Rose. **Ojo:** Se puede cambiar el orden de los mensajes mediante el "Diagrama de secuencia asociado" (F5). En ese diagrama se puede pintar la dependencia entre mensajes (para numeración jerárquica) y se puede cambiar el orden.
 - Recordar que para borrar realmente un mensaje hay que eliminarlo del modelo (CTRL+D).
 - Para añadir más mensajes en el mismo sentido, lo mejor es editar el enlace (pestaña "Messages").
 - Los nombres de los mensajes se van a definir como operaciones (sin argumentos) de las clases de análisis (facilita la consistencia y reutilización). **Ojo:** hay que seleccionar el texto del mensaje y no la flecha. **Ojo:** hay que hacerlo sobre la marcha, ya que es más complicado cuando hay varios mensajes en la misma flecha.
 - Esto hará que las operaciones aparezcan en los diagramas de clases. Se puede evitar que aparezcan (en los diagramas parciales) editando las propiedades de cada clase en los diagramas.
- Descripción textual de las colaboraciones
 - Se pueden poner como una nota en el diagrama de colaboración.

- Otra opción sería ponerlo como descripción de la realización del caso de uso (menos recomendable porque es difícil hacer referencia a los mensajes del diagrama).

2.2 Analizar cada clase de análisis

- Los resultados de esta etapa se verán reflejados en el diagrama "Main" del paquete "Clases de análisis".
- En este diagrama se puede cambiar la forma de representar los estereotipos para que se vean mejor las responsabilidades y atributos (opción "*decoration*").

2.2.1 Identificar las responsabilidades

- Son las operaciones identificadas según se van completando los diagramas de colaboración de cada caso de uso.

2.2.2 Identificar los atributos

- Se definen en cada clase sin asignarles tipo.
- Esto hará que los atributos aparezcan en los diagramas de clases. Se puede evitar que aparezcan en los diagramas parciales editando las propiedades de cada clase en los diagramas.

2.2.3 Identificar asociaciones, agregaciones y generalizaciones

- Se definen en el diagrama de clases principal de todo el análisis (Diagrama "Main" del paquete "Clases de Análisis").
- Para definir agregación, se modifica el extremo de la asociación que corresponda, diciendo que es agregado.
- Para definir agregación fuerte, se modifica el tipo de "*containment*" del extremo opuesto a la agregación.
- Para definir las cardinalidades, se modifica el extremo que corresponda.