

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

Primeros pasos con Enterprise Architect y UML 2.x

0. Índice de contenidos.

- [1. Introducción.](#)
- [2. Entorno.](#)
- [3. Instalación y arranque.](#)
- [4. Elección de modelos.](#)
- [5. Diagramas de casos de uso.](#)
- [6. Matrices de trazabilidad requisitos-casos de uso.](#)
- [7. Diagramas de actividad.](#)
- [8. Diagramas de secuencia.](#)
- [9. Diagramas de comunicación.](#)
- [10. Diagramas de estado.](#)
- [11. Conclusiones.](#)

1. Introducción

Siempre que me toca dar un curso de UML o técnicas de análisis y diseño, dedico un rato a revisar las herramientas y ponerme un poco al día ... todo cambia.

En este caso voy a probar la versión de evaluación de Enterprise Architect 7.5 y aprovechar para ver alguna peculiaridad como podría ser su proposición de modelos, las matrices de trazabilidad y la edición tabular de diagramas de estados.

Como es costumbre, voy a capturar unas pantallas y compartirlas con mis compañeros de Autentia y con todos vosotros aunque no pretende ser completo ni exhaustivo aunque si daré unas pinceladas de cómo me gusta usarlos.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Asus G50Vseries (Core Duo P8600 2.4GHz, 4GB RAM,320 GB HD).
- Sistema operativo: Windows Vista Ultimate.
- Mozilla Firefox 2.0.0.20

3. Instalación y arranque.

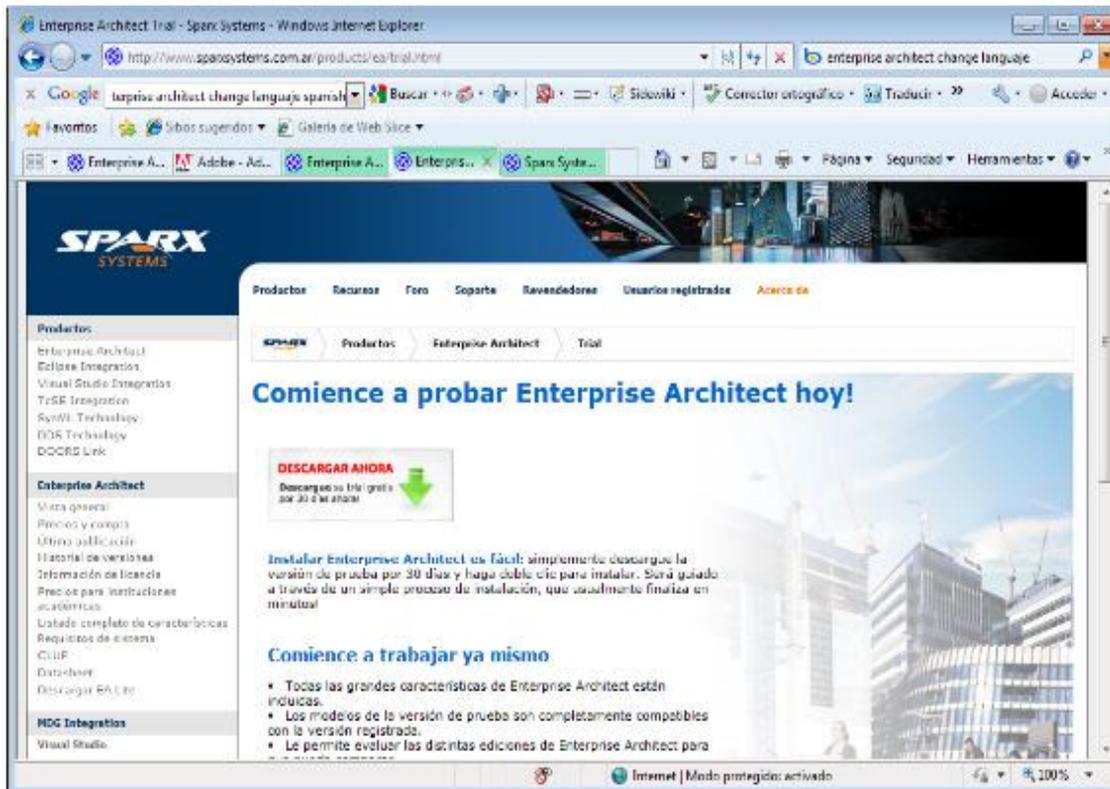
Lo primero que haremos por tanto es bajarnos la herramienta: Nos vamos al site <http://www.sparxsystems.es/New/products/ea.html>



Recordad que a lo mejor yo escribo el tutorial ahora (Enero de 2009) y tú lo lees dentro de años por lo que es posible que las cosas no sean exactamente iguales: pues nada, tampoco estará tan lejos.

Con pinchar evaluar ya tenemos automáticamente el fichero para la descarga.

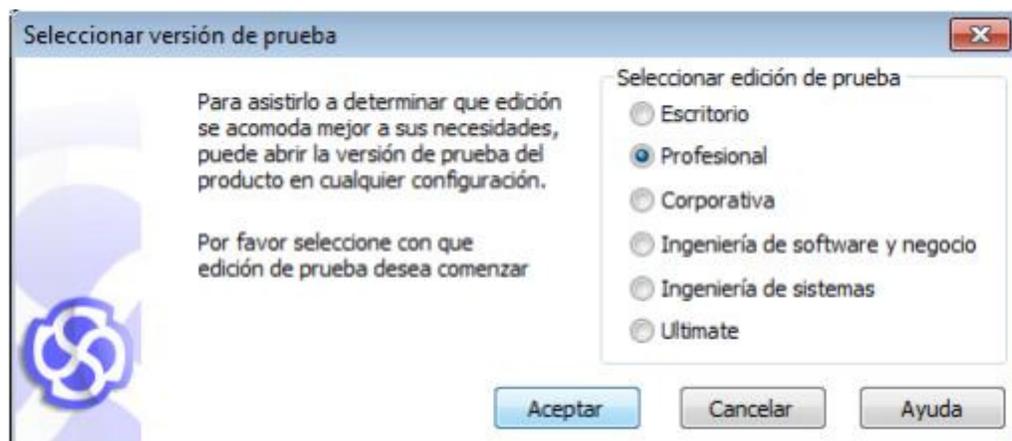
Desde la página en argentina te la puedes descargar en castellano pero francamente pienso que si trabajas con varias herramientas UML distintas utilices la versión en Inglés porque te puedes volver loco con las traducciones que aporta cada herramienta (aunque de esta me quejo poco).



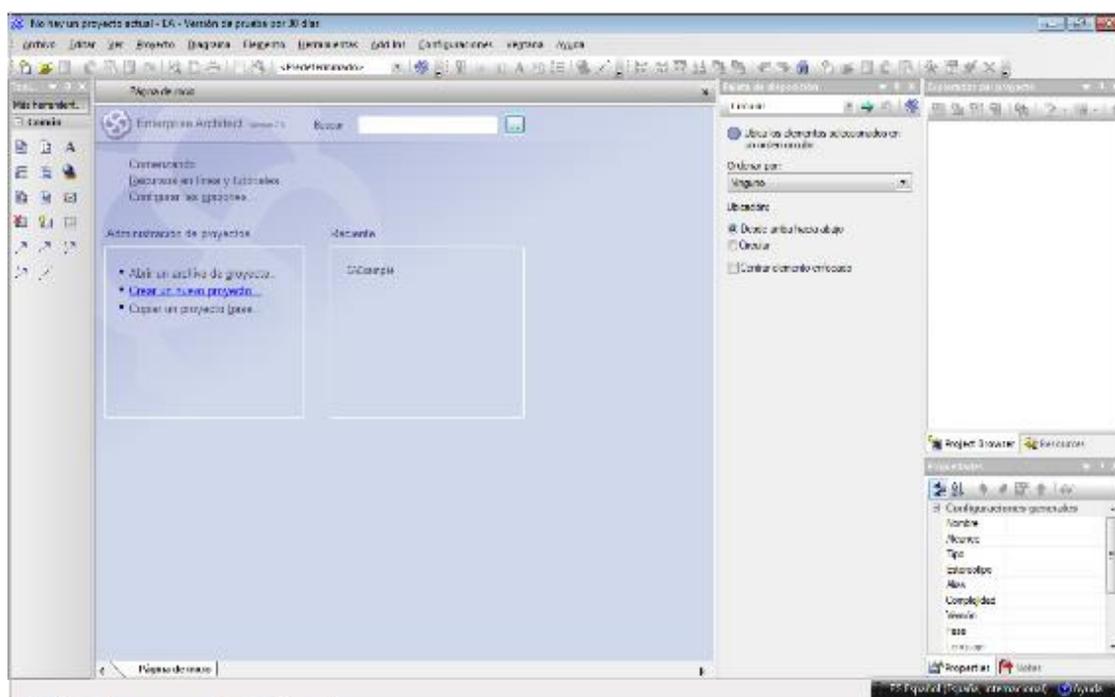
Nos lo bajamos e instalamos en castellano.



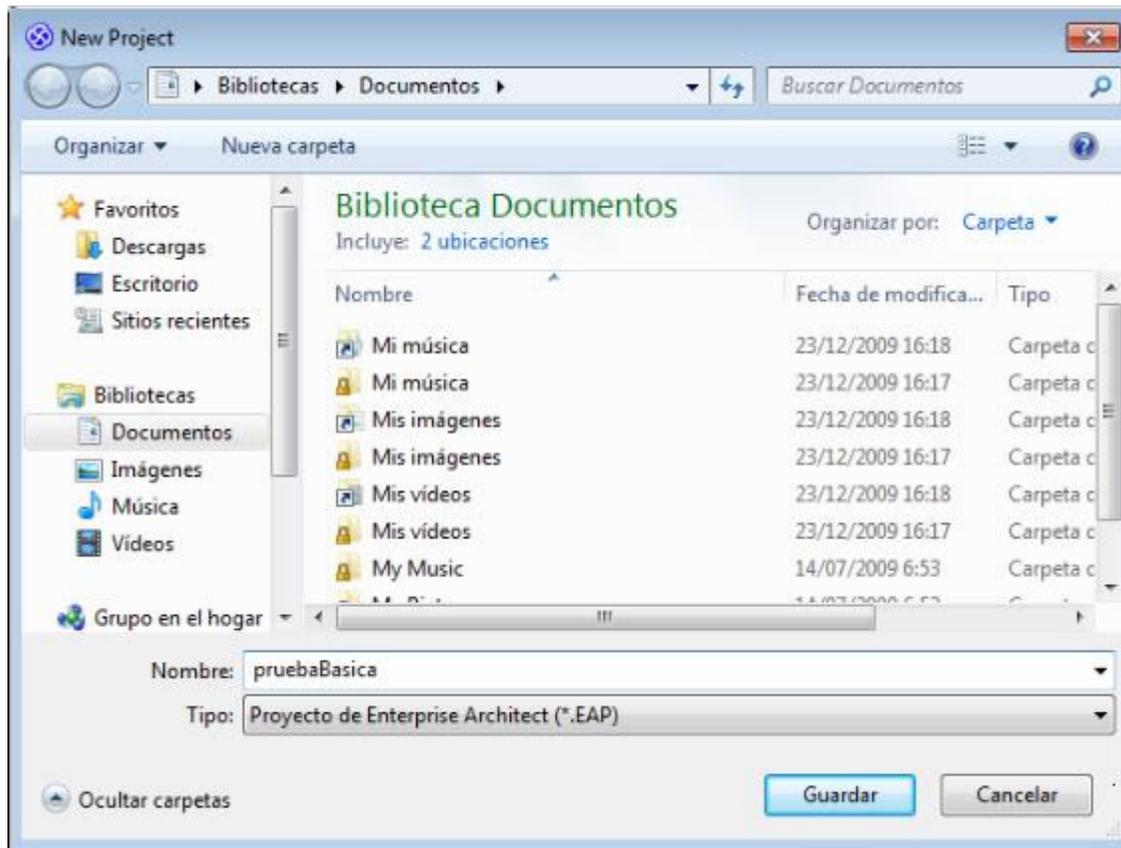
Como estamos en evaluación, podemos elegir simular la versión de Enterprise Architect que queremos utilizar ... todo un detalle. Yo elijo la profesional.



El aspecto es inmejorable aunque choca el orden respecto a otras herramientas (aunque siempre suele ser configurable): Aparece el menú de navegación a la derecha y los artilugios a utilizar en la izquierda.



Lo primero que hacemos es crear un proyecto. Vemos como queda en el reciente Windows 7.



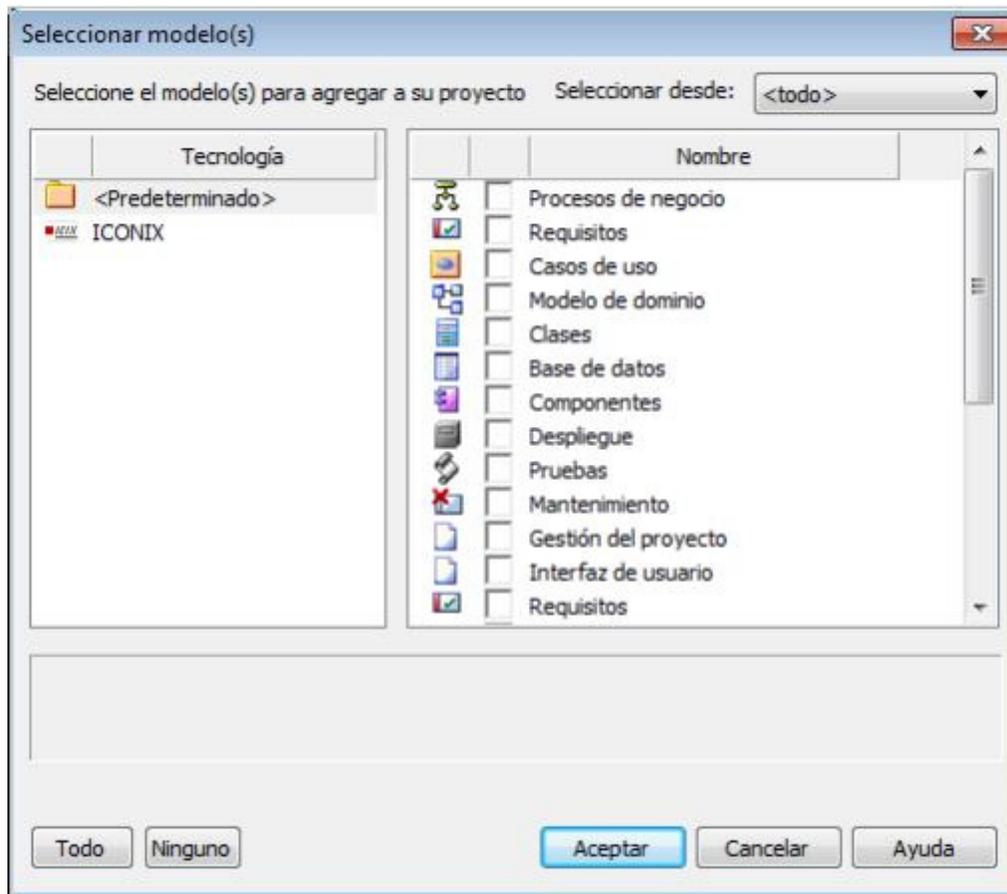
4. Elección de modelos.

Nuestro trabajo podría ser muy distinto:

- Modelar los procesos de negocio.
- Definir los requisitos de un proyecto.
- Diseñar una solución y construir el esqueleto.
- Codificar utilizando técnicas ágiles y desarrollo guiado por pruebas.
- Etc.

Enterprise Architect nos permite elegir los modelos que queremos utilizar. Fijaros en la opción “seleccionar desde” porque nos filtrará entre los modelos propios y otras opciones (como lo particulares del proceso unificado de desarrollo de software). Una vez elegidos nos crea una estructura con elementos de ejemplo que nos pueden guiar muy bien en donde poner cada cosa (que para un principiante es uno de los mayores de los problemas).

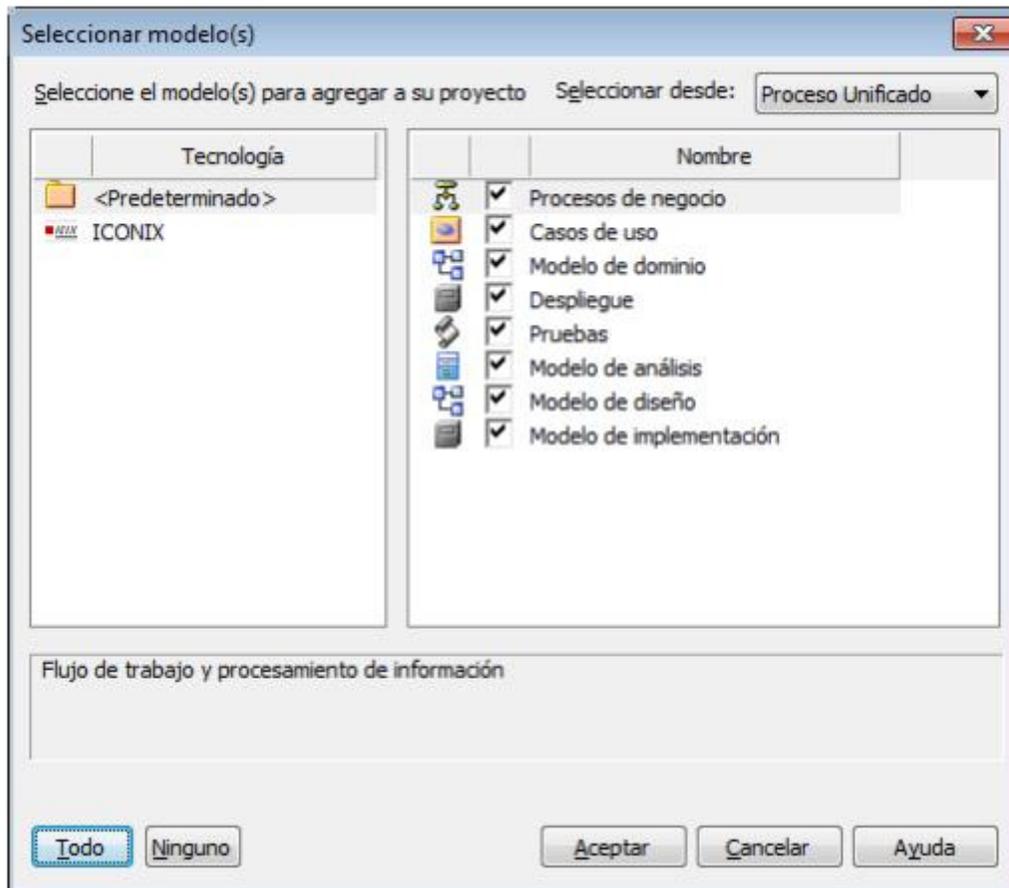
Vemos todos los modelos



Filtramos por UP. Recordemos que el proceso unificado es:

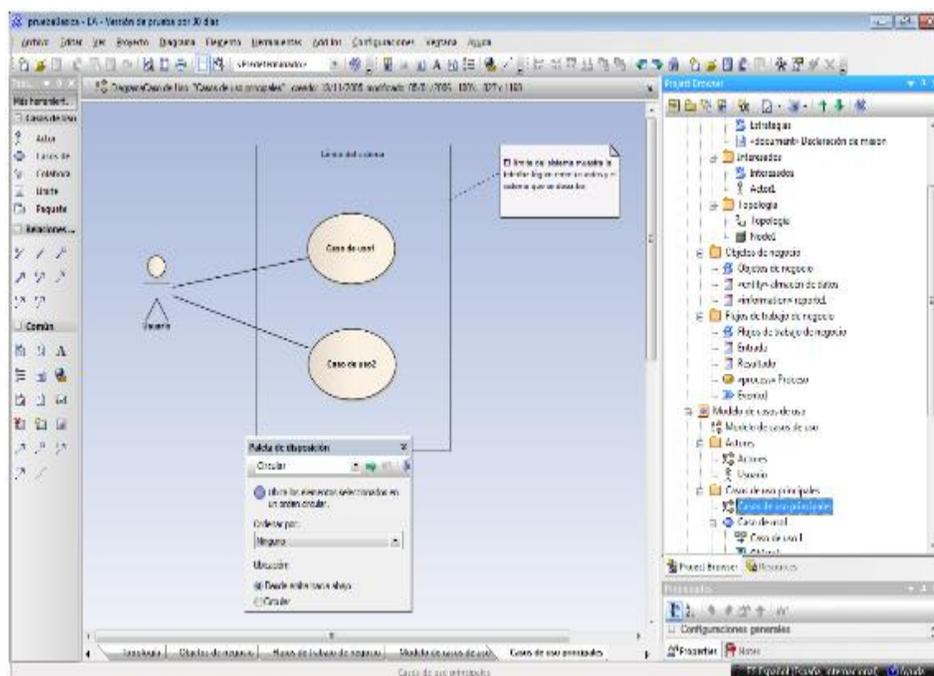
- Guiado por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

Esto nos da ya la idea de por donde empezar a modelar los diagramas: POR ENTENDER EL COMO SE GANA DINERO EN LOS PROCESOS DEL NEGOCIO (era para despistar).



Creamos un proyecto basado en el proceso unificado eligiendo todos los modelos inicialmente. La verdad es que para alguien que no trabaje habitualmente con estas herramientas o que no posea un esqueleto propio para organizar los elementos te va a crear una estructura base excelente, poco confusa y hasta con un pequeño ejemplito de cada diagrama UML o artilugio.

El programa es básicamente como todos los demás: Si pinchas sobre cualquier elemento aparece una flecha inteligente que te permite crear los elementos posibles a asociar.

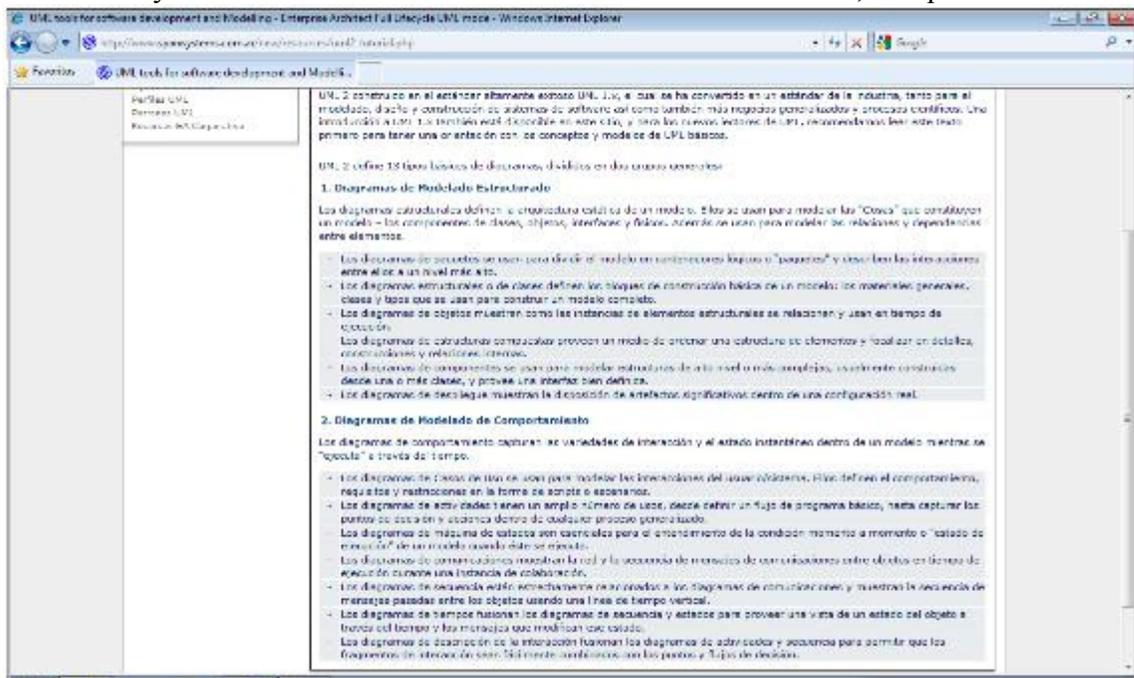


Creamos nuestra jerarquía de actores y casos de uso.

Para cacharrear con las opciones podemos crear un proyecto nuevo donde ir probando los distintos modelos de un modo aislado.

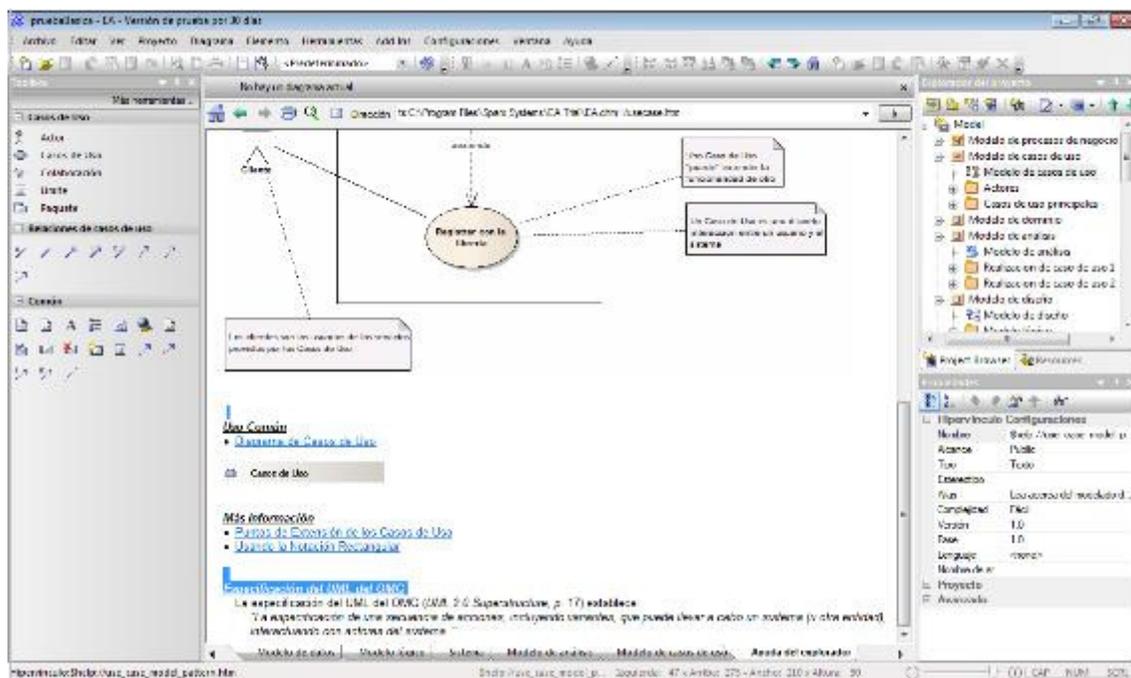
Una de las cosas más interesantes que debemos hacer es pasarnos por la documentación online: http://www.sparxsystems.com.ar/new/resources/uml2_tutorial.php (si en castellano no encontráis contenido, quitar el punto .ar y probar buscar en inglés).

La ayuda es de las mejores que encontrareis en este tipo de herramientas. De hecho hay veces que estoy dando un curso de UML y directamente re direcciono a este Web a mis alumnos, aunque estemos usando otra



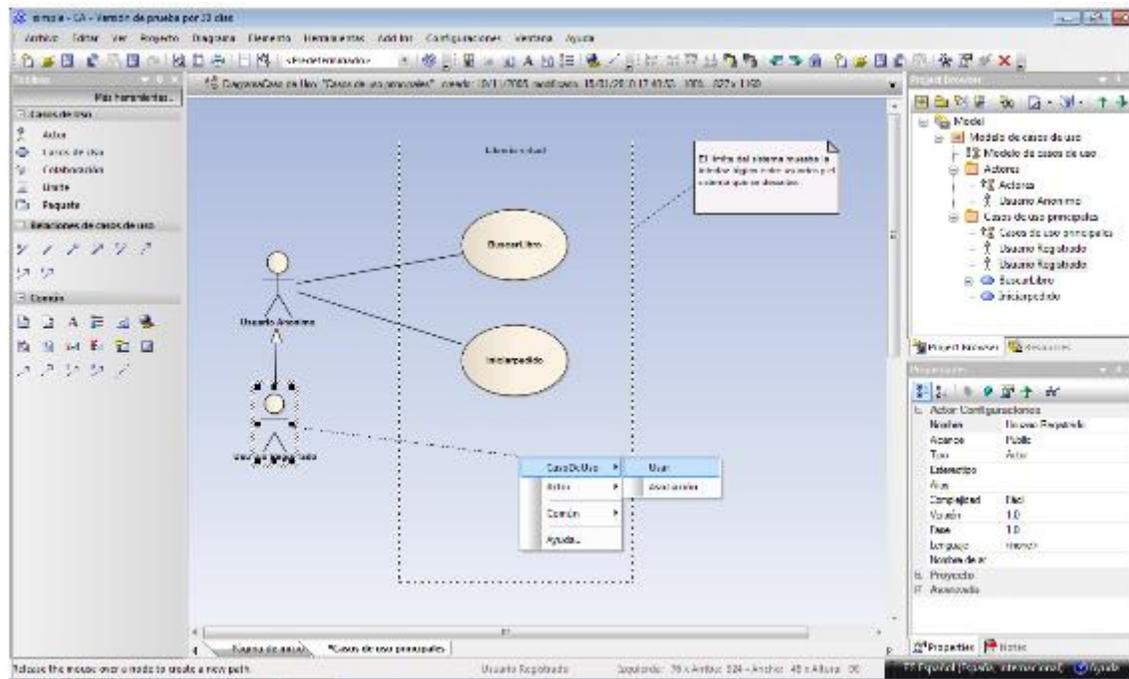
herramienta.

Además tiene un detallazo ... cuando navegas en la propia herramienta en su ayuda, te referencia a lo que dice la especificación (por cierto, suele caer todo en la misma)

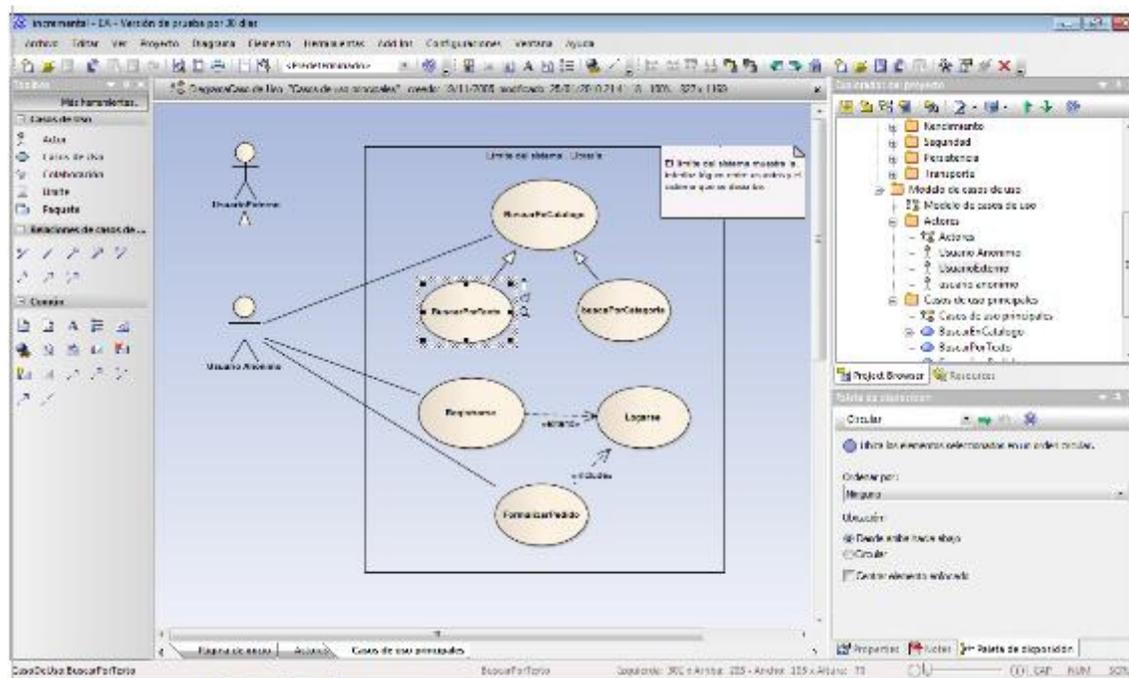


5. Diagramas de casos de uso.

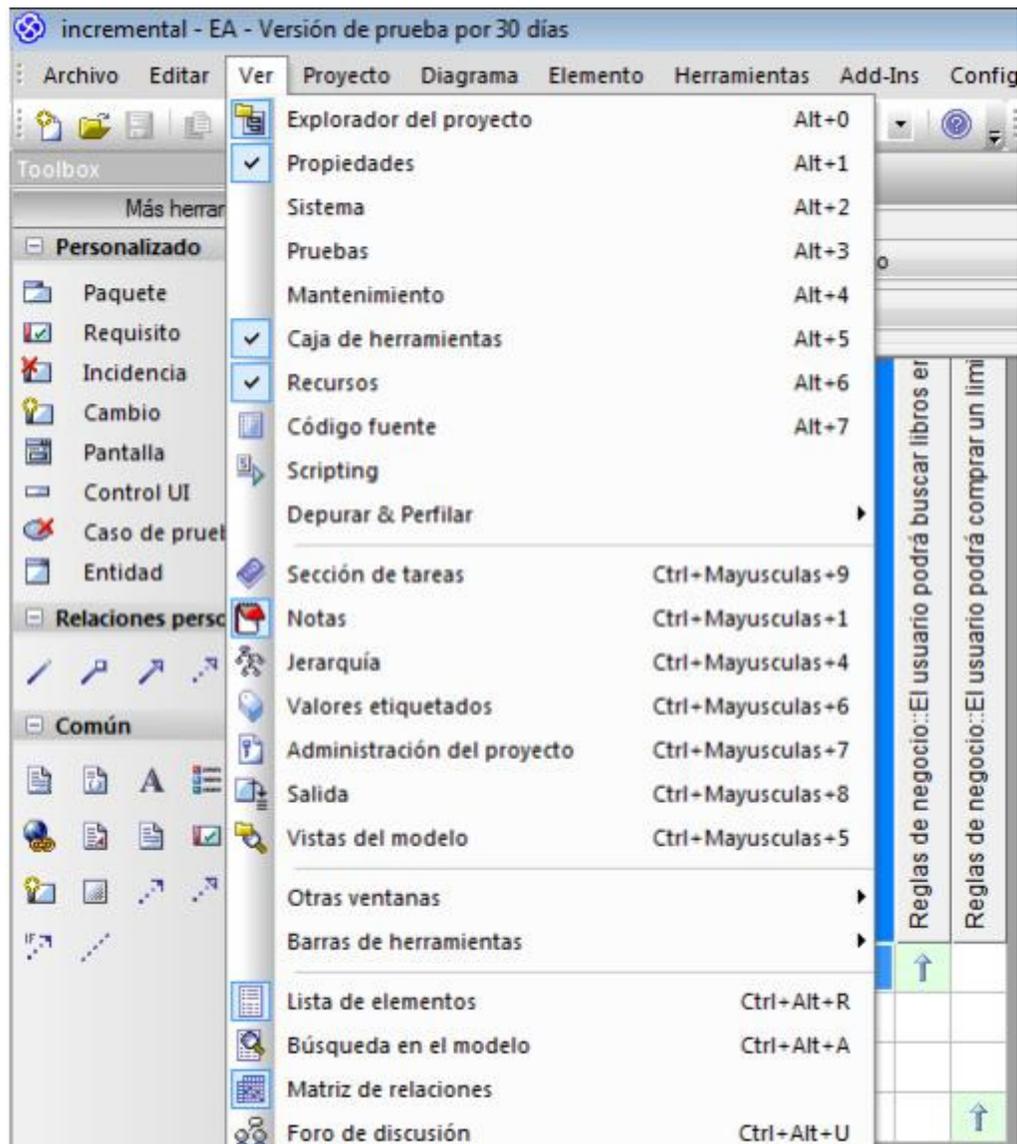
Vemos el aspecto de modelo de casos de uso (árbol de la derecha). Recordad que si lo hacéis así (eligiendo solo un modelo y colocando los elementos donde mejor os parece) y luego vais añadiendo otros modelos, posteriormente tendréis que mover cada elemento al sitio adecuado (si no queréis volveros locos).



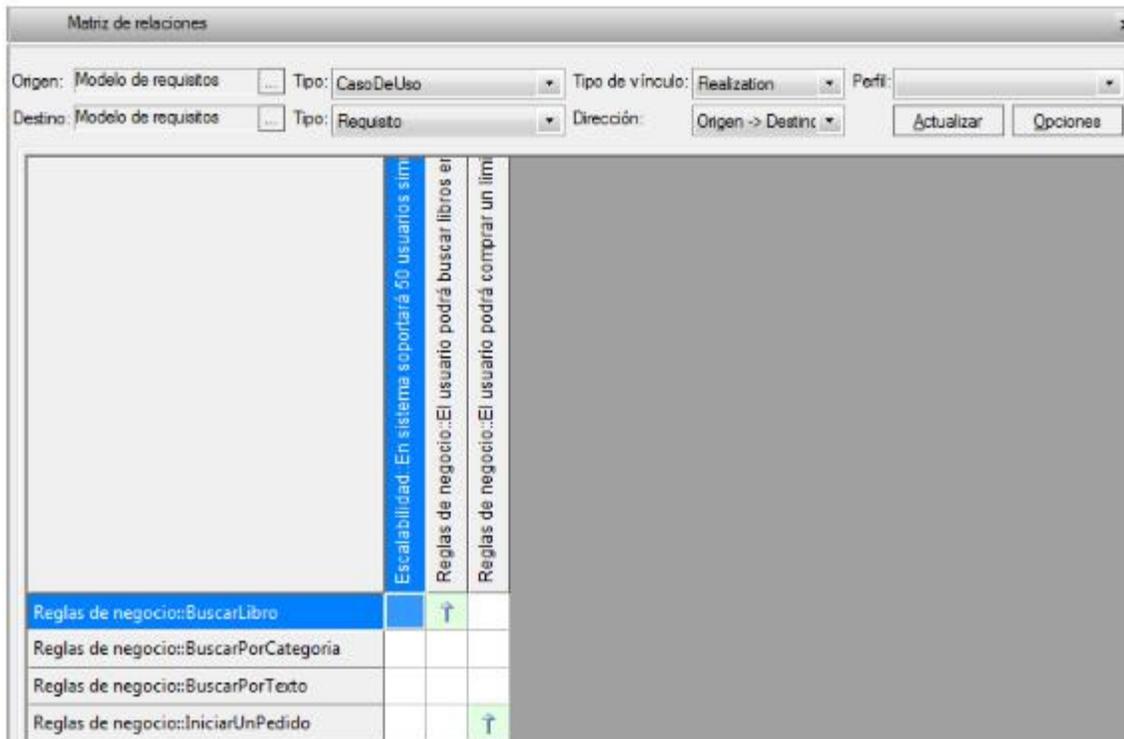
Todo es bastante intuitivo



6. Matrices de trazabilidad requisitos-casos de uso.



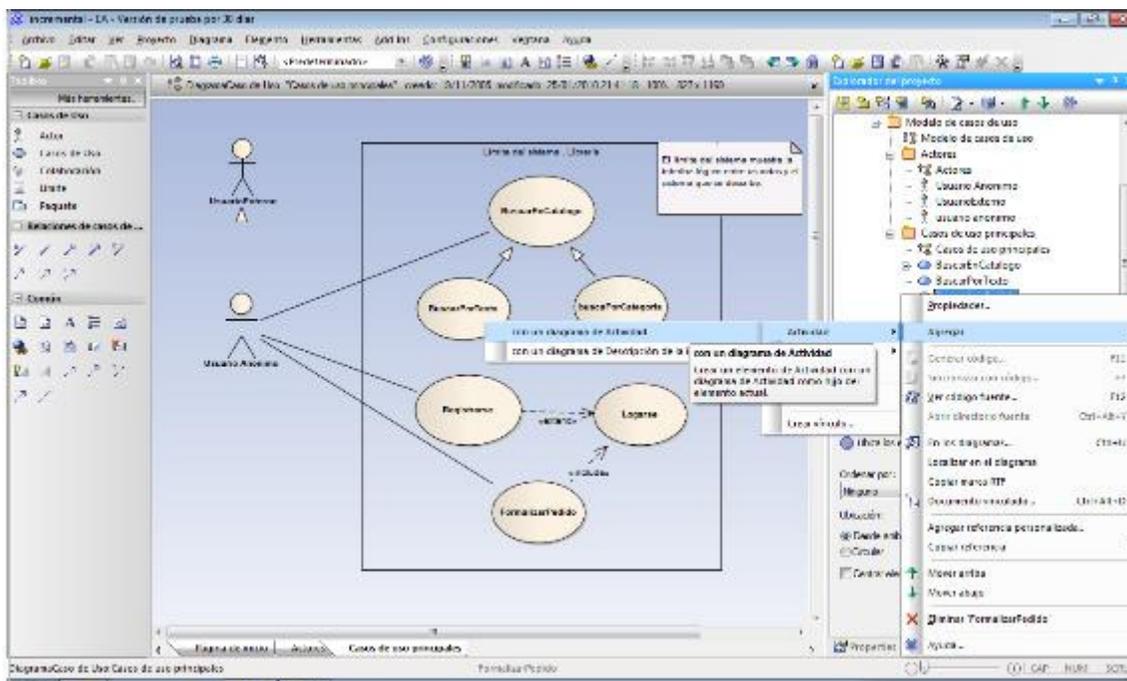
Es sencillo y me recuerda a las matrices de otras herramientas como Requisit Pro.



Como en casi cualquier herramienta, podemos detallar un diagrama con otros.

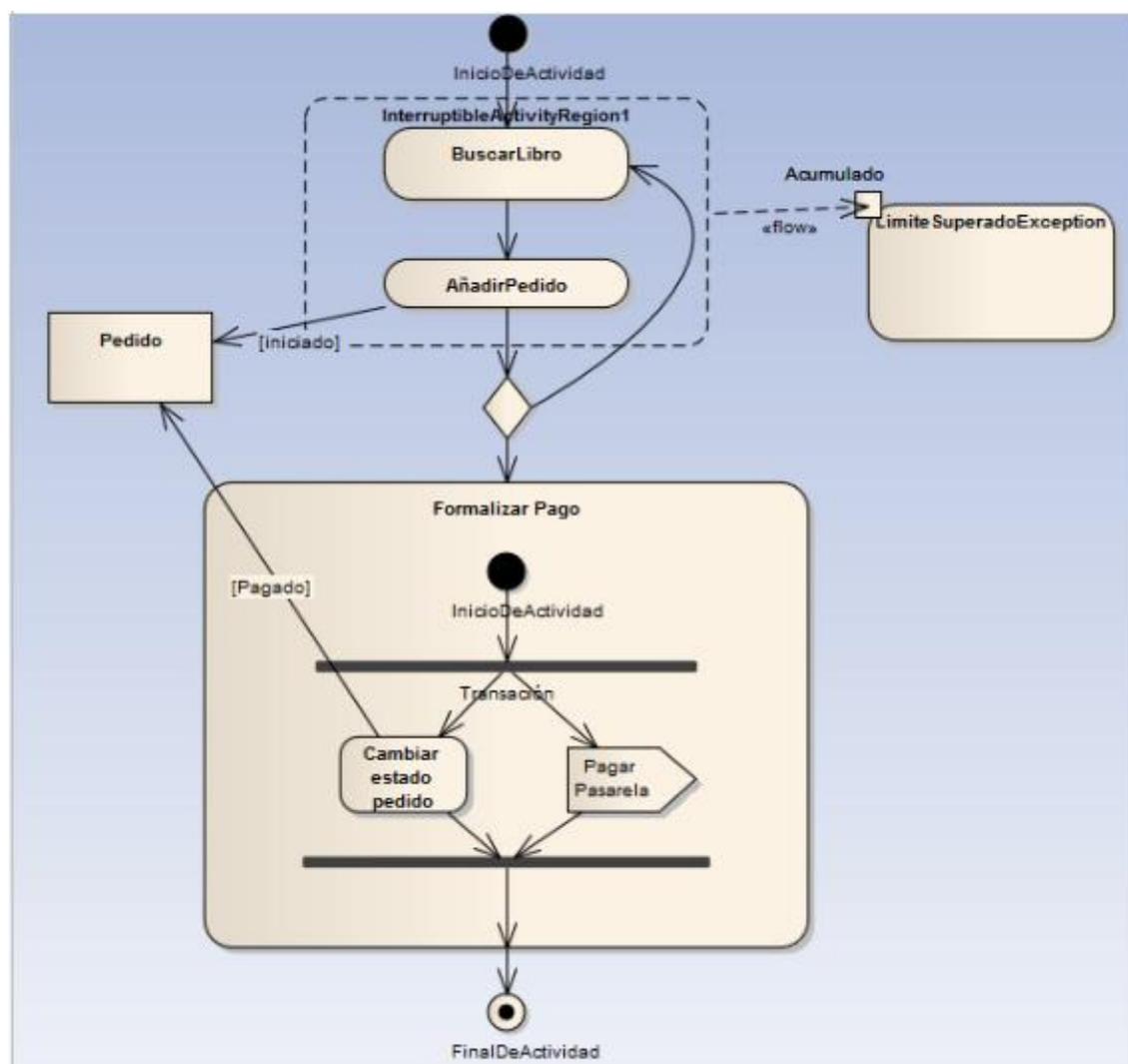
7. Diagramas de actividad.

Añadimos un diagrama que ha cambiado un poquito más de la primera versión de UML a la 2.x, el de actividad que utilizamos para explicar un caso de uso. Pulsando el botón derecho sobre el caso de uso ...



Recordad que una buena práctica podría ser limitar el número de elementos a pintar en un diagrama. También que representamos una abstracción del mundo, no un mundo a distintos niveles de zoom: No hay que

representar todo en el mismo esquema sino solamente lo significativos. Es más, la mayoría de las veces no existe ni siquiera consenso entre usuarios de UML sobre como representar distintas cosas por lo que siempre es aconsejable abusar de las notas o aclaraciones textuales adicionales.



8. Diagramas de secuencia.

Un caso de uso (o casi mejor dicho una colaboración concreta) se puede representar también como un diagrama de secuencia.

A alto nivel y tirando de los patrones de GRASP y de las clases de análisis, podemos reducir un sistema a interfaces, controladores (me gusta más la palabra gestores) y entidades.

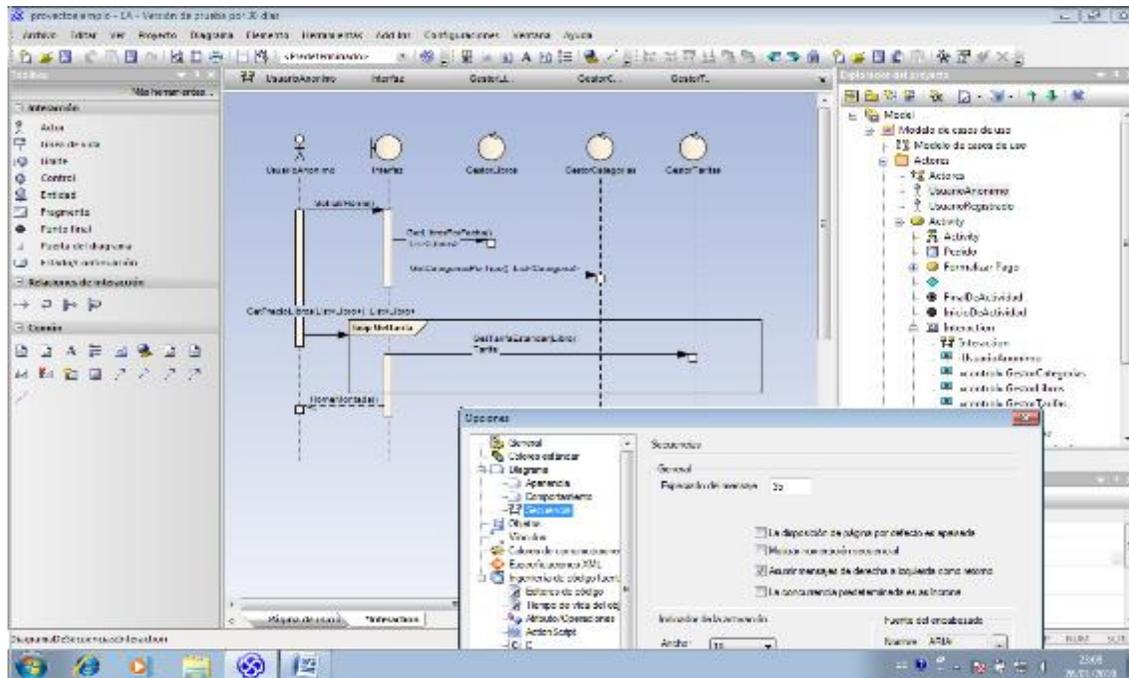
En Enterprise Architect quedan realmente bonitos. Ahora entramos con la guerra de cómo pintan/ocultan los mensajes de retorno.

Para mi gusto, no se pintan los mensajes de retorno en diagramas de secuencia a menos estemos a muy alto nivel (a nivel de diagrama de secuencia de sistema) o que se quiera resaltar algún valor concreto (ya que la estructura de un mensaje es auto-explicativa). Los valores de retorno también se pueden utilizar para indicar salidas excepcionales.

En las opciones de Enterprise Architect se pueden indicar que los mensajes de derecha a izquierda son de

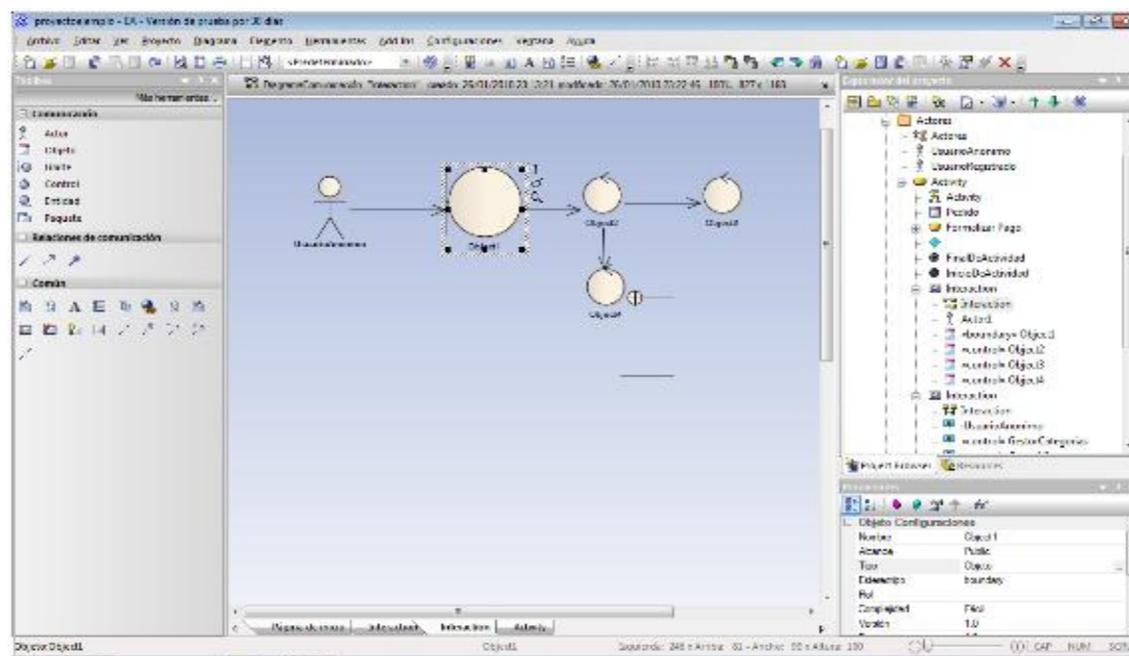
retorno.

En UML 2 se puede mostrar bloques alternativos, bucles, etc. dentro de un diagrama de secuencia. Enterprise Architect lo hace muy bien pero francamente es una práctica que no me acaba de gustar demasiado. Veo más un diagrama de secuencia como una colaboración lineal que como un árbol donde representar varias colaboraciones (aunque supongo que es cuestión de gustos).



9. Diagramas de comunicación.

Los diagramas de colaboración evolucionaron a diagramas de comunicación donde fundamentalmente es una misma vista de una secuencia pero con el objetivo de ver el nivel de acoplamiento. Yo los uso poco.

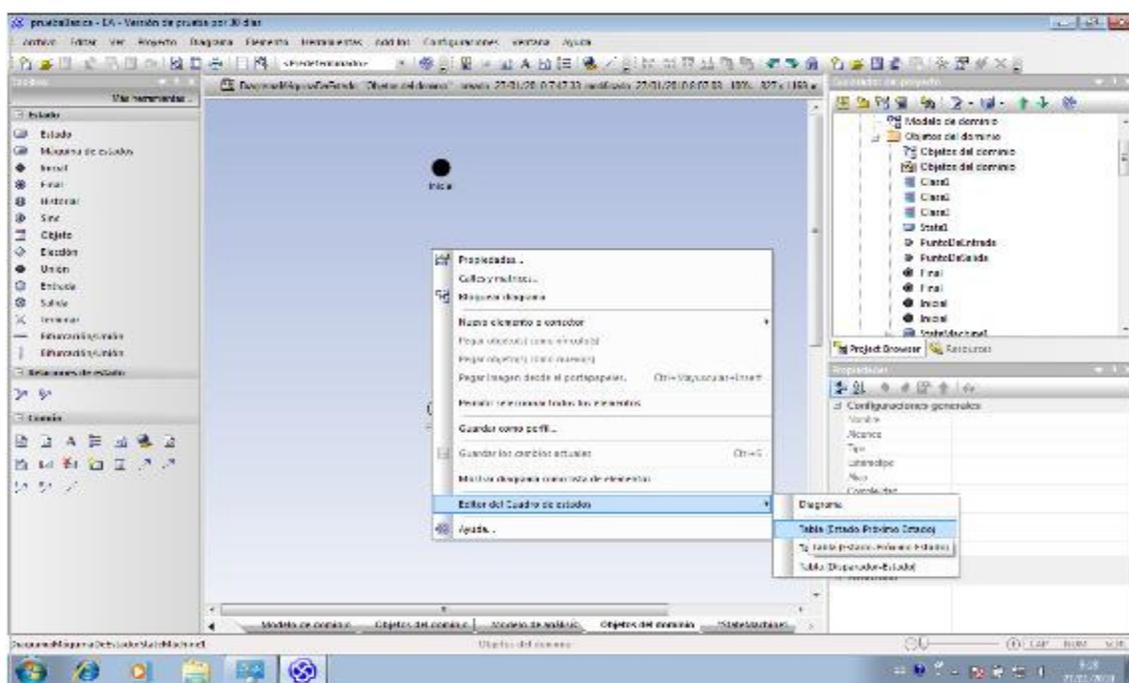


Echo de menos tener a mano la opción de convertir diagramas de secuencia en comunicación y al revés. Seguro que está (incluso he leído que hay una extensión para ello).

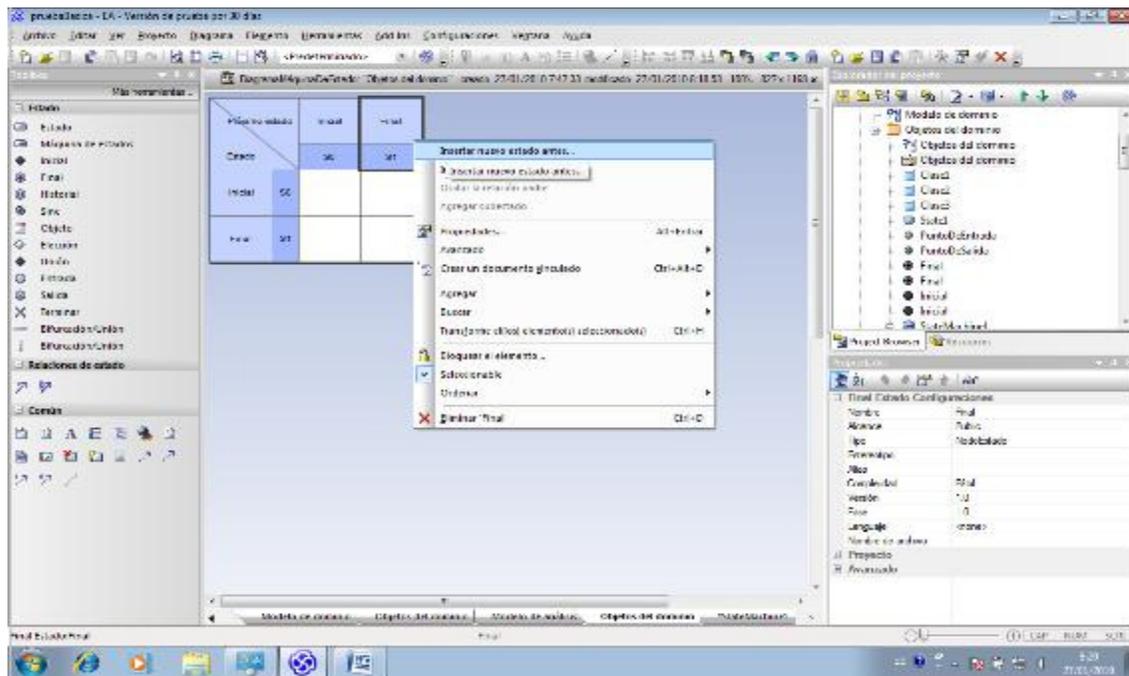
9. Diagramas de estado.

Unos de los diagramas más importantes, para mi gusto, en UML son los diagramas de estados. Recordad que para que algo cambie de estado o bien debe haber una pantalla o bien un proceso (batch) que provoque la transición. ¿Cómo podemos dimensionar un sistema si no somos capaces de acotar los estados de las entidades?

Tiene una herramienta en forma de tabla para permitirnos ver y editar las transiciones. Empezamos del modo clásico pintando un diagrama con su punto de inicio y final (no es obligatorio hacerlo así). Y conmutamos al editor tabular.



Podemos añadir estados y transiciones



Vemos la ventana de detalle.

Estado : State3

General | Requerimientos | Restricciones | Vínculos | Escenario | Ar

Nombre:

Estereotipo: Abstracto

Autor: Estado:

Alcance: Complejidad:

Alias:

Lenguaje:

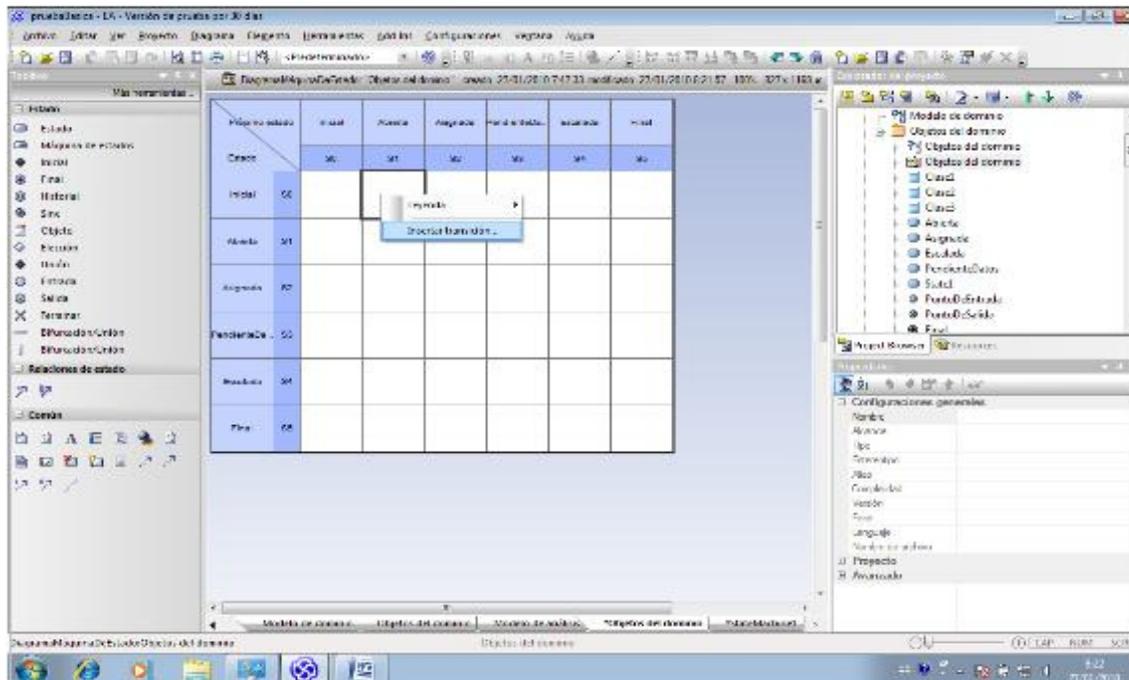
Palabras clave:

Fase: Versión:

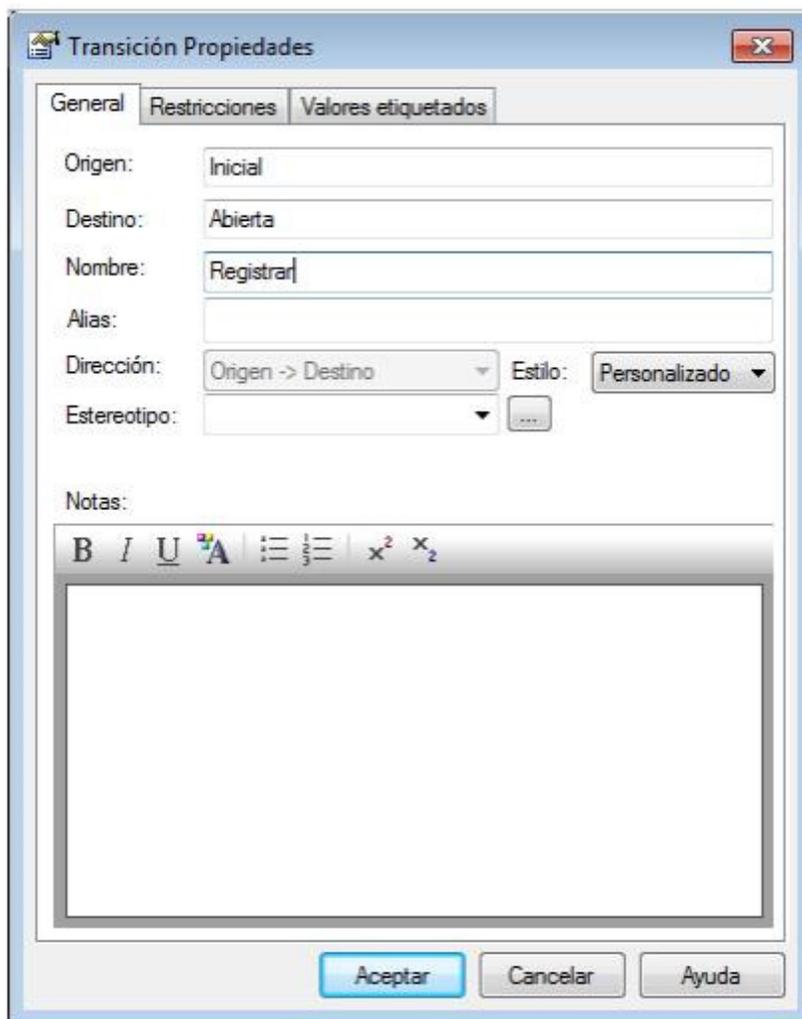
Notas:

B I U A x^2 x_2

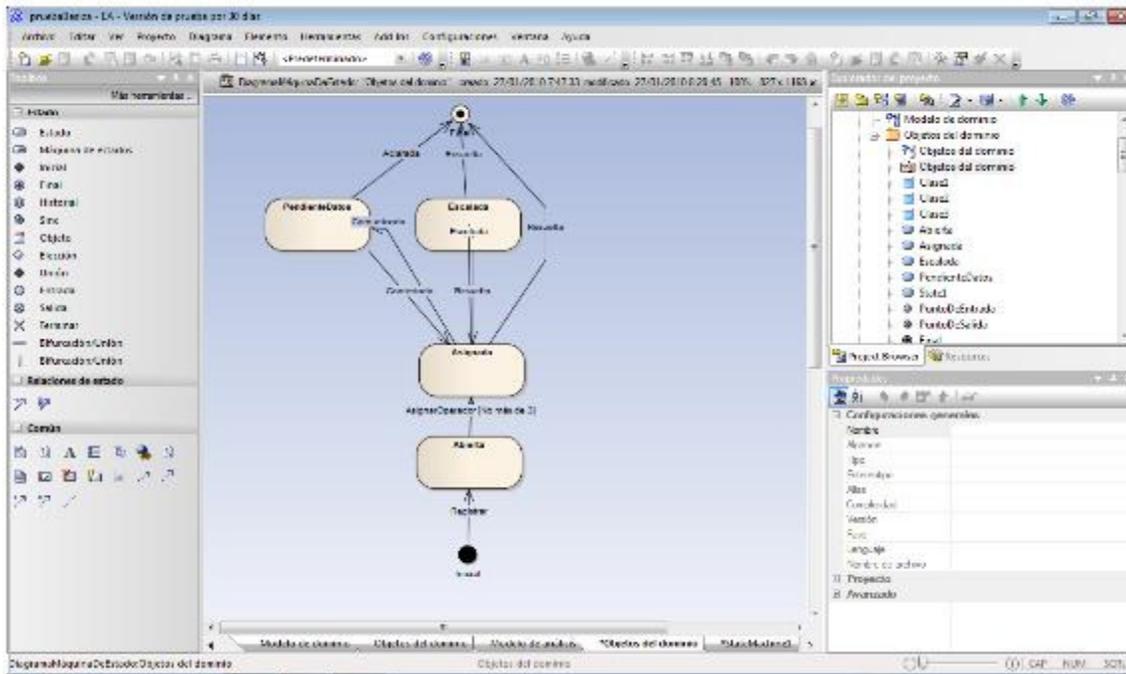
Y se nos va conformando la tabla.



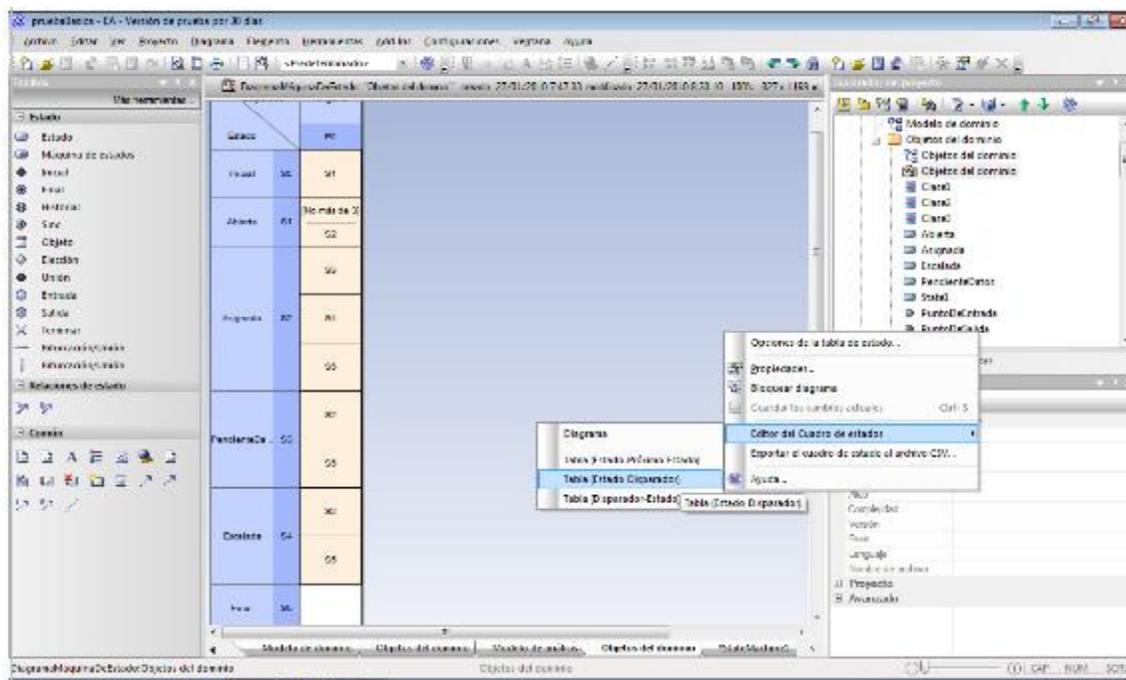
Sobre la cuadrícula pinchamos en insertar transición.



Y si cambiamos a modo gráfico, vemos que se une todo. Probablemente no os quede muy bonito pero podéis dar a auto-organizar (aunque normalmente te toca organizarlo a mano igual)

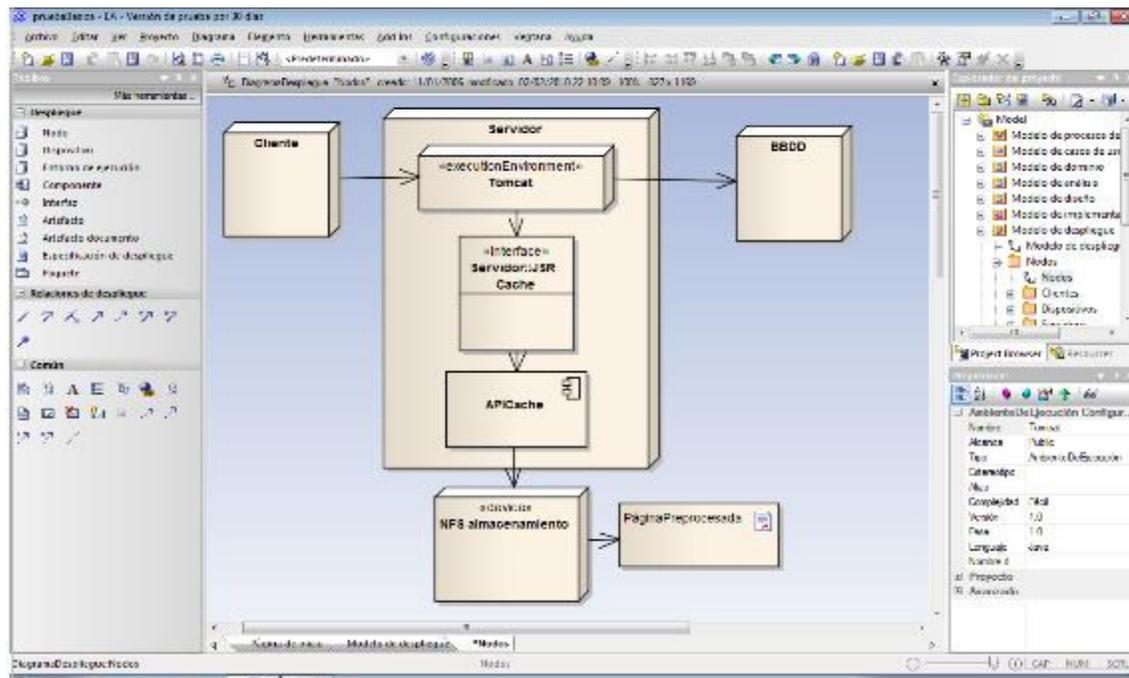


Tenemos más modos de ver la misma tabla de estados comprobar de cual pasa a cual.



Tampoco me voy a enrollar mucho más. Enterprise Architect me parece una herramienta muy buena, rápida y no ha me ha hecho cosas raras.

Ahora solo hay que sacarle partido pero recordad una cosa (siempre bajo mi criterio), es más importante tener un método secuencial para descubrir un proyecto en base a unos diagramas concretos que saber para que vale la última bolita de la herramienta (que siempre puede y debe aclararse con notas) o forzar usar todos los tipos de diagramas.



11. Conclusiones.

Espero haberos aclarado ciertas dudas o ideas del "maravilloso" mundo de los diagramas UML... como veis no son tan complicados como parecen. Si teneis más inquietudes sobre diagramas UML o bien quereis ampliar vuestros conocimientos sobre el tema deciros que en Autentia impartimos cursos sobre esta temática y otras (Patrones de diseño, buenas prácticas, etc.).