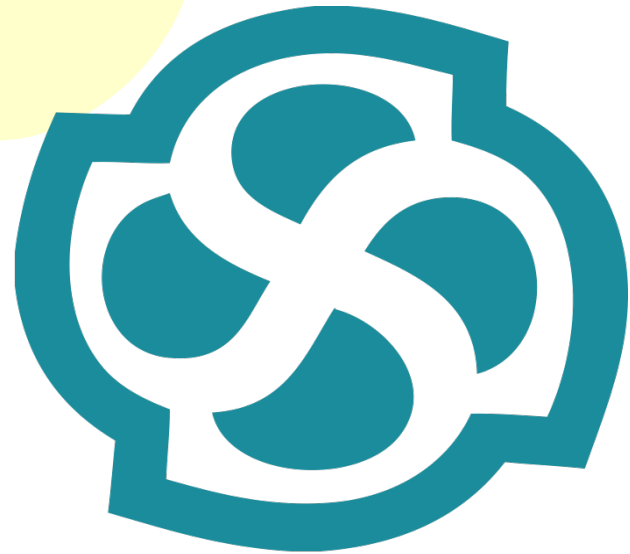


# ENTERPRISE ARCHITECT



# TEMARIO

## OBJETIVOS

PROMOVER EL USO DE LA HERRAMIENTA CORPORATIVA EN LAS FASES DE ANÁLISIS, DISEÑO FUNCIONAL, ARQUITECTURA Y DISEÑO TÉCNICO.

# INTRODUCCIÓN

## LA NECESIDAD DE MODELAR

Si quieres construir una caseta para un perro, se puede comenzar con unos tablones de madera y unos clavos ... ¿Y si quieres hacer un bloque oficinas?

# INTRODUCCIÓN

<<UN MODELO ES UNA SIMPLIFICACIÓN DE LA REALIDAD>>

Dispositivos eléctricos, centralitas telefónicas, incluso películas de cine (storyboarding) son MODELADAS antes de implementarse

# INTRODUCCIÓN

<<CONSTRUIMOS UN MODELO PORQUE NO  
PODEMOS COMPRENDER EL SISTEMA EN SU  
TOTALIDAD>>

El enfoque OO forma parte de la tendencia principal en la construcción de software, simplemente porque ha demostrado su validez.

# INTRODUCCIÓN

Unified Modeling Language nació de por iniciativa privada de una colaboración entre grande compañías (IBM, Rational, Oracle, HP, Texas Ins, Unisys) en 1997, agrupadas en el OMG (Object Management Group) para su estandarización.

La versión actual es la 2.5 y data de Junio de 2015

# INTRODUCCIÓN

Enlaces de interés:

<http://www.omg.org/>

<http://www.uml.org/index.htm>

<http://www.omg.org/issues/uml-rtf.html>

# UML

Modelar usando el lenguaje UML nos permite:

VISUALIZAR  
ESPECIFICAR  
CONSTRUIR  
DOCUMENTAR





# UML

## VISUALIZAR

Describimos qué hace y cómo hace nuestro sistema de forma estándar, detallando aspectos que trascienden a la programación textual

# UML

## ESPECIFICAR

Se emplea un modelado preciso, sin ambigüedades, que cubre las decisiones de análisis, diseño, e implementación durante el desarrollo y el despliegue

# UML

## CONSTRUIR

Gracias a la analogía conceptual entre los lenguajes de programación y UML, se permite la ingeniería directa, generando código a partir del modelo

# UML

## DOCUMENTAR

La arquitectura del proyecto, queda inherentemente descrita, además de proporcionar un lenguaje para expresar requisitos, pruebas, actividades de planificación y gestión de versiones

# UML- VOCABULARIO

HAY 3 BLOQUES DE CONSTRUCCIÓN

ELEMENTOS

RELACIONES

DIAGRAMAS

# UML- ELEMENTOS

## ELEMENTOS

ESTRUCTURALES  
DE COMPORTAMIENTO  
DE AGRUPACIÓN  
DE ANOTACIÓN

# UML- ELEMENTOS

## ELEMENTOS ESTRUCTURALES

CLASES

INTERFACES

COLABORACIONES

CASO DE USO

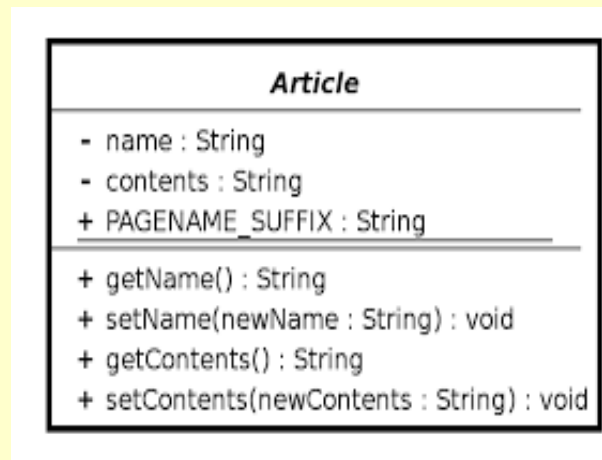
CLASES ACTIVAS

COMPONENTES

NODOS

# E ESTRUCTURALES

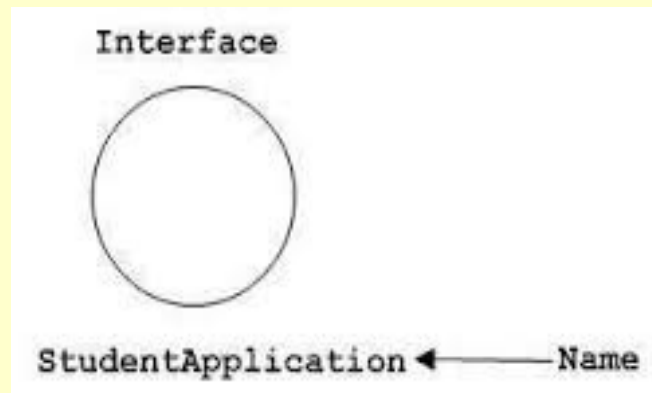
CLASES.- Conjunto de objetos que comparten atributos, operaciones, relaciones y semántica





# E ESTRUCTURALES

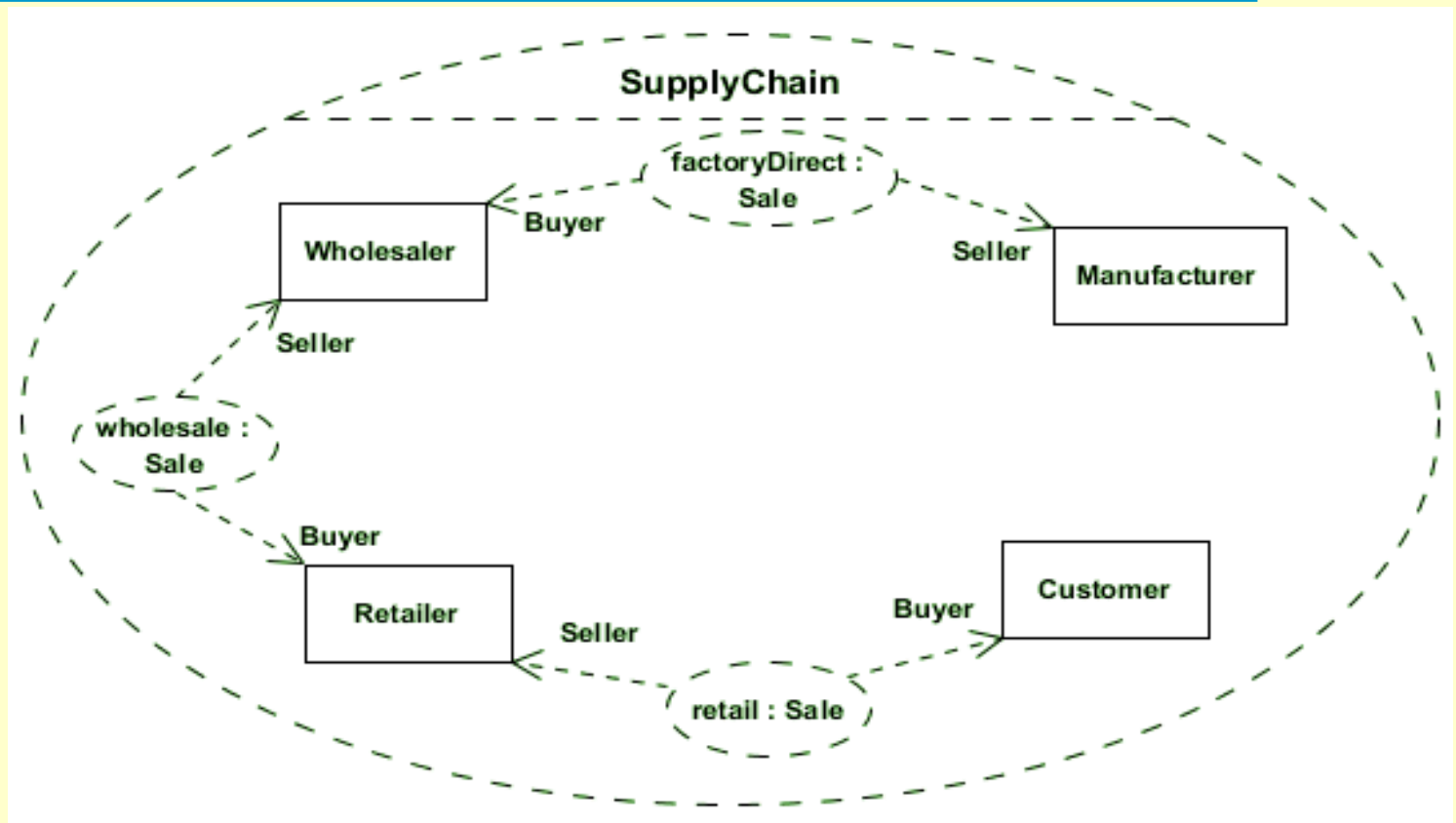
INTERFACES.- Colección de operaciones que especifican un servicio o comportamiento visible (externo)



# E ESTRUCTURALES

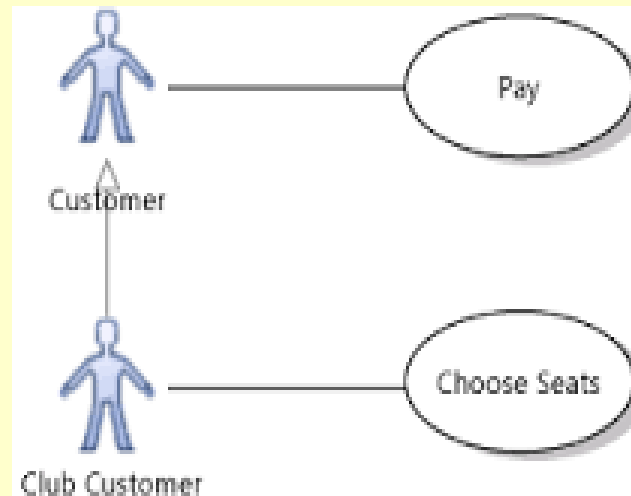
COLABORACIONES.- interacción que define un comportamiento cooperativo mayor que la suma de comportamientos de sus elementos. Ofrecen por tanto una dimensión estructural y funcional o de comportamiento

# E ESTRUCTURALES



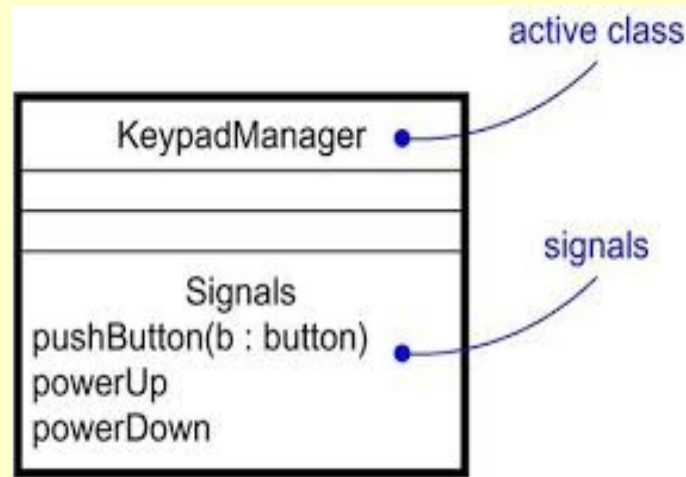
# E ESTRUCTURALES

CASO DE USO.- Descripción de secuencia de acciones que el sistema ejecuta y produce un resultado observable y de interés para un actor



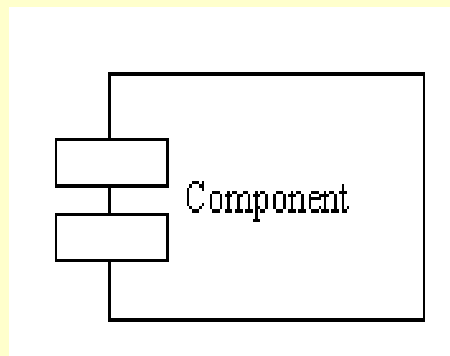
# E ESTRUCTURALES

CLASE ACTIVA.- Sus objetos tienen uno o más hilos de ejecución, pudiendo dar lugar a actividades de control. Objetos con comportamiento concurrente



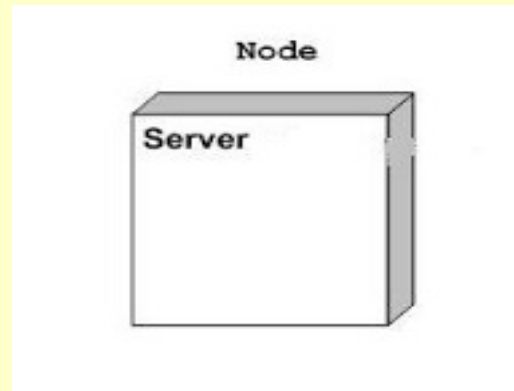
# E ESTRUCTURALES

COMPONENTE.- Parte física reemplazable que conforma un conjunto de elementos



# E ESTRUCTURALES

NODO.- Elemento físico que existe en ejecución y representa un recurso computacional, que dispone de memoria y tiempo de procesamiento



# UML- ELEMENTOS

## ELEMENTOS DE COMPORTAMIENTO

Parte dinámica. Los VERBOS de un Modelo, representan comportamiento en espacio y tiempo

INTERACCIÓN

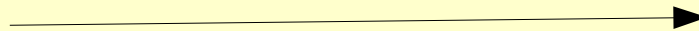
ESTADO



# E DE COMPORTAMIENTO

INTERACCIÓN.- Mensajes intercambiados entre un conjunto de objetos en un contexto particular, así como secuencias de acción y enlaces

dibujar



# E DE COMPORTAMIENTO

ESTADO.- Expresa el estado por el que atraviesa un objeto o interacción durante su vida, en respuesta a eventos y sus reacciones y evoluciones

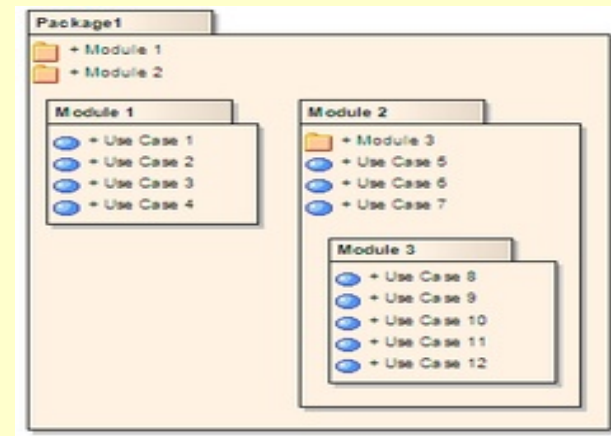


ESPERANDO

# UML- ELEMENTOS

ELEMENTOS DE AGRUPACIÓN .- Partes organizativas. Las cajas en las que se puede descomponer un modelo

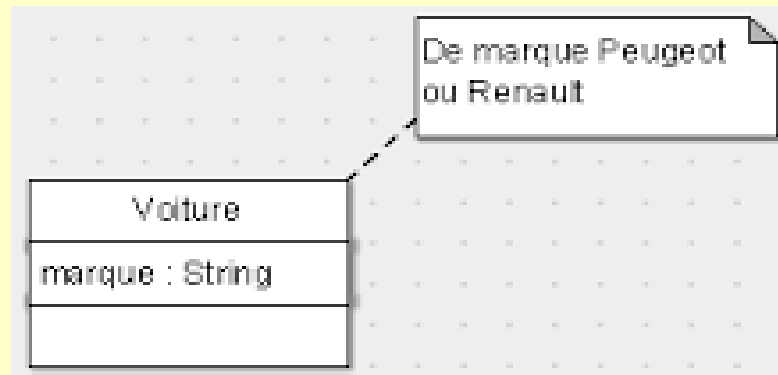
PAQUETES



# UML- ELEMENTOS

ELEMENTOS DE ANOTACIÓN .- Partes explicativas. Comentarios que se pueden aplicar sobre cualquier elemento del modelo

NOTA



# RELACIONES

HAY 4 TIPOS DE RELACIONES

DEPENDENCIA ----->

ASOCIACIÓN 0..1 patrón \_\_\_\_\_ empleado

GENERALIZACIÓN —————>

REALIZACIÓN .....>

# DIAGRAMAS

El diagrama es la representación gráfica de un conjunto de elementos, visualizado como un grafo conexo.

Un diagrama puede contener cualquier combinación de elementos y relaciones.

# DIAGRAMAS

La última versión de UML (2.5 Jun '15) define 14 tipos diferentes de diagramas, clasificados en dos grupos: Estáticos y Dinámicos.

Cada tipo cuenta con sendos 7 diagramas.

# DIAGRAMAS Estáticos

Los diagramas estáticos definen la arquitectura estática de un modelo. Se modelan las cosas físicas y lógicas que componen el sistema, así como para especificar relaciones y dependencias entre los distintos elementos.



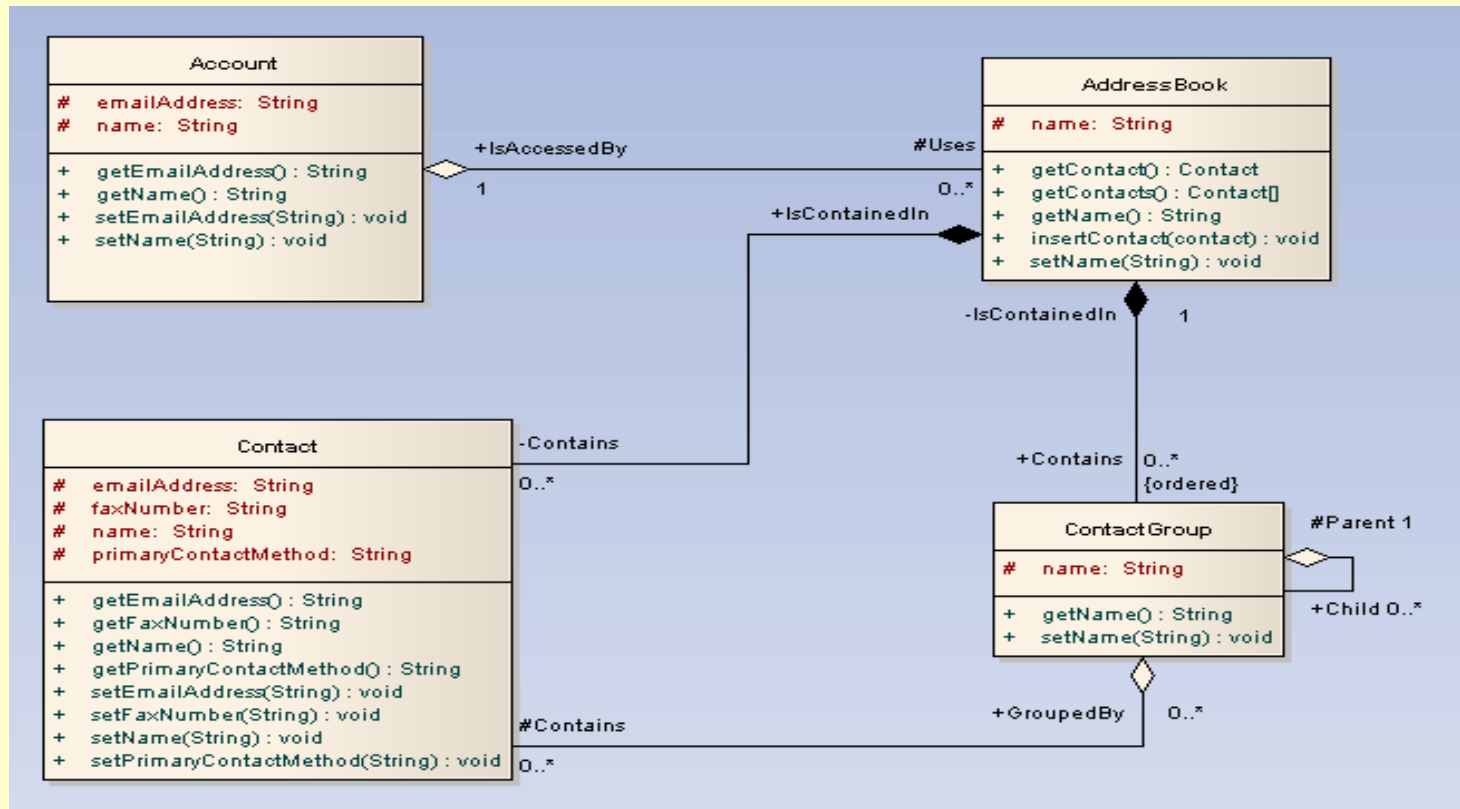
# DIAGRAMAS Dinámicos

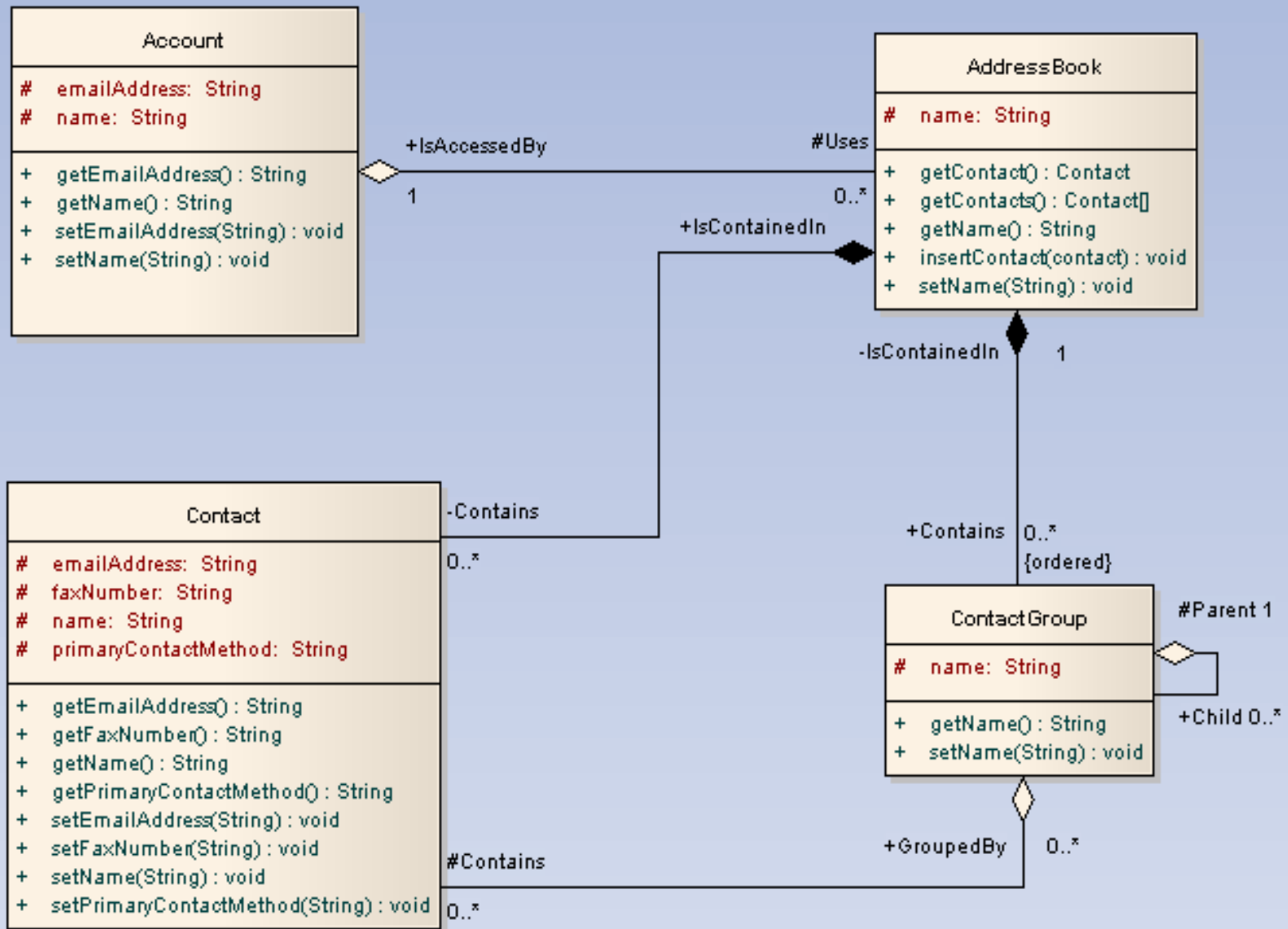
Los diagramas dinámicos o de comportamiento describen la interacción y los estados del sistema durante su ejecución. Da respuesta a modelar cómo se comporta nuestro sistema, y qué efectos producen las operaciones y eventos y cuáles son los resultados de interés producidos.

# DIAGRAMAS Estáticos

DIAGRAMA DE CLASES o ESTRUCTURAL.- Qué clases y métodos alberga mi sistema bajo la perspectiva de la OO. Refleja también las relaciones entre clases e interfaces, así como generalizaciones, agregaciones y asociaciones colaboraciones y sus relaciones por composición y uso

# DIAGRAMA de CLASES

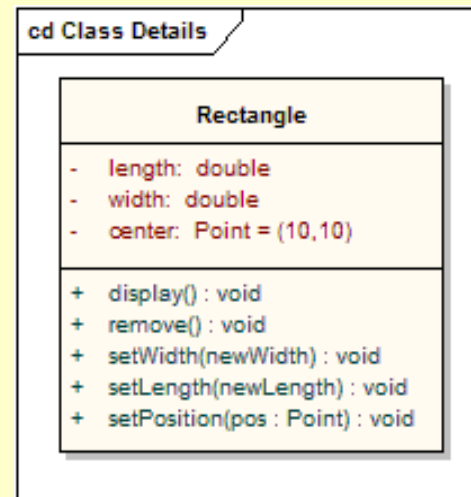




# DIAGRAMA de CLASES

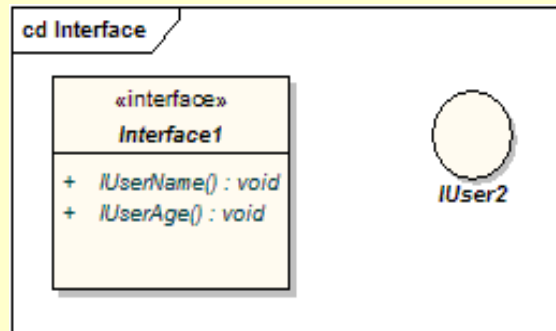
Se indican los atributos y métodos, siguiendo a siguiente convención para la visibilidad

- + Públicos
- Privados
- ~ Paquete



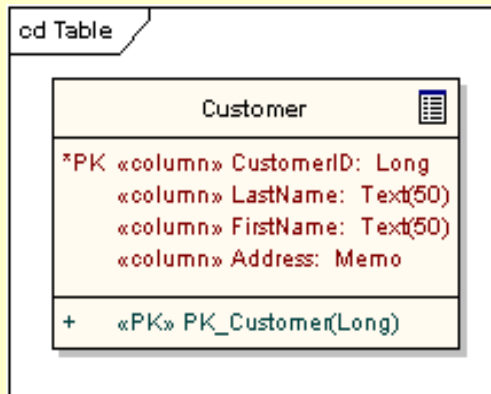
# DIAGRAMA de CLASES

Las interfaces expresan un comportamiento y se pueden representar alternativamente de dos modos:



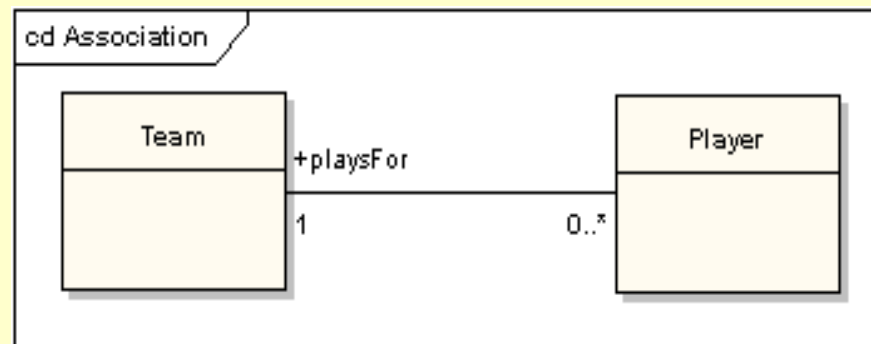
# DIAGRAMA de CLASES

Gracias al empleo de estereotipos como extensión de UML; puedo definir clases “persistentes” con el icono de la tabla en la esquina superior derecha:



# DIAGRAMA de CLASES

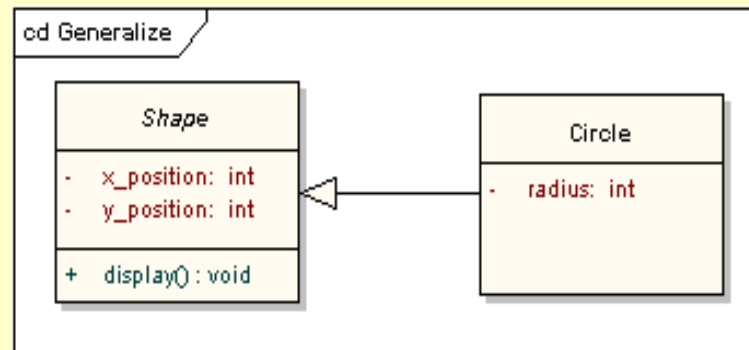
Las asociaciones, normalmente implementadas con un atributo por composición. En tal caso, el nombre de la asociación, constituye a posteriori el nombre del atributo





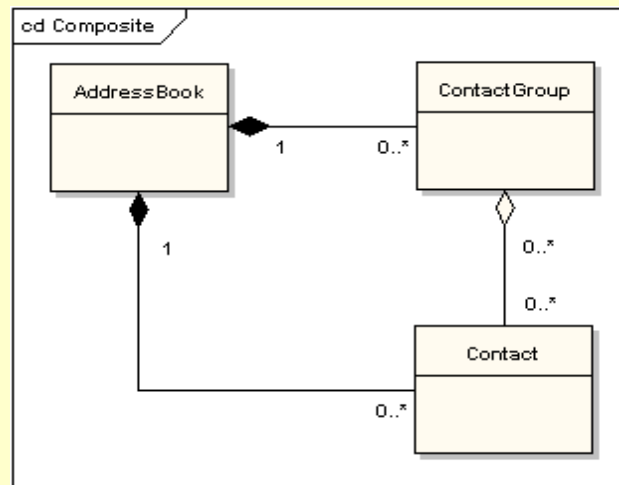
# DIAGRAMA de CLASES

Generalización: para expresar relaciones de herencia



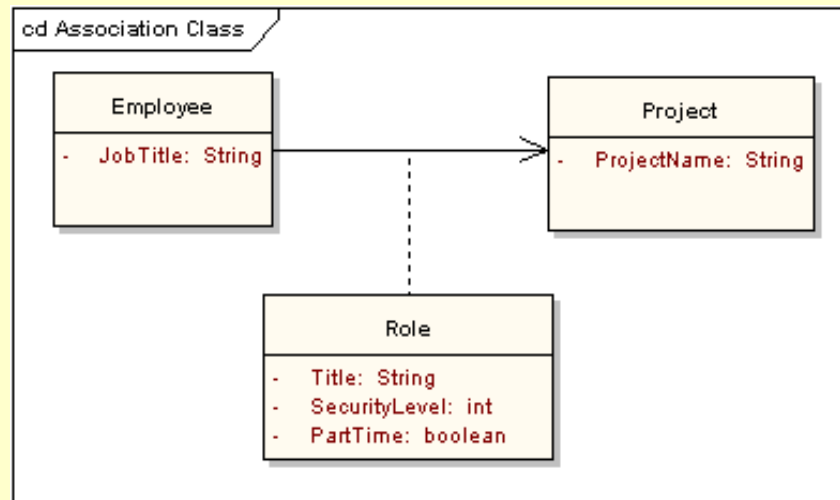
# DIAGRAMA de CLASES

Las agregaciones describen qué elementos más pequeños constituyen otro. La agregación fuerte (punta rellena) indica dependencia en existencia



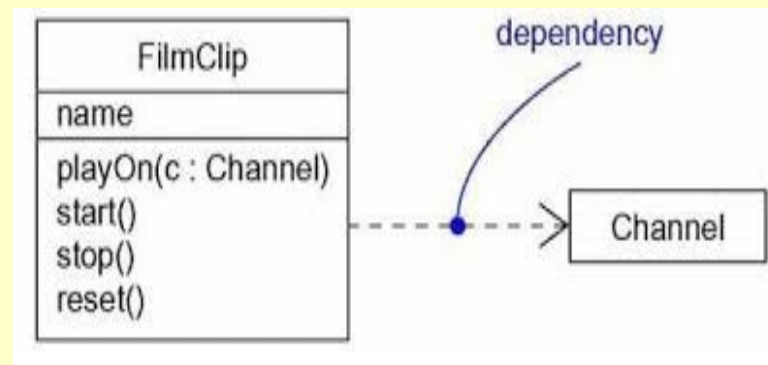
# DIAGRAMA de CLASES

Si una relación entre dos clases necesita de clases y atributos da lugar a una ASOCIACIÓN



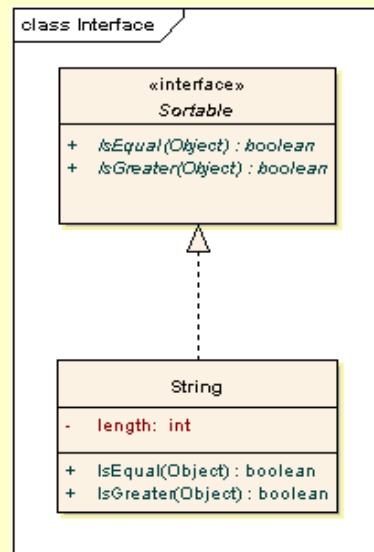
# DIAGRAMA de CLASES

Si una una fase preliminar del diseño se intuye un tipo de relación entre clases pero no cuál concretamente, se habla de dependencia



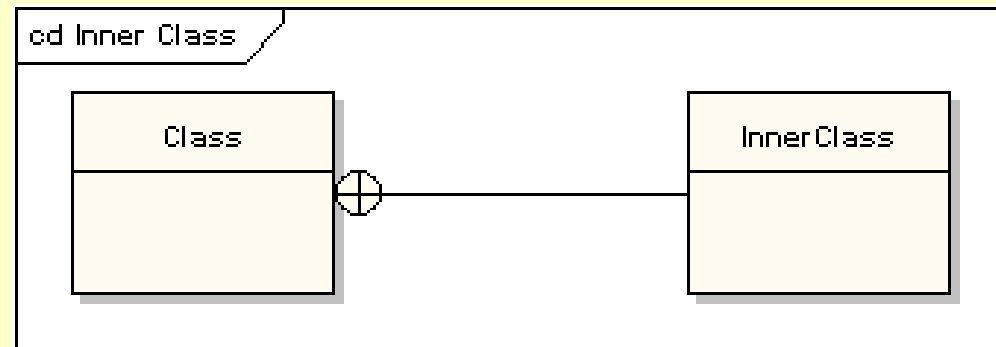
# DIAGRAMA de CLASES

La realización es la implementación de una interfaz por una clase (flecha cerrada)



# DIAGRAMA de CLASES

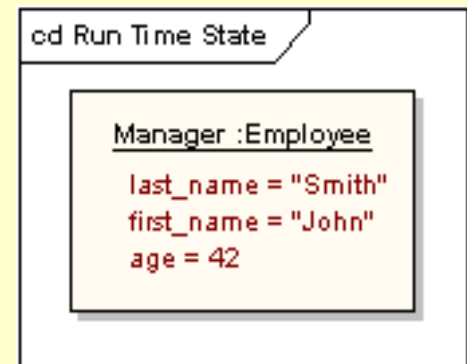
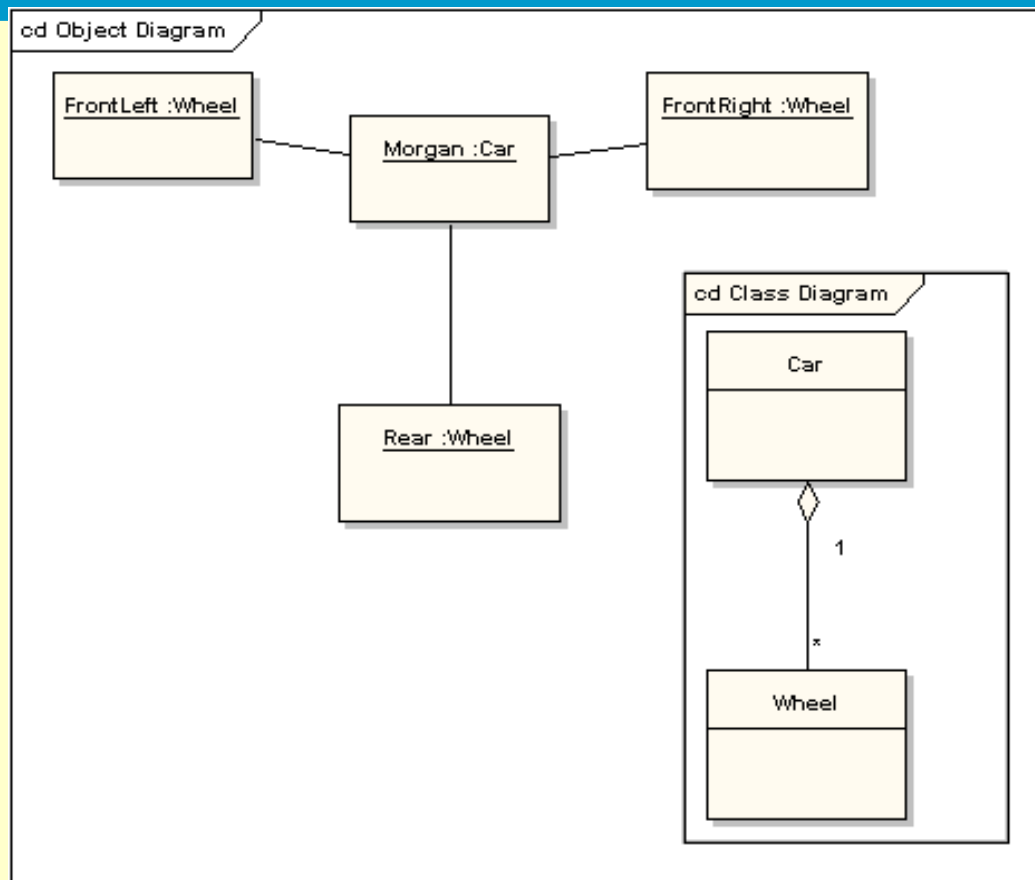
Cuando se quiere explicitar la existencia de una clase interna (inner class), se usa la notación de anidamiento



# DIAGRAMAS Estáticos

DIAGRAMA DE OBJETOS.- Un caso particular del diagrama de clases. Da vida, enfatiza a los diagramas de clases, ejemplificando roles y cardinalidades de las relaciones, ayudando a comprender partes del diagrama de clases

# DIAGRAMA de OBJETOS

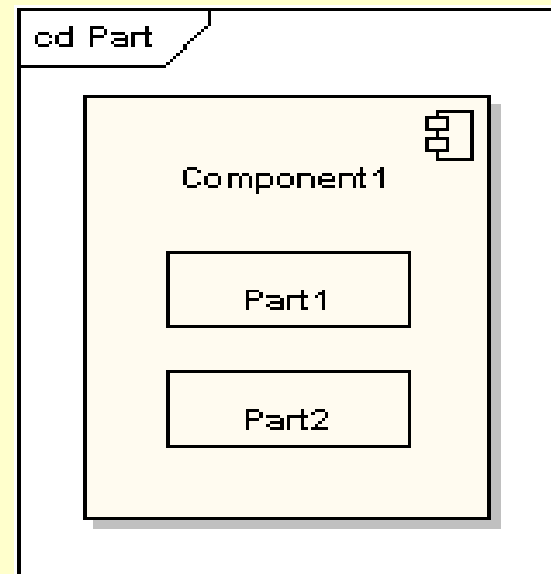
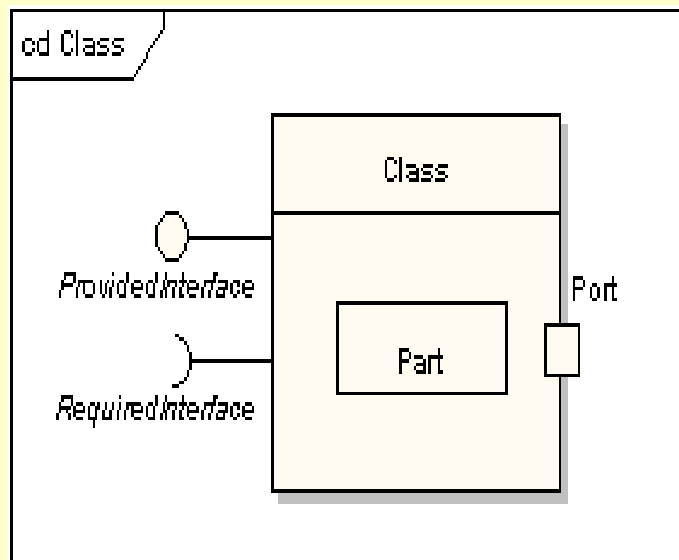




# DIAGRAMAS Estáticos

DIAGRAMA COMPUESTOS.- Aplicado a Clases o Componentes complejos de los cuales quiere obtenerse mayor detalle, especificando las partes, los puertos, las interfaces ofrecidas y consumidas, así como las colaboraciones y el comportamiento que ofrece la estructura interna.

# DIAG COMPUESTOS



# DIAG COMPUESTOS

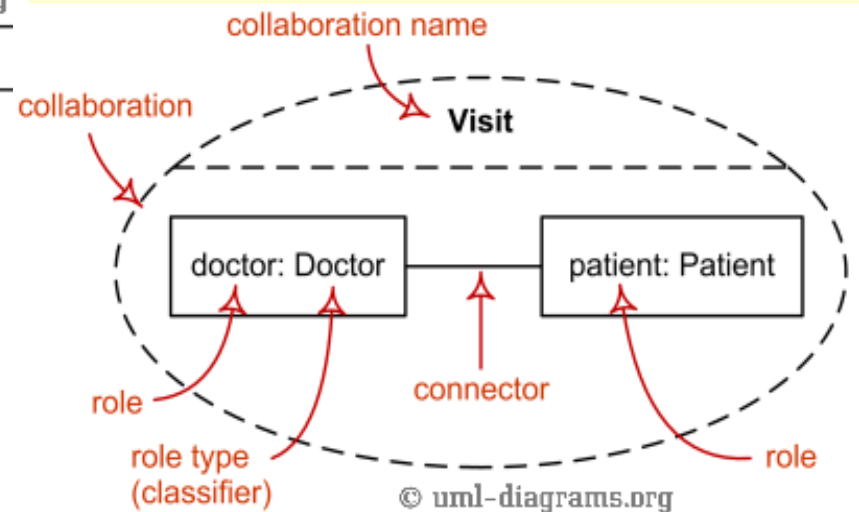
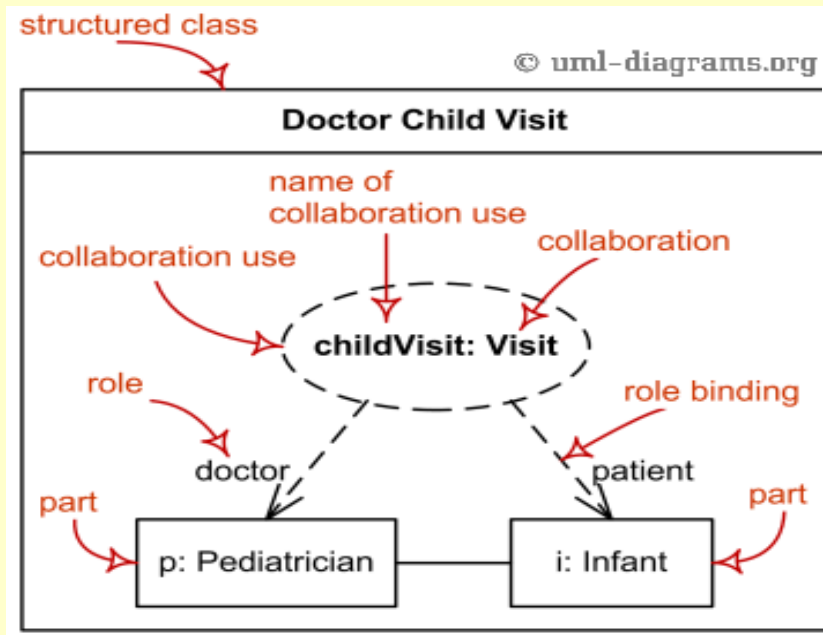
Partes.- Caja en blanco dentro del elemento. Parte identificable del componente

Puerto.- Interacción con su entorno como consumidor o proveedor

Interfaz.- Provista (círculo completo) o requerida (abierto)

Colaboración.- Roles, clases o componentes implicados.

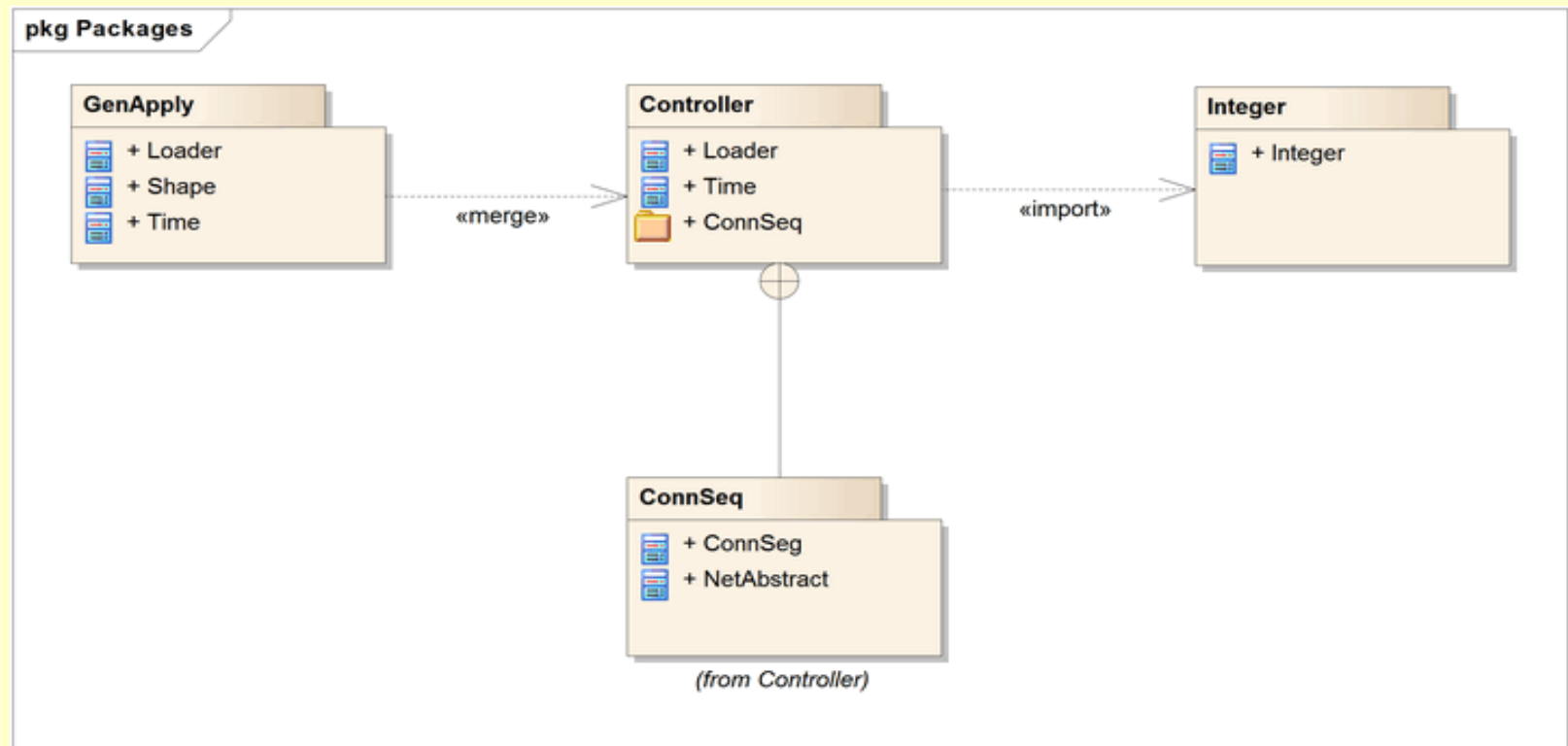
# DIAG COMPUESTOS



# DIAGRAMAS Estáticos

DIAGRAMA DE PAQUETES.- Refleja la organización de paquetes y sus elementos . Normalmente, se emplean para agrupar diagramas de clases y/o diagramas de casos de uso, aunque no está limitado a esos elementos UML

# DIAGRAMA de PAQUETE



# DIAGRAMA de PAQUETE

Los elementos incluidos comparten namespace, luego deben tener nombres únicos (ID)

Los paquetes expresan relaciones físicas y lógicas. Se deben incluir en el mismo paquete clases que heredan de un jerarquía común, tiene relaciones por composición o clases que colaboran con ellas

# DIAGRAMA de PAQUETE

La visibilidad de los elementos se establece mediante

- + Visibilidad pública
- Visibilidad privada
- # Visibilidad reservada



# DIAGRAMA de PAQUETE

El conector «merge» entre dos paquetes, identifica que el componente del paquete fuente aumenta su definición e incluye al elemento del paquete destino.

«Import» refleja que el paquete origen emplea elementos del destino sin referenciar a su espacio de nombres

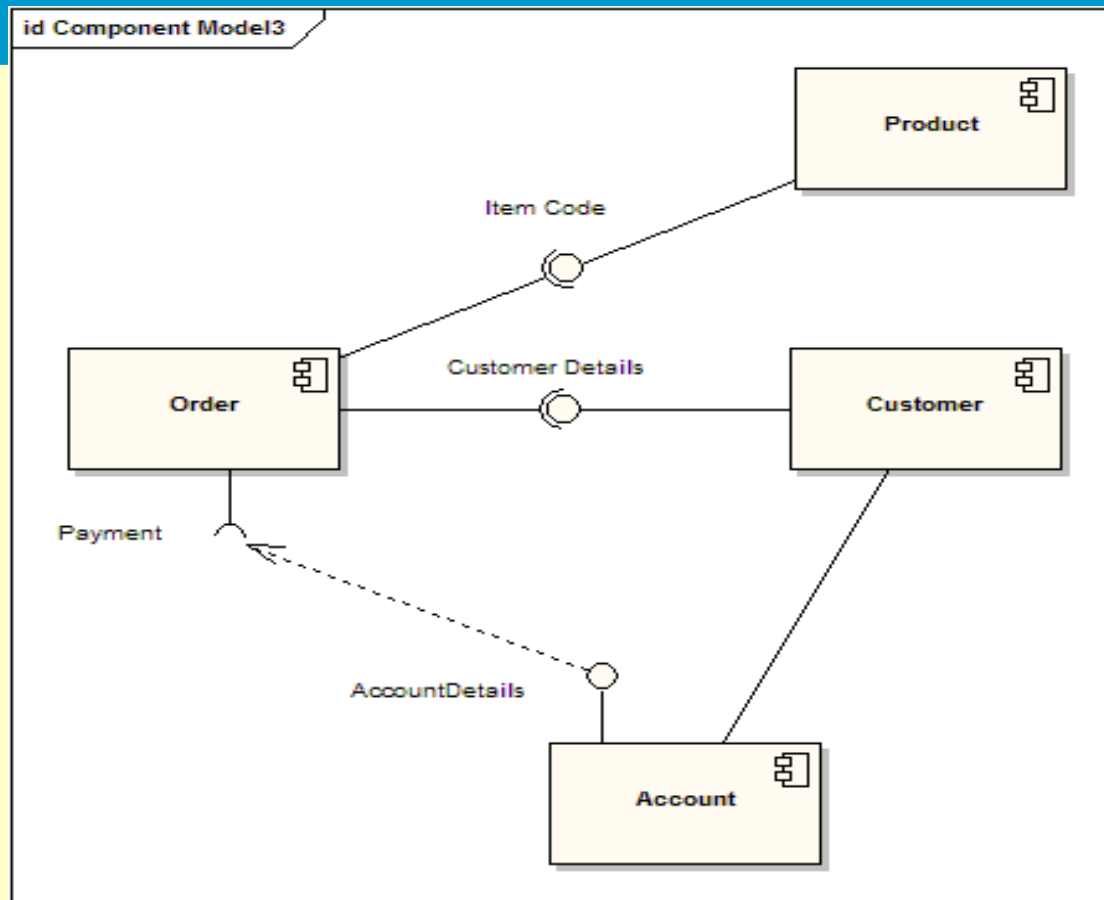
# DIAGRAMA de PAQUETE

El conector anidado, refleja que un paquete, está contenido dentro de otro. En el lado del contenedor, el paquete contenido se refleja como un carpeta.

# DIAGRAMAS Estáticos

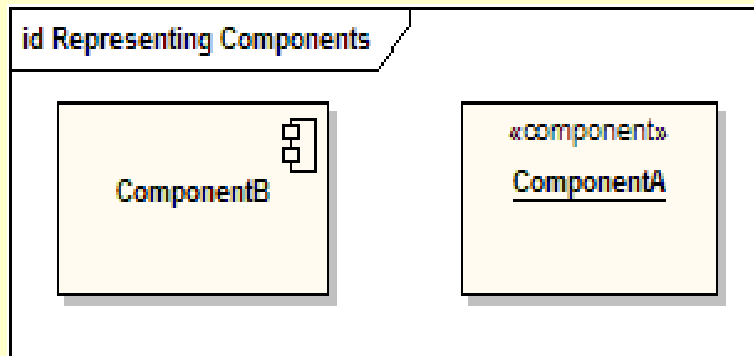
DIAGRAMA DE COMPONENTES.- Es un nivel de abstracción superior al diagrama de clases (normalmente, un componente está formado por varias clases) y parecido al de paquetes . Refleja qué componentes al más alto nivel constituyen el sistema y su relación entre si. Su misión es visualizar qué componentes pueden compartirse entre otros del sistema y por otros sistemas

# DIAG COMPONENTES



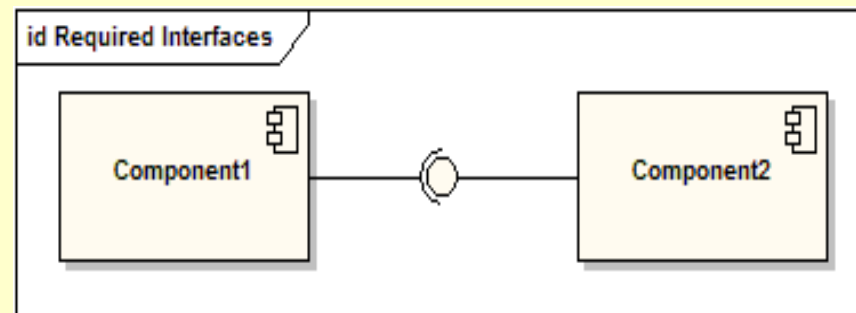
# DIAG COMPONENTES

Se pueden representar de dos maneras, de forma alternativa:



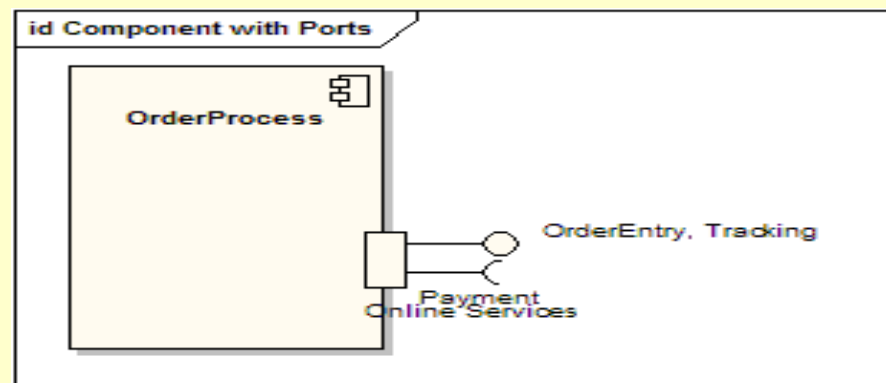
# DIAG COMPONENTES

Empleando el conector de ensamblado, se representa que un componente consume los servicios que otro ofrece:



# DIAG COMPONENTES

Se pueden indicar puertos de entrada y salida como punto de contacto del componente con el exterior (a modo de consumidor o proveedor de esos servicios)



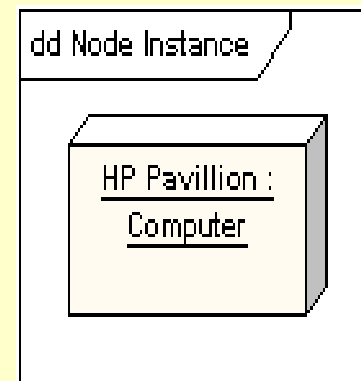
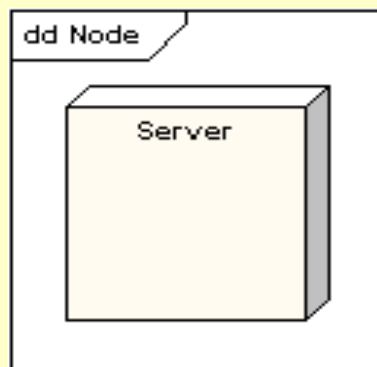
# DIAGRAMAS Estáticos

DIAGRAMA DE DESPLIGUE.- Qué elementos hardware forman el sistema (nodos) y qué elementos software hay desplegados en ellos.



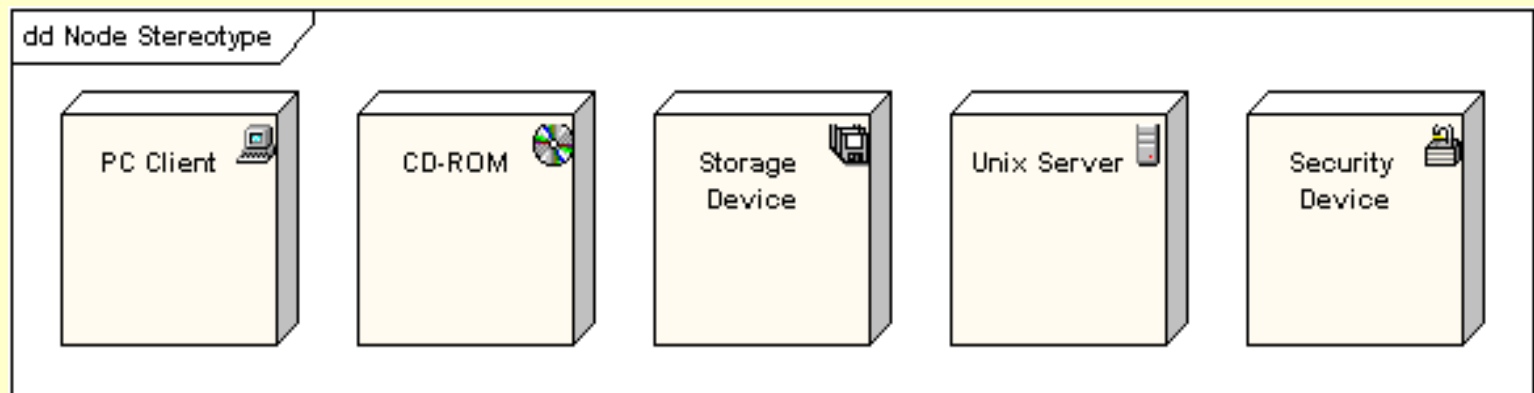
# DIAG DE DESPLIEGUE

Los nodos pueden ser elementos software o hardware, y se modelan como una cubo. Puede además, asumir un valor concreto (nodo instancia)



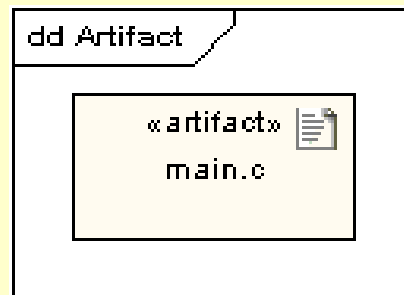
# DIAG DE DESPLIEGUE

Hay varios nodos estereotipo creados para facilitarnos la tarea en el diagrama de despliegue



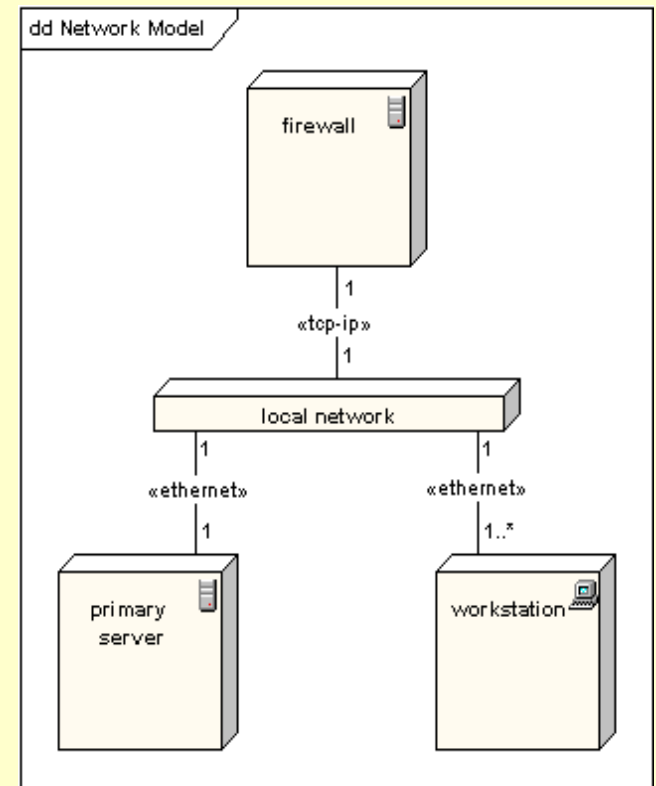
# DIAG DE DESPLIEGUE

Los artifacts (producto de un proceso de desarrollo software) pueden referirse a fuentes, ejecutables, informes, manuales. Se refieren con <<artifact>>



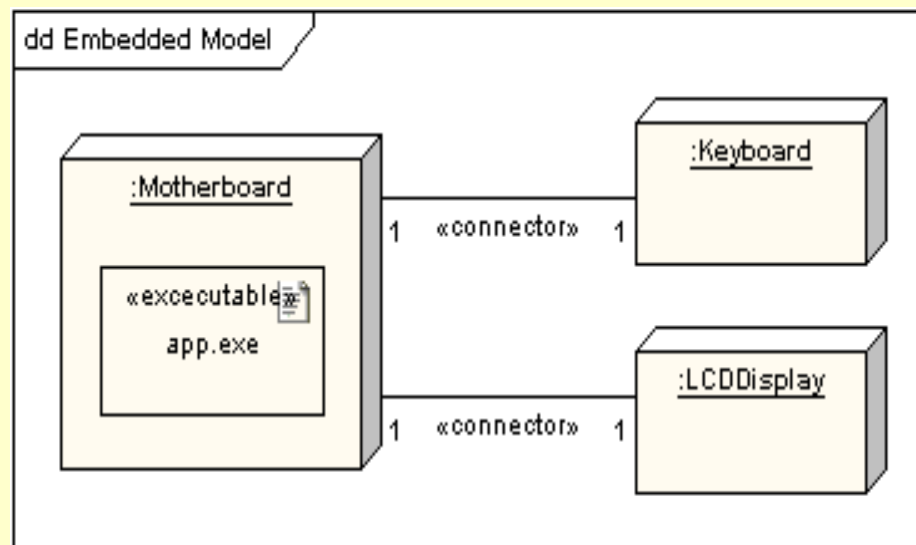
# DIAG DE DESPLIEGUE

La asociación se emplea como medio de comunicación Entre varios nodos del diagrama de despliegue indicando su cardinalidad



# DIAG DE DESPLIEGUE

Puedo expresar que un nodo contiene un artefacto o varios



# DIAGRAMAS Estáticos

DIAGRAMA DE PERFIL .- Permite confeccionar modelos UML personalizados, extendiendo la especificación estándar para definir un tipo específico adaptado a un dominio o a una restricción.

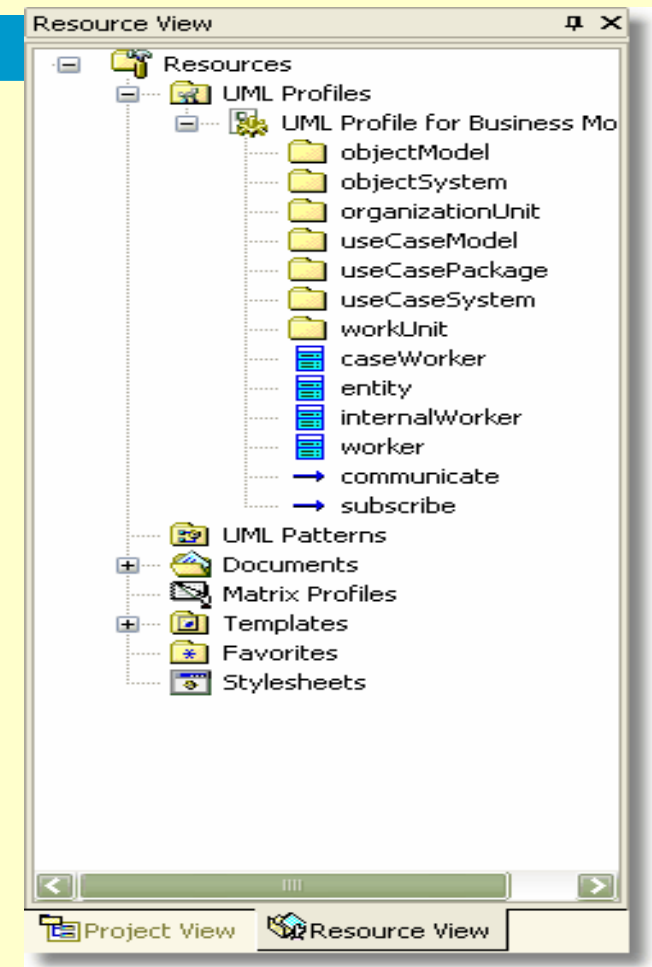
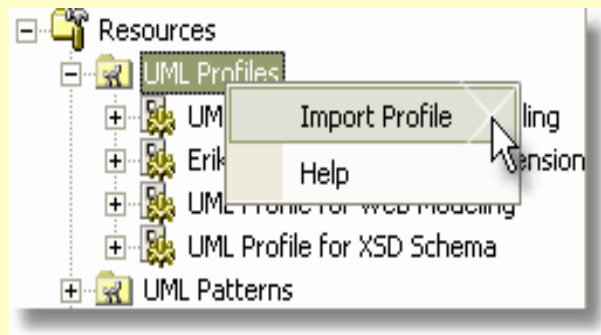
David Carlson

"Modeling XML Applications with UML"

# DIAGRAMAS de PERFIL

La herramienta Enterprise Architect, en sus versiones avanzadas (>8) permite la definición de extensiones, estereotipos y estructuras evolucionadas y heredadas del UML estándar, que quedan recogidas en un XML y permite su importación

# DIAGRAMAS de PERFIL





# DIAGRAMAS Dinámicos

Los diagramas de modelado dinámico o de comportamiento capturan las interacciones del sistema y los estados que adopta el sistema en su ejecución.

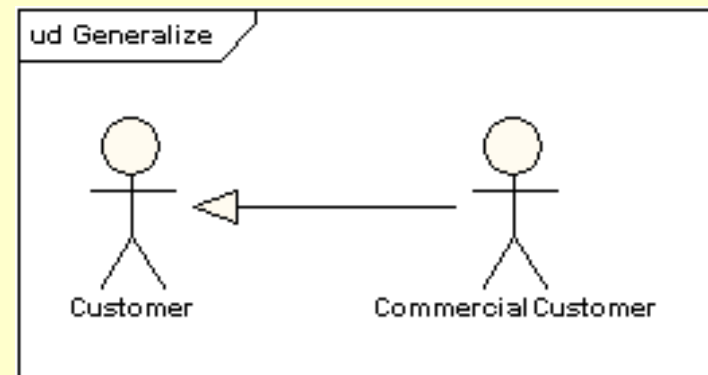
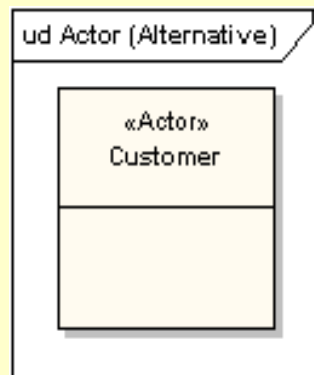
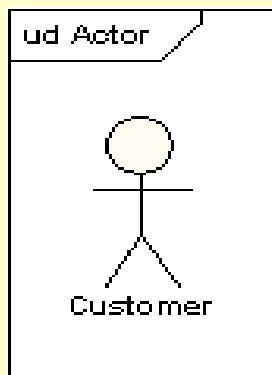
Presta atención a los resultados obtenidos que denotan cierto interés en la comprensión de la dinámica del sistema modelado.

# DIAGRAMAS Dinámicos

DIAGRAMA DE CASOS DE USO .- Captura los requisitos del sistema. Escenifica qué es el sistema para los usuarios y el resto del entorno

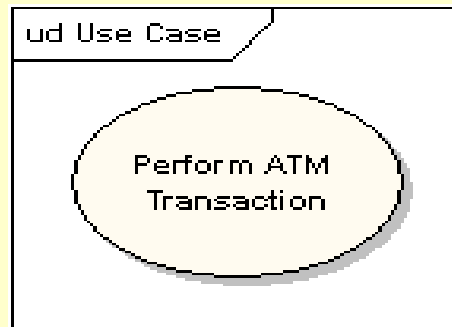
# DIAG De CASOS de USO

Los Actores, estancias externas al sistema que interactúan con él son representados con un muñeco o una caja <<Actor>>, pudiendo existir generalizaciones de actores



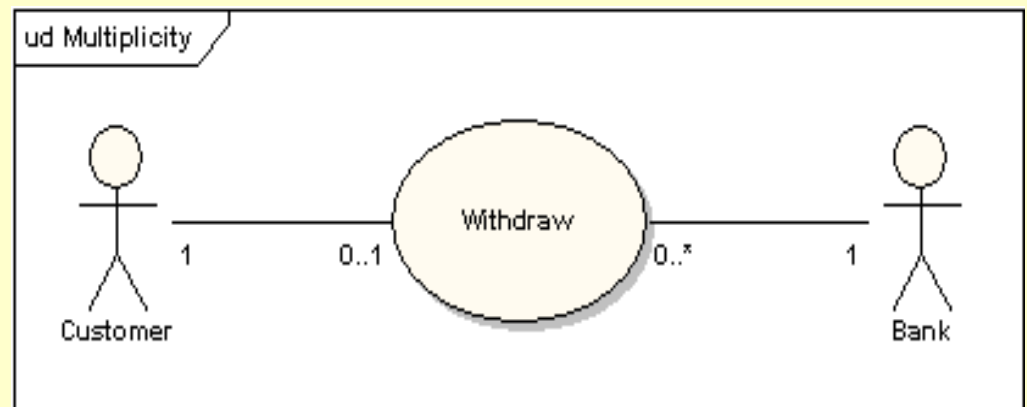
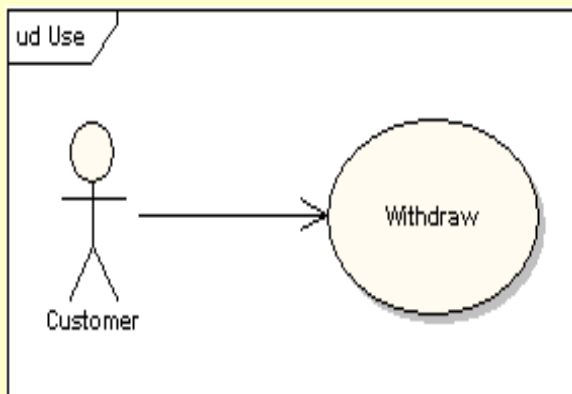
# DIAG De CASOS de USO

Un Caso de USO es una unidad de trabajo significativo para alguien ajeno al sistema. Representa por tanto, comportamiento a alto nivel. Se representa con una elipse y una descripción



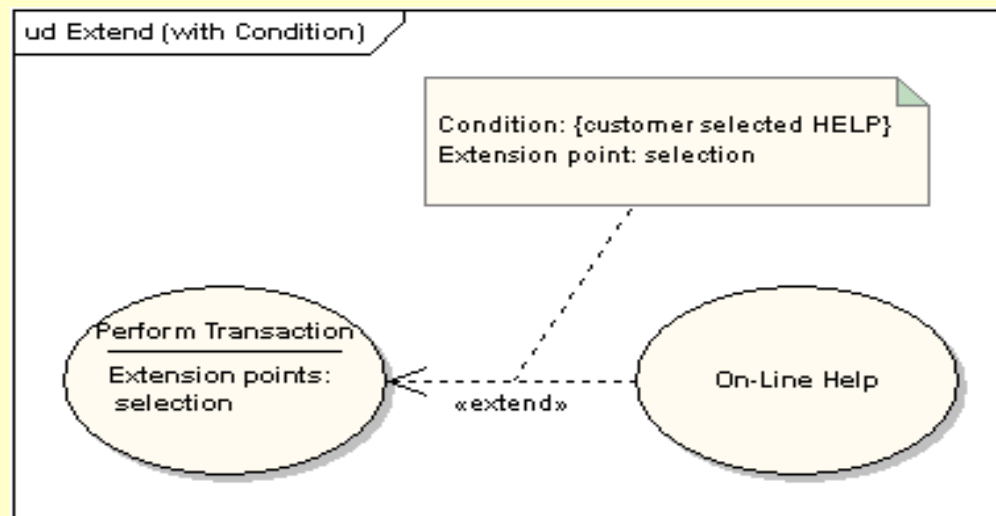
# DIAG De CASOS de USO

Para expresar que un usuario USA un caso de uso, se conectan mediante una flecha actor y caso. También puede añadirse multiplicidad para captar cuántas veces se produce ese uso.



# DIAG De CASOS de USO

Puede que un uso pueda incluir (extend) o implique (include) a otro bajo alguna condición, lo que se representaría así:



# DIAGRAMAS Dinámicos

DIAGRAMA SECUENCIA .- Muestra la vida de los objetos y las interacciones entre ellos para cubrir un proceso de interés, en el que cooperan varios actores o partes del sistema.

# DIAGRAMA Secuencia

Línea de vida : Punteada bajo el objeto referido (X para indicar su final)

Mensajes

- Síncronos (flecha de punta rellena)

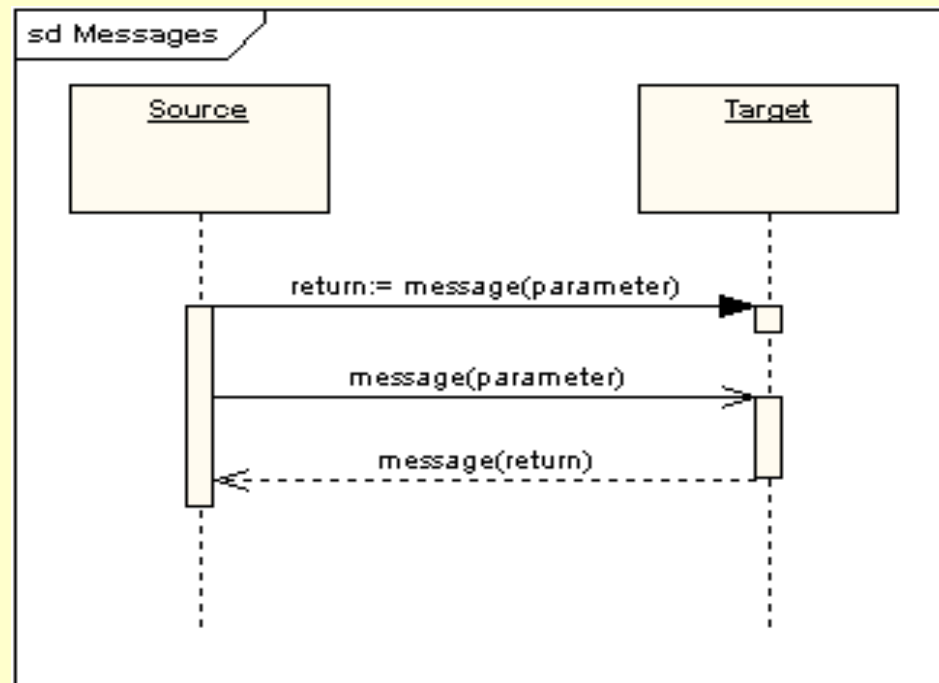
- Asíncronos (flecha abierta)

- Asíncrono y punteado (de respuesta)

Cuadrado: Mensaje entre objetos perdidos y de usuario desconocido



# DIAGRAMA Secuencia

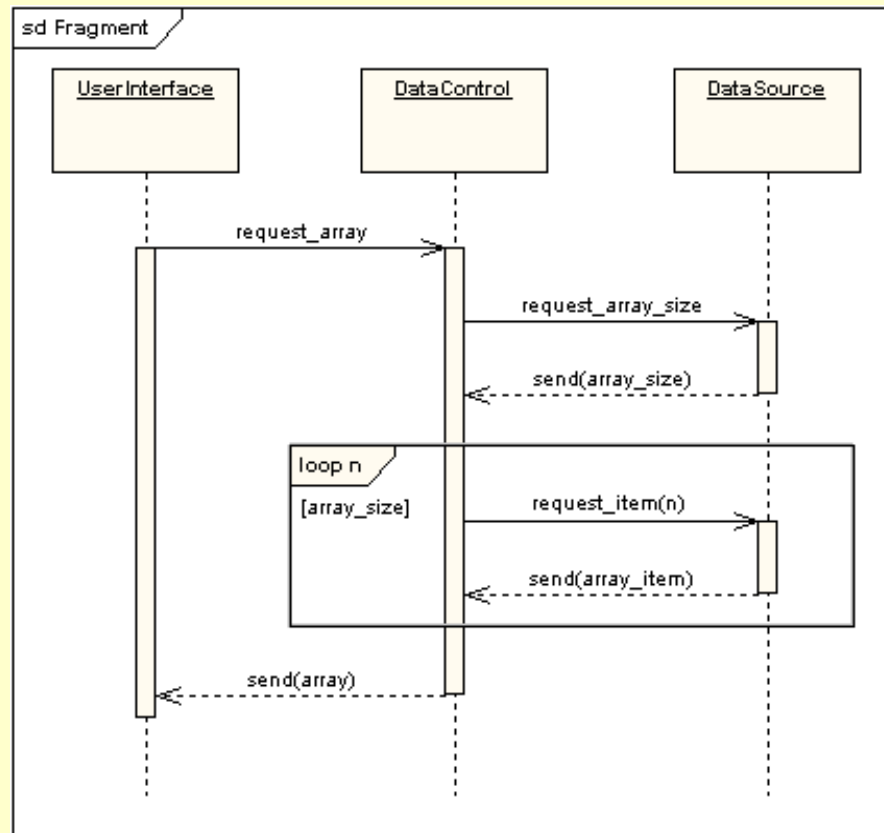


# DIAGRAMA Secuencia

Rectángulo: Tiempo de procesamiento

Se pueden agrupar líneas de vida y combinar así fragmentos, para expresar la cooperación mediante una condición o bucle

# DIAGRAMA Secuencia



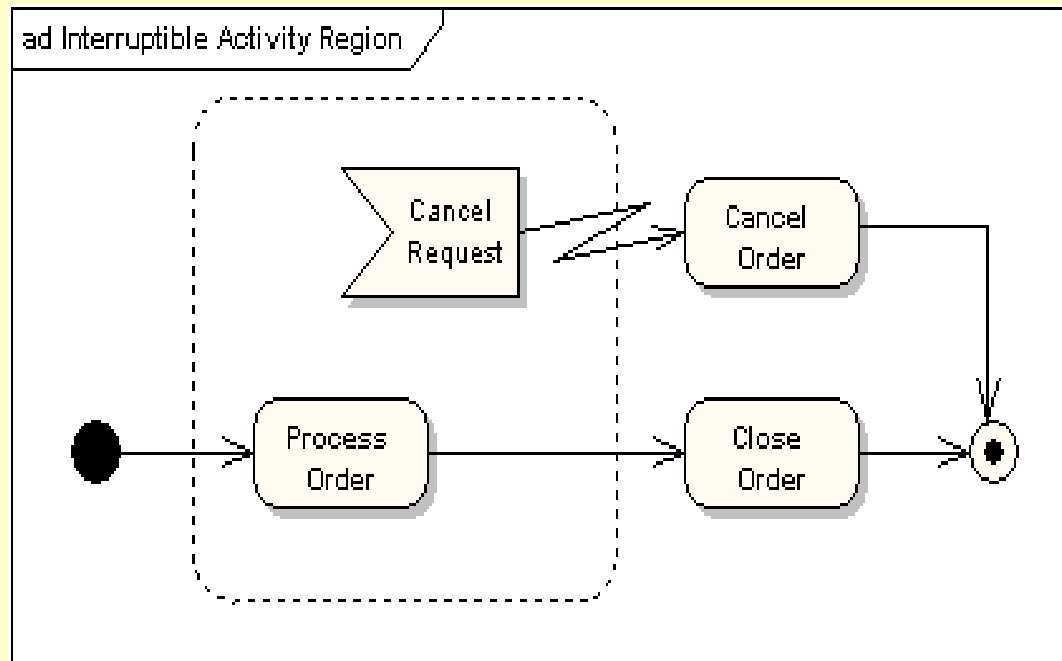
# DIAGRAMAS Dinámicos

DIAGRAMA DE ACTIVIDAD .- Se muestra conjunto de actividades enlazadas, reflejando las decisiones y los y eventos en la ejecución del proceso. Se deben usar para detallar procesos de negocio

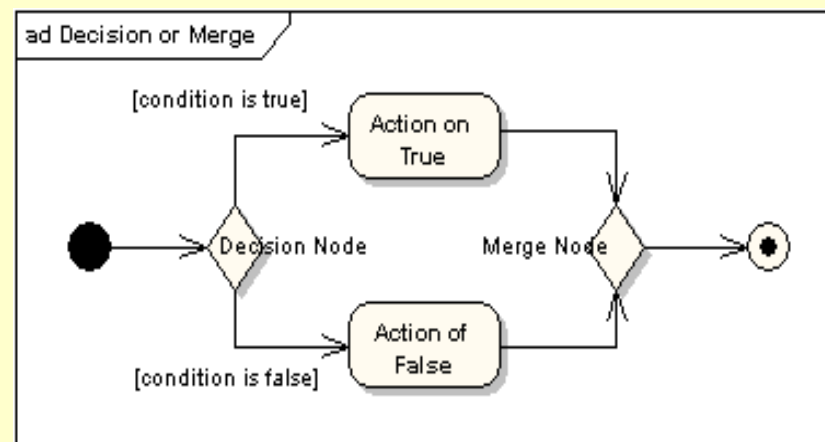
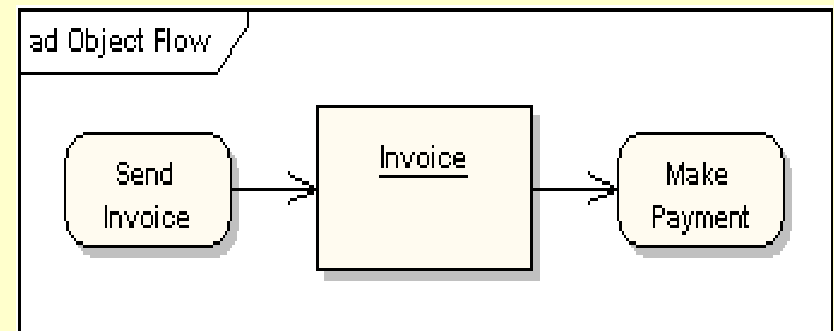
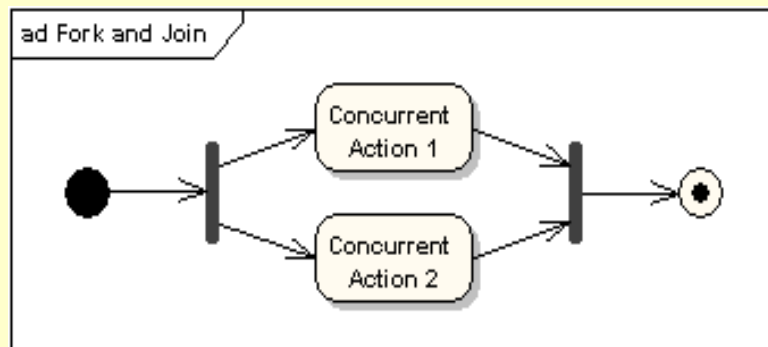
# DIAGRAMA de Actividad

Se muestran las actividades enlazadas por el flujo con un punto inicial y final, pudiendo representarse los datos en la transición (rectángulos), bifurcaciones (rombos), creación de hilos y sincronización (barritas) así como interrupciones (rayos)

# DIAGRAMA de Actividad



# DIAGRAMA de Actividad



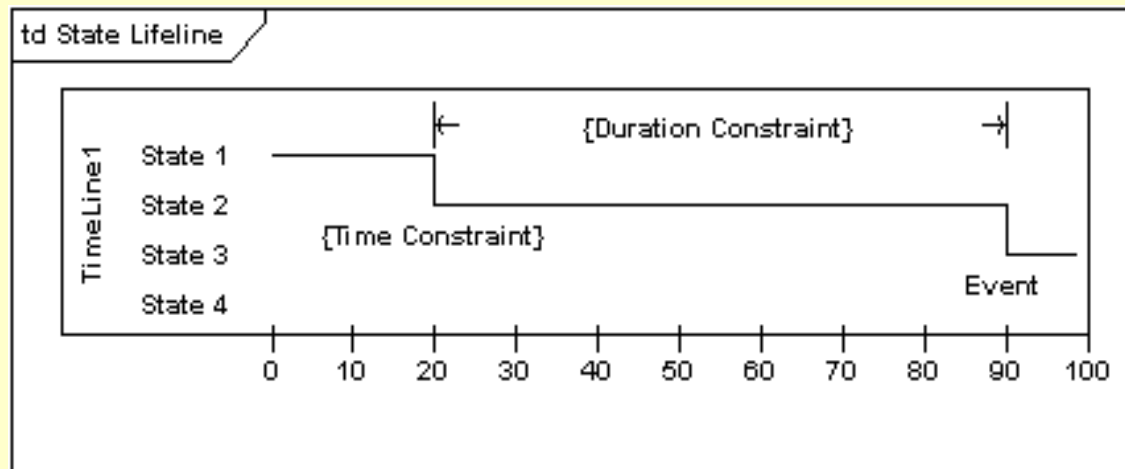
# DIAGRAMAS Dinámicos

DIAGRAMA DE TIEMPO .- Reflejan del estado de uno o más valores en el tiempo. Puede también mostrar la interacción entre eventos de tiempo y las restricciones que los sujetan



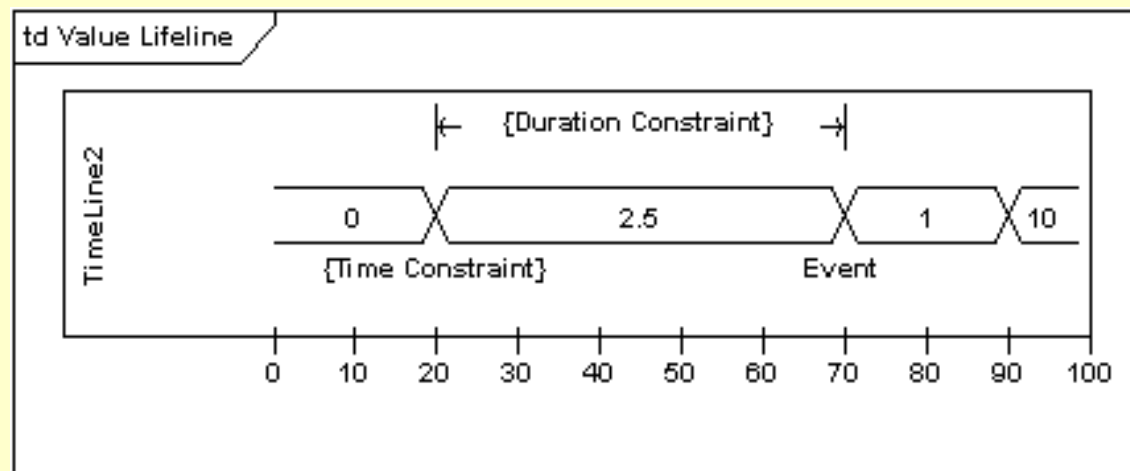
# DIAGRAMA de Tiempo

Las equis indican una medida de tiempo y las ies un estado para cada valor



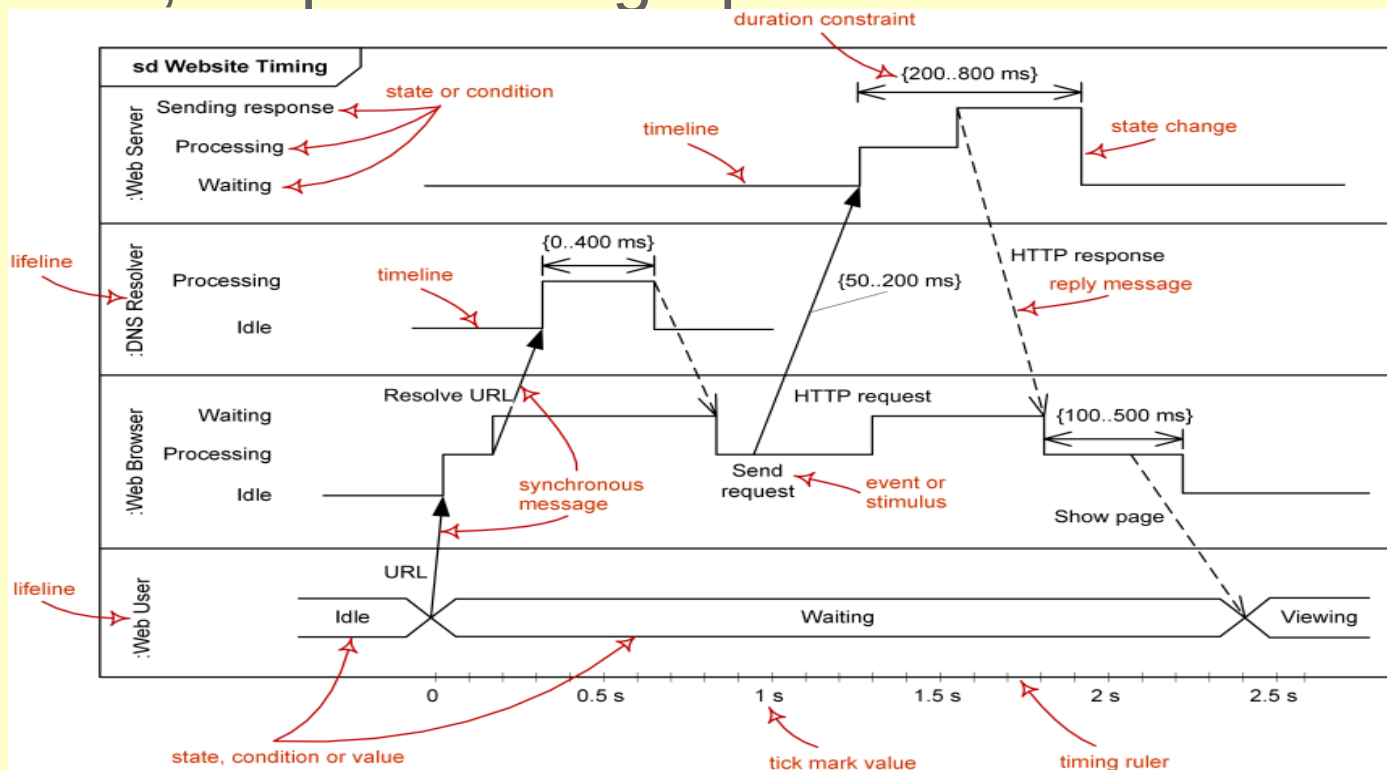
# DIAGRAMA de Tiempo

También pueden indicarse valores y estado dentro del un límite establecido por las líneas temporales



# DIAGRAMA de Tiempo

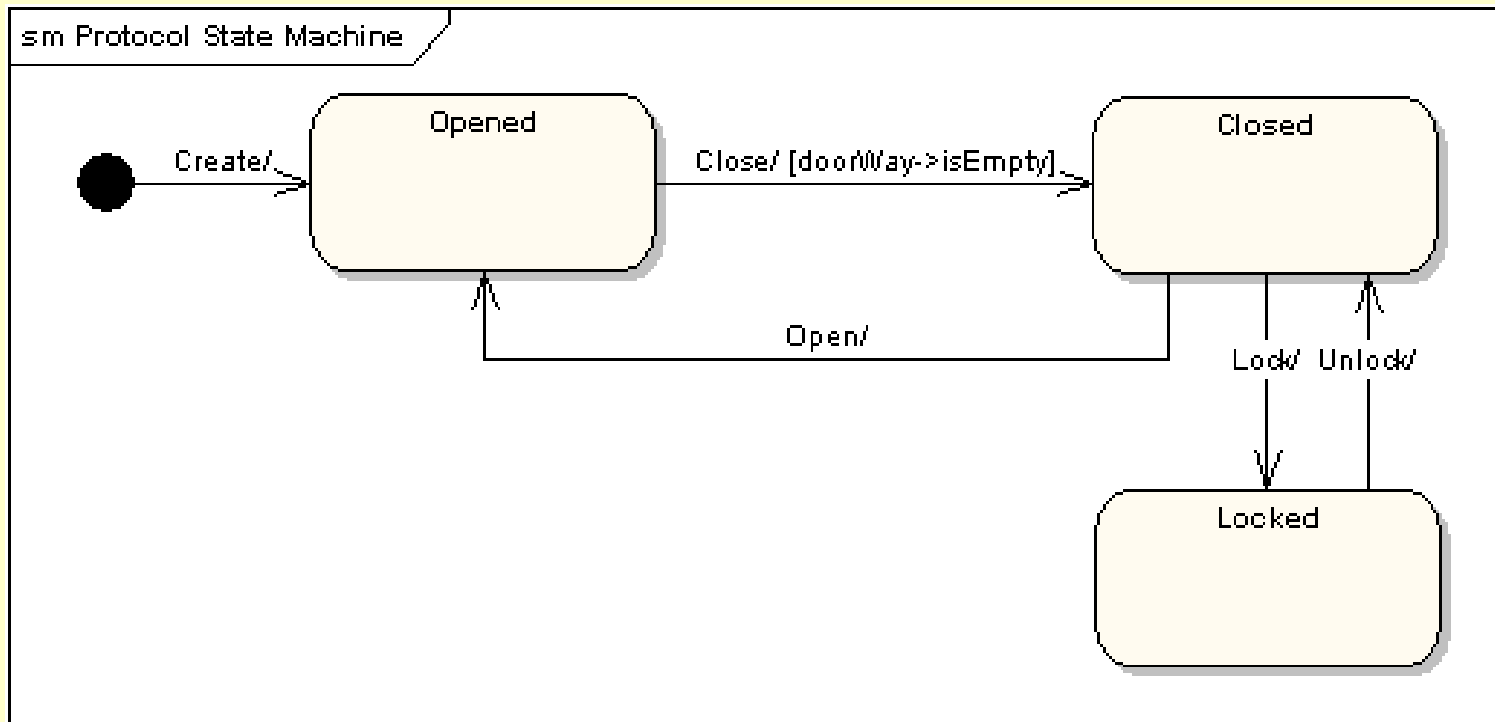
Además, se pueden agrupar actores



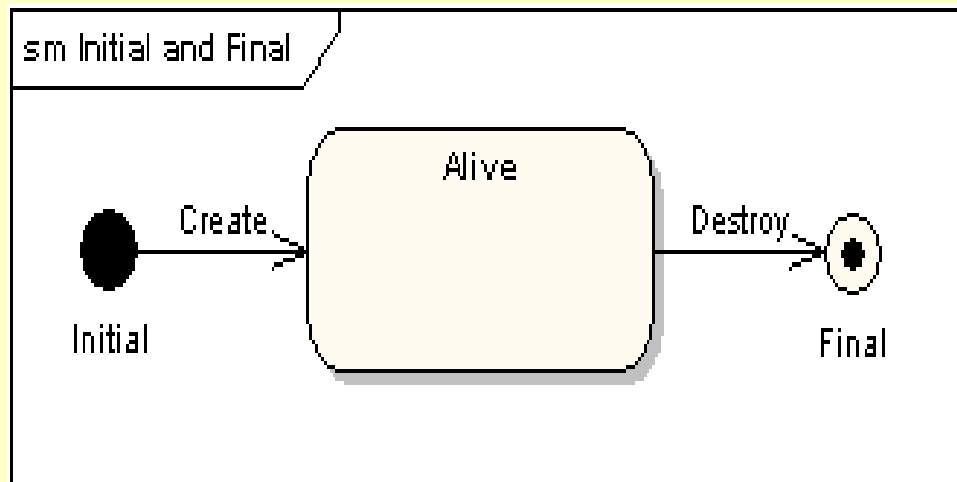
# DIAGRAMAS Dinámicos

DIAGRAMA DE ESTADOS .- Un diagrama de estados o una máquina de estados, muestra el comportamiento de un único objeto, detallando la secuencia de estados que presenta durante su tiempo de vida, en respuesta a los eventos que recibe

# DIAGRAMA de Estados

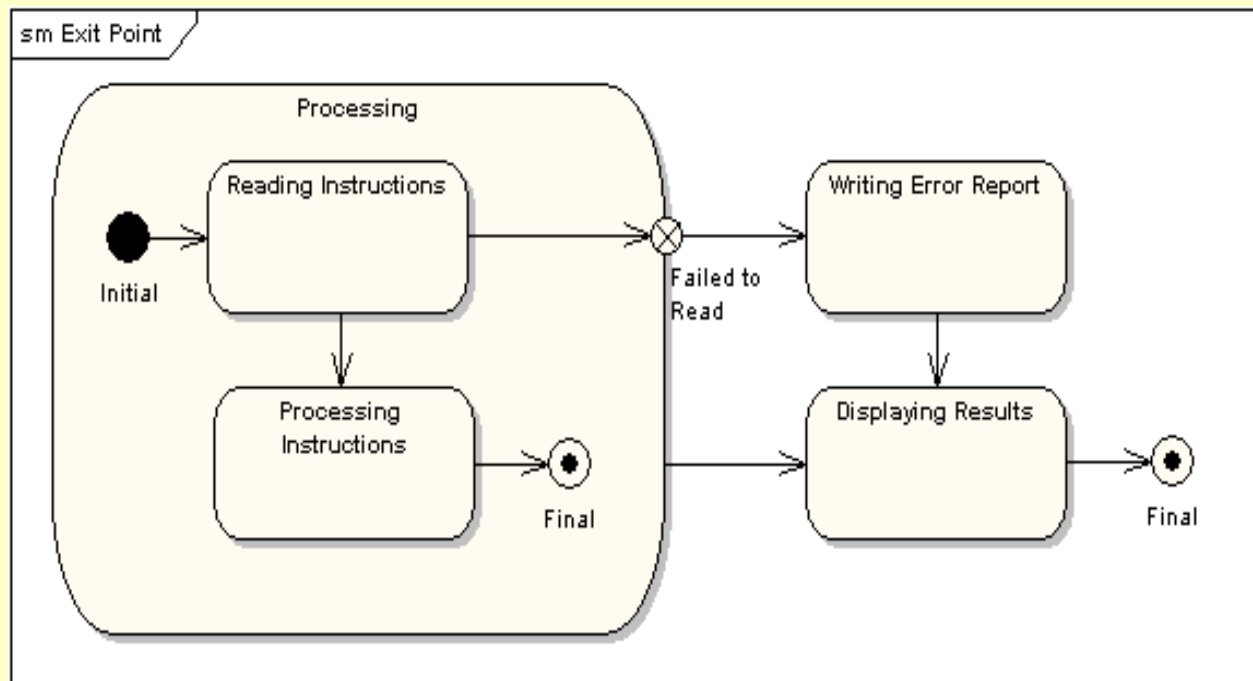


# DIAGRAMA de Estados



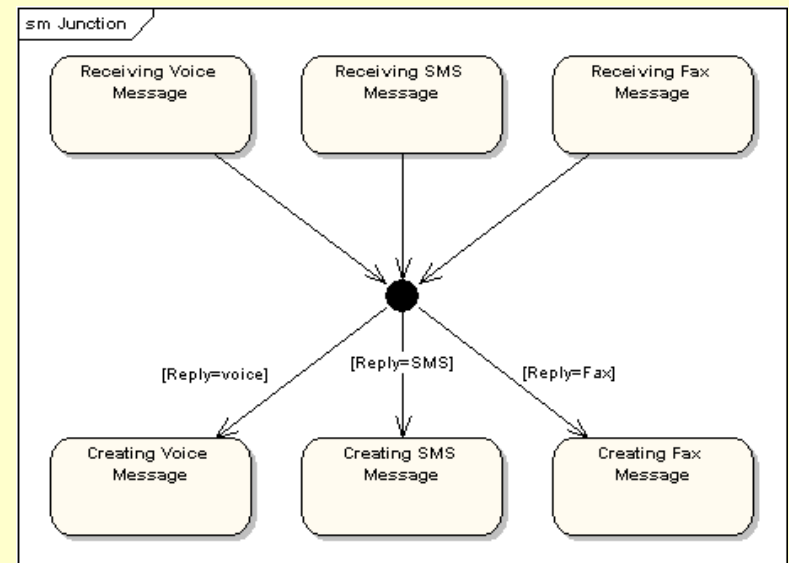
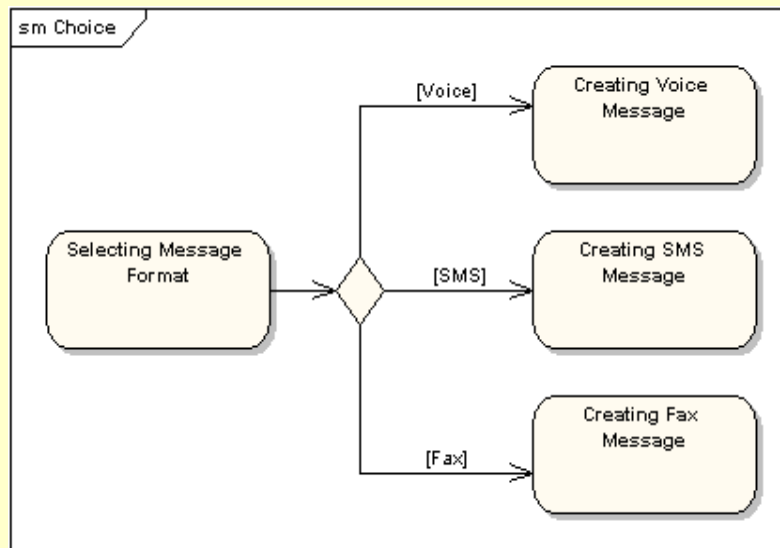
# DIAGRAMA de Estados

También pueden crearse estados compuestos



# DIAGRAMA de Estados

Bifurcaciones entre estados o puntos de unión



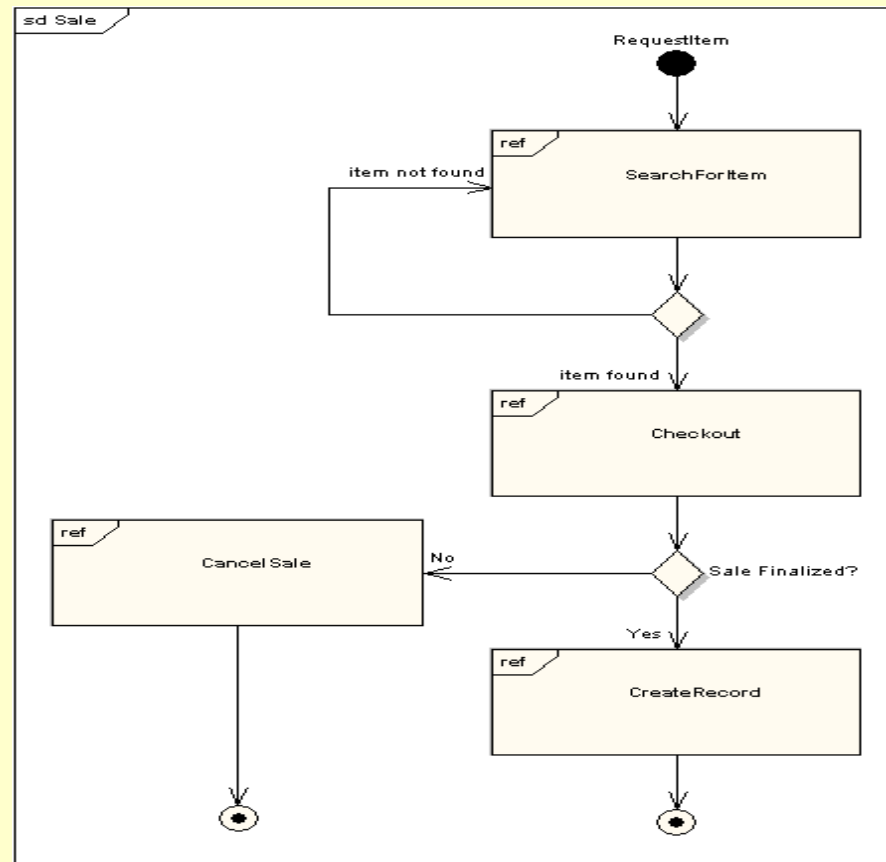


# DIAGRAMAS Dinámicos

DIAGRAMA DE INTERACCIÓN .- Fusiona los diagramas de actividad y secuencia, al ser sus nodos REFERENCIAS a diagramas de actividad, que son secuenciados

Su gramática es igual la de los diagramas de actividad.

# DIAG de INTERACCIÓN

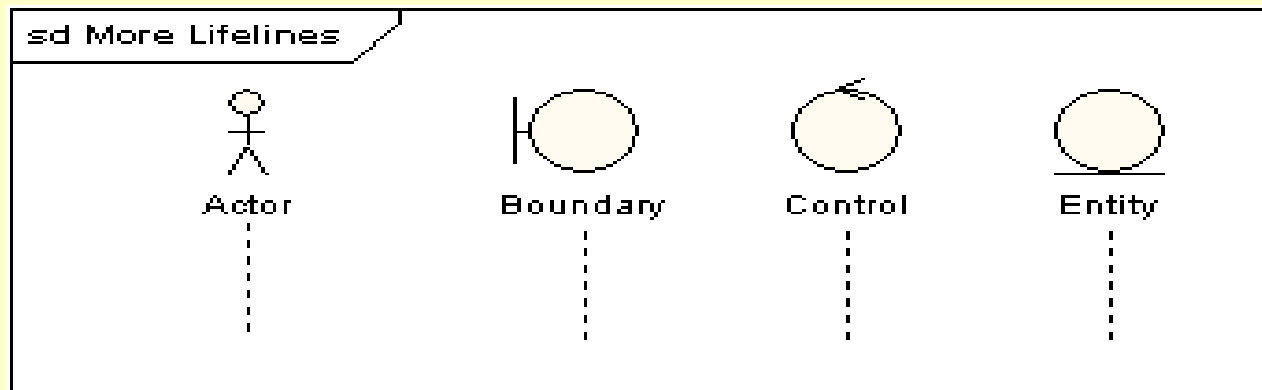


# DIAGRAMAS Dinámicos

DIAGRAMA DE COLABORACIÓN O COMUNICACIÓN .- Parecido a los diagramas de secuencia pero focalizados en la relación establecida entre los objetos (omitiendo el aspecto temporal de las relaciones y procesos)

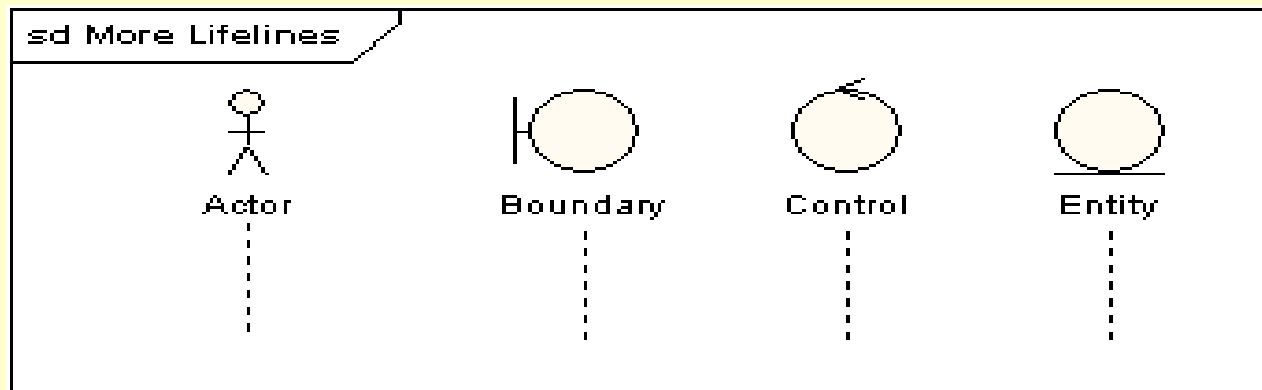
# DIAG de Colaboración

Puedo incluir diferentes tipos de entidad/ actores, siendo más específicos en pro de una mayor expresividad (esto también ocurría en los SEC )



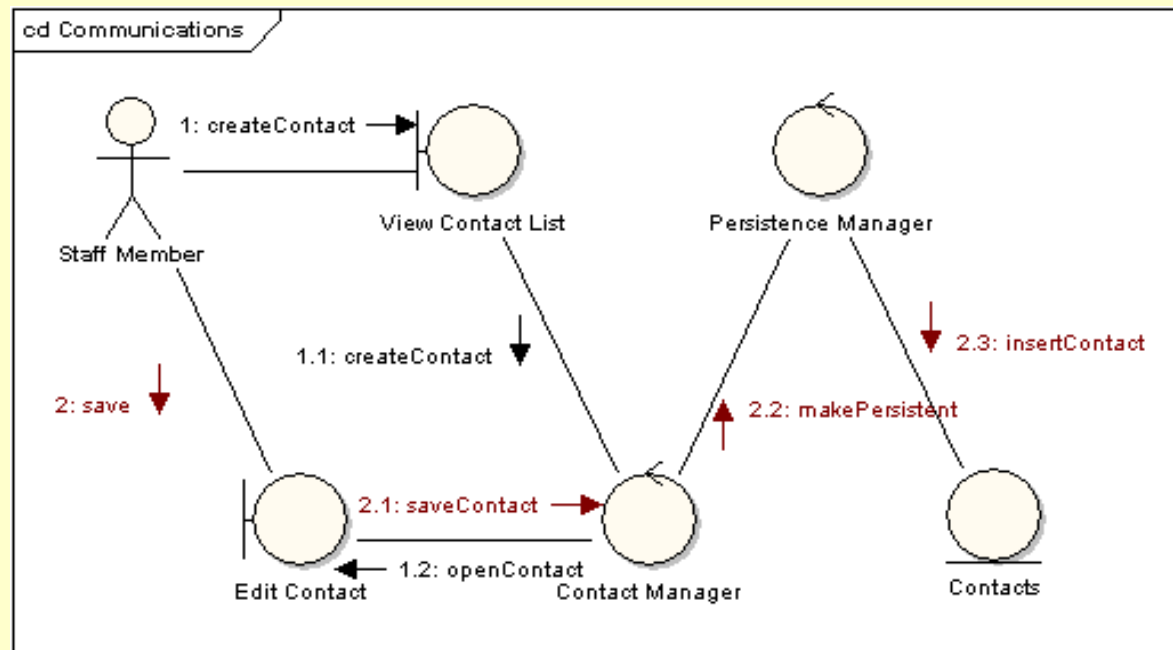
# DIAG de Colaboración

Puedo incluir diferentes tipos de entidad/ actores, siendo más específicos en pro de una mayor expresividad (esto también ocurría en los SEC )



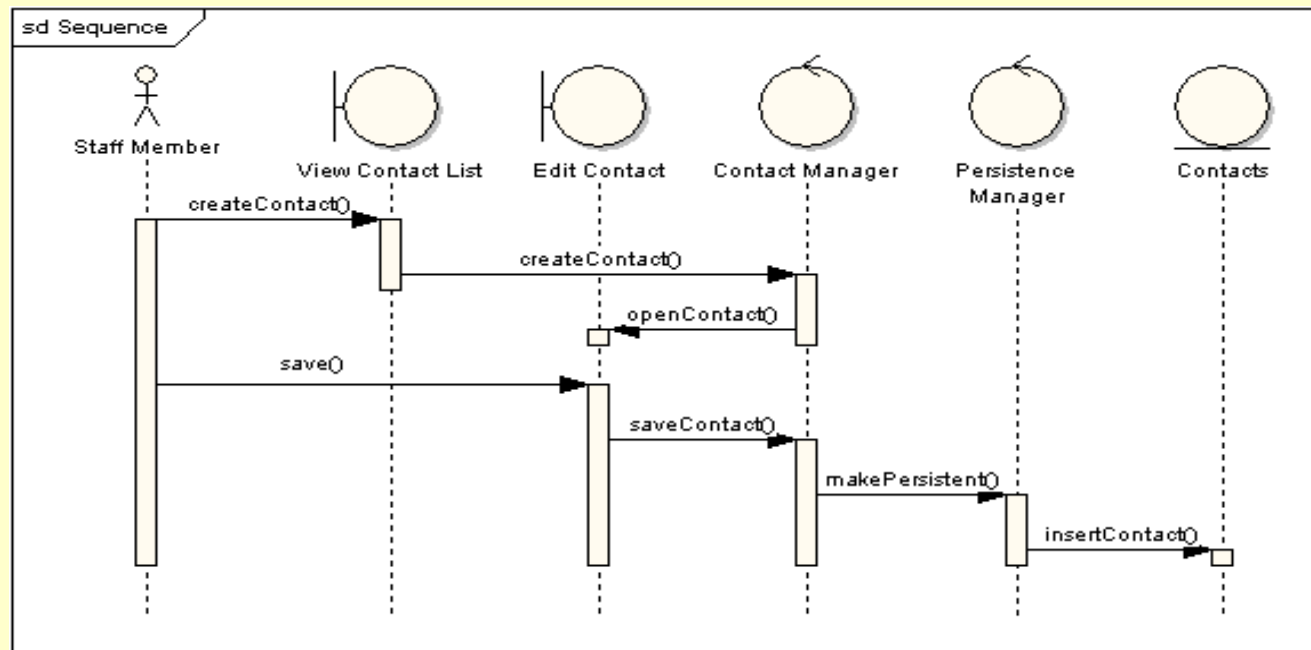
# DIAG de Colaboración

El siguiente diagrama de colaboración



# DIAG de Colaboración

Es análogo a este de secuencia



# DIAGRAMAS

DIAGRAMA DE COLABORACIÓN.- Diagrama de interacción que resalta la organización estructural de los elementos que envían y reciben mensajes

Colaboración y Secuencia son *isomorfos*, por lo que se pueden transformar uno partiendo del otro y viceversa



# ARQUITECTURA

La arquitectura tiene que ver con la estructura, el comportamiento, el uso la funcionalidad el rendimiento, la capacidad de adaptación, la reutilización, la capacidad de ser comprendido, las restricciones tecnológicas y presupuestarias, así como aspectos estéticos. Por ello, observamos mejor la arquitectura de un sistema si dividimos su contemplación e vistas

# VISTAS

De diseño: Clases, interfaces y colaboraciones

De procesos: Funcionamiento, hilos y procesos

De implementación: Componentes, archivos

De despliegue: Nodos que forman la tipología hw

De casos de uso: Percepción del sistema por usuarios finales, analistas y probadores

# ELEMENTOS ESTÁNDAR

Hay mecanismos extensibles que permiten que UML se adapte y/o crezca según las necesidades de un entorno

ESTEREOTIPOS

VALORES ETIQUETADOS

RESTRICCIONES

# ELEMENTOS ESTÁNDAR

## ESTEREOTIPOS

Actor – Aplicado a Clase, define un rol

Create – Aplicado a mensaje, especifica que el objeto destino es creado por el evento/mensaje

Executable – Aplicado a componente, especifica que un componente puede ejecutarse en un nodo

Table – Aplicado a componente, especifica que es una tabla de una base de datos

# ELEMENTOS ESTÁNDAR

## VALORES ETIQUETADOS

{NOMBRE ETIQUETA = VALOR}, bajo el elemento referido. Aporta valor documental, pudiendo emplearse como nombre etiqueta:

Documentation.- Especifica un comentario del elemento al que se asocia

# ELEMENTOS ESTÁNDAR

Location.- Especifica el nombre o componente donde reside un elemento

Persistence.- Especifica si la instancia se mantiene después de que finalice el proceso que la creo

Semantics: Especifica el significado de una clase u operación

# ELEMENTOS ESTÁNDAR

## RESTRICCIONES

La restricción, extiende la semántica de un bloque.

Normalmente se representa entre llaves {} y estos son algunos de sus valores más representativos:

{new} Especifica que el elemento se crea durante la interacción que lo contiene

# ELEMENTOS ESTÁNDAR

`{complete}` Especifica que todos los hijos de una generalización han sido detallados y no se admiten nuevos subtipos

`{or}` Especifica, sobre un conjunto de asociaciones que exactamente se manifiesta una para cada objeto asociado



# ELEMENTOS ESTÁNDAR

`{implicit}` Especifica que una relación es sólo conceptual y no empírica o manifiesta

`{destroyed}` Especifica, sobre una instancia o enlace, que es destruido antes de terminarse la interacción que lo contiene

# Enterprise Architect

Actualmente, la versión estable de producción es la 13

Ofrece un alto grado de adaptación al estándar UML

Añade algunos diagramas menos usuales

Ofrece varias versiones (ultimate, la más completa)

Cuenta con herramientas, plugins de integración para IDE's como Eclipse y Visual Studio agrupadas bajo el concepto Model Driven Generation (MDG)

# TECNOLOGÍA Adyacente

## RUP Proceso Unificado de Rational

Si bien UML se muestra independiente del proceso, RUP es un enfoque sobre el ciclo de vida software que se adapta especialmente bien a UML

RUP define fases, iteraciones, flujos, artefactos y modelos.

# TECNOLOGÍA Adyacente

## BMP Bussines Project Management

BMP ofrece una perspectiva más funcional, focalizada en los procesos, con mayor independencia de los detalles de implementación.

Similar a un Diagrama de Actividad de UML, está más orientada a analistas de negocio que a desarrolladores

# Estructura EA

El elemento raíz en la perspectiva de proyecto es llamado el MODELO, cuya cardinalidad es 1:N por cada proyecto o sistema modelado

En él, tendré anidados Views y Packages. Éstos a su vez contendrán los diagramas, elementos, relaciones y metainformación asociada

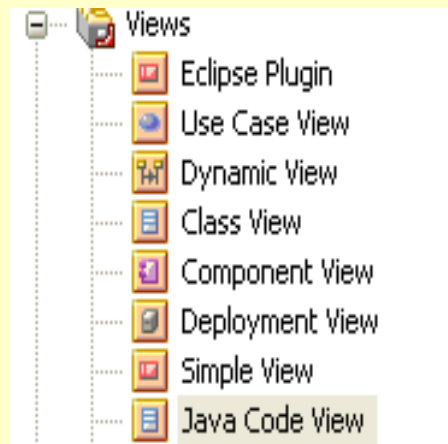
# Estructura EA

Dependiendo de la complejidad del Sistema, puedo tener un solo modelo o varios modelos que expliquen parte del Sistema (normalmente Modelo Raíz – root node)

Puedo tener varios elementos raíz o modelos compuestos por otros (habitual en un Sistema Distribuido)

# EA Views

Las vistas son también paquetes, pero con un significado conceptual añadido



# EA Views

Las VISTAS, sirven para estructurar el proyecto al más alto nivel. Existen 6 tipos predefinidos

Use Case View – Diagramas de casos de uso y Análisis (DA simplificado)

Dynamic View – Diagrama de actividad, de comunicación, de secuencia y estados



# EA Views

Class View – Diagramas de clases, Code Engineering y Modelos de datos

Component View - Diagrama de Componentes

Deployment View -Diagrama de Despliegue

Simple View – Tipo de Vista personalizado, para modelar aspectos particulares de mi proyecto

# EA Packages

Los paquetes, son básicamente meros contenedores que agrupan elementos, diagramas y otros paquetes



Su operativa básica no excede de crear, renombrar, borrar copiar o mostrar el listado de su contenido

Puedo definir uno siempre que me interese

# EA Packages

Class View – Diagramas de clases, Code Engineering y Modelos de datos

Component View - Diagrama de Componentes

Deployment View -Diagrama de Despliegue

Simple View – Tipo de Vista personalizado, para modelar aspectos particulares de mi proyecto

# EA MODELView

EA ofrece un modelo de vistas adecuado a cada usuario o perfil que participa en el grupo, mediante la opción Start → Explore → Model Views.

# EA MODELView

Model Views – Almacenadas en el modelo y visible a todos los usuarios

My Views – Almacenadas localmente sólo visibles para el usuario de la máquina donde se definen

Technology-defined Views - Basados en MDG y de solo lectura

# EA MODELView

En el interior de un ModelView, crearé carpetas como elementos raíz y en su interior:

- Search View Folders .- Vistas que defino a modo de búsqueda con un criterio de actualización y cuya ejecución me ofrece una conjunto parcial de diagramas o información del modelo. Si cambio de modelo, se actualiza automáticamente

# EA MODELView

- Favorites .- Para los items más habituales. Basta con arrastrar del Project Browser los elementos deseados y soltarlos en la carpeta de favoritos
- SlideShowFolder .- Ideal para presentaciones, agrego a esta carpeta los elementos y diagramas que quiero que se reproduzcan a modo de vídeo.

# IMPORTANDO Código

EA incorpora ciertos mecanismos de ingeniería inversa para recuperar las relaciones explícitas de un paquete o código fuente y mostrarla en forma de clase.

Basta seleccionar desde la vista deseada botón derecho Code Engineering → Import Source Folder



# MIDIENDO EL ESFUERZO

EA incorpora una medición de proyecto mediante la estimación aplicada a los casos de uso.

Esta estimación, está basada en el modelo teórico de Karner, que define una escala de complejidad aplicable a los escenarios / caso de uso previstos.

# USE CASE METRICS

La complejidad que puedo asociar a cada nodo (uso) de de diagrama, consiste en:

- Easy – Procesos sencillos (menos de 5 clases)
- Medium – Procesos de 5 a 10 clases
- Hard – Procesos de más de 10 clases

# USE CASE METRICS

Para obtener el informe botón derecho sobre el paquete que contiene mi diagrama con estimaciones,

Documentation → Package Metrics

Y obtengo el baremo de horas y coste. Puedo exportar con Report, un documento rtf

# USE CASE METRICS

## PARÁMETROS:

UCCP- Puntos de casos de uso sin ajustar

TCF - Technica Complexity Factor

ECF - Enviroment Complexity Factor

UCP -  $UCCP * TFC * EFC$

Horas – Cuántas horas es un UCP

Coste – Cuánto dinero cuesta una hora de trabajo

# USE CASE METRICS

Para definir TFC y ECF vamos al menú principal  
Configure Project Types Estimation Factors

TCF Quedan definidos la lista de dificultades técnicas valoradas mediante un peso y un valor

ECF Quedan definidos la lista de dificultades del entorno igualmente en escala de peso valor

# USE CASE METRICS

No hay una ciencia exacta para determinar el esfuerzo de un proyecto. Será la experiencia y el contraste de la estimación con el coste real lo que lleve a incluir factores de corrección y parámetros iniciales con mayor precisión.

De hecho, una práctica aconsejable a la hora de definir los valores de un proyecto, es basarse en otro similar

# EA Requirements

EA incorpora un subtipo de diagrama, en la sección Extended (ni Behavioral ni Structural) denominado Requirements y pensado para definir textualmente los requisitos del Sistema o Módulo, de modo que puedan ser definidos, revisados y editados durante el ciclo de vida del software

# EA Requirements

EA incorpora un subtipo de diagrama, en la sección Extended (ni Behavioral ni Structural) denominado Requirements y pensado para definir textualmente los requisitos del Sistema o Módulo, de modo que puedan ser definidos, revisados y editados durante el ciclo de vida del software



# EA Requirements

Los tipos de requisitos previstos son:

Funcionales, De Negocio, De Usuario, No Funcionales, De Arquitectura, Implementación, Regulatorio y Seguridad

# EA Requirements

Es siempre conveniente emplear una nomenclatura que ayude a definir claramente los requisitos e identificarlos, creando una lista.

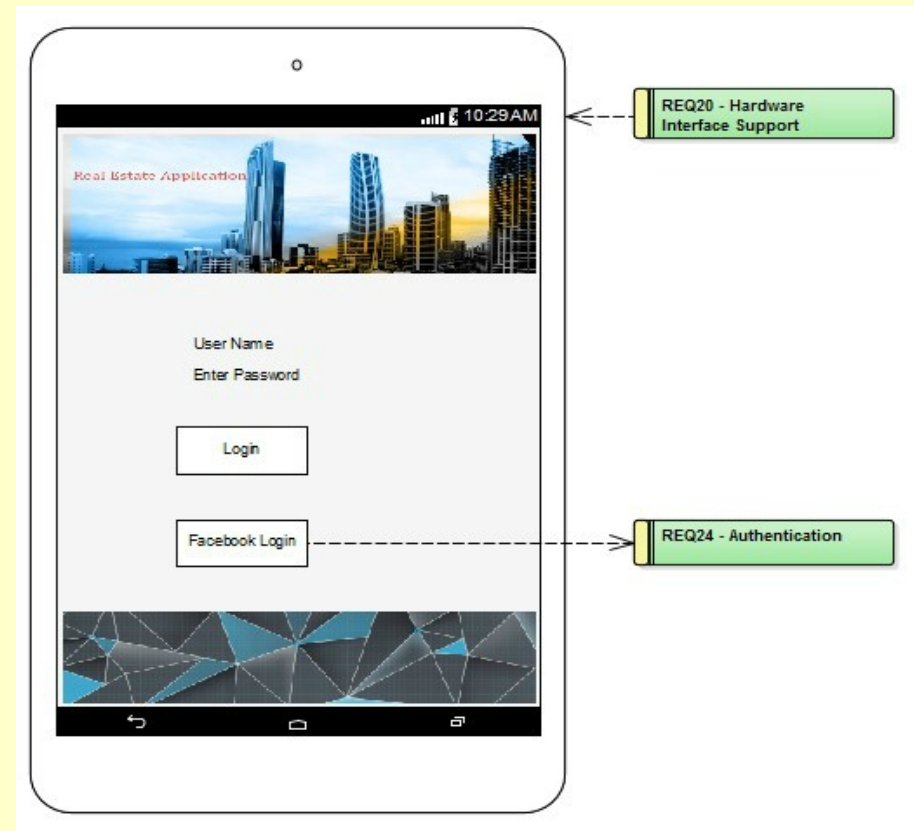
R1 – SPRING 4.2 (arquitectura)

...

R11 – Menores de 18 no permitidos (normativo)

# EA Requirements

Resulta especialmente útil, arrastrar luego los elementos de tipo requisito al diagrama o elemento donde aplica, para facilitar el cumplimiento y la implementación



# EA Discussions

Se puede crear debates o diálogos textuales y asociados a cada Paquete, Vista, Elemento o incluso Módulo de forma que se permita la interlocución entre miembros del equipo, cliente y proveedor o para anotaciones

# EA Discussions

Desde el Menú principal Construct → Discussion  
→ Discussion Element despliega una ventana que puedo ubicar en varios sitios del proyecto.

Cuando seleccione un elemento del explorador, automáticamente se incluye una entrada en la ventana desplegada y me permite iniciar una conversación o anotación

# EA TEAMReview

Si bien las Discussions se refieren a un elemento particular, EA proporciona la facilidad de crear una especie de foro-debate para los usuarios de un proyecto: el Team Review

Construct → Project → Team Review

# EA TEAMReview

En esta ventana, la información queda jerarquizada por tres niveles:

CATEGORÍA

ASUNTO

DOCUMENTOS Y COMENTARIOS

# EA TEAMReview

Al editar un documento o generar una respuesta, y visualizarse, se muestra la última edición, indicando fecha, hora y autor.

Por defecto, los nuevos mensajes se me cargan como no leídos.



# EA TEAMReview

Al trabajar con los documentos y edita texto, puedo enlazar parte de esos textos, a algún elemento existente en el modelo.

Basta con que seleccione el texto, y con el botón derecho Create → Link to existing element

# EA Project Management

EA provee de herramientas de seguimiento y medición de esfuerzo.

La perspectiva Gantt, es aplicable a cualquier diagrama y resulta de interés a la hora de asignar recursos y mantener una planificación temporal

# EA Project Management

## DEFINIR EL PERSONAL

El paso necesario es definir los colaboradores y los roles. Esto se puede hacer desde

Configure → Reference Data → Project Types → People

# EA Project Management

## PLANIFICAR CADA DIAGRAMA

Por cada Diagrama, puedo seleccionar con el botón derecho → View Diagram As → View as Gantt.

Para cada elemento del diagrama, puedo seleccionar recurso y tiempos

# EA Project Management

Automáticamente, EA actualiza la vista Gantt para todo el proyecto con la información introducida, agrupando la información por recursos.

Desing → View Gantt

# EA Navegación UI

Para simular la navegación por el menú de una interfaz gráfica, EA provee un mecanismo para enlazar elementos visuales.

Basta seleccionar un elemento y botón derecho, New Child Diagram → Select Composite Diagram y elegir diagrama deseado

# EA Navegación UI

De este modo, si defino diferentes pantallas y represento cada una en organigramas independientes, puedo enlazar un botón a otro diagrama.

# TIPS útiles

Fijar el toolbox particular boton derecho sobre ocualquier elemento pin in tool box

FIND → In Project Browser



# TIPS útiles

Si estoy en un diagrama y quiero localizar su posición en el panel principal (Project Explorer); basta con seleccionar el elemento y con el botón derecho

FIND → In Project Browser