

Con la colaboración de:

ORACLE®

IBM®



Introducción a UML

Unified Modeling Language

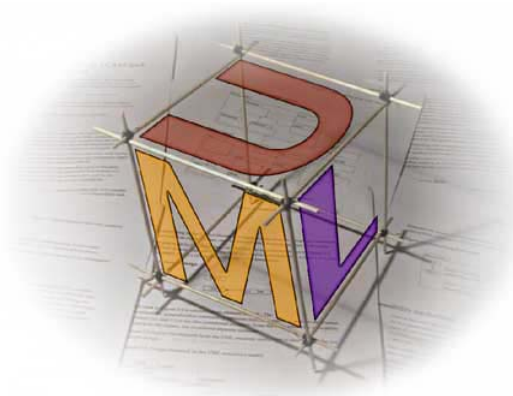
Josep Vilalta Marzo

www.vico.org

Universitat Internacional de Catalunya

Qué es UML

Unified Modeling Language “UML”, es una notación patrocinada por el Object Management Group (OMG), que se ha convertido en un estándar para definir, organizar y visualizar los elementos que configuran la arquitectura de un sistema.



Qué es UML

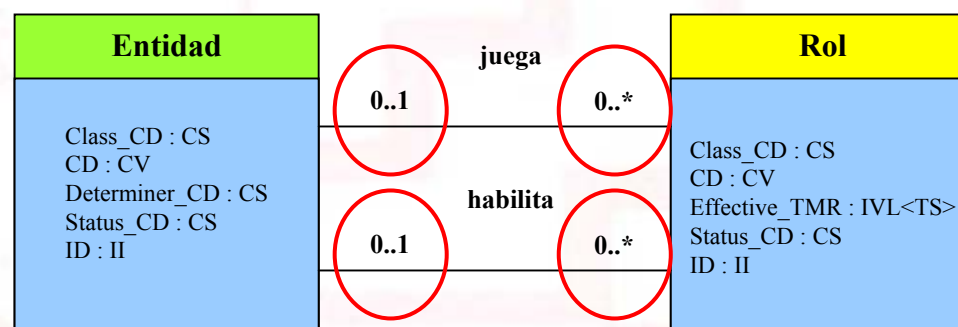
A través de la notación UML podemos comunicar y compartir el conocimiento de una arquitectura gracias a la combinación simultánea de distintas perspectivas:

- **Definir.- Fijar, determinar, especificar, explicar un concepto a través de sus atributos distintivos. Señalar sus límites y dar una idea exacta de lo que es esencial y de lo que es circunstancial en su ciclo de vida.**



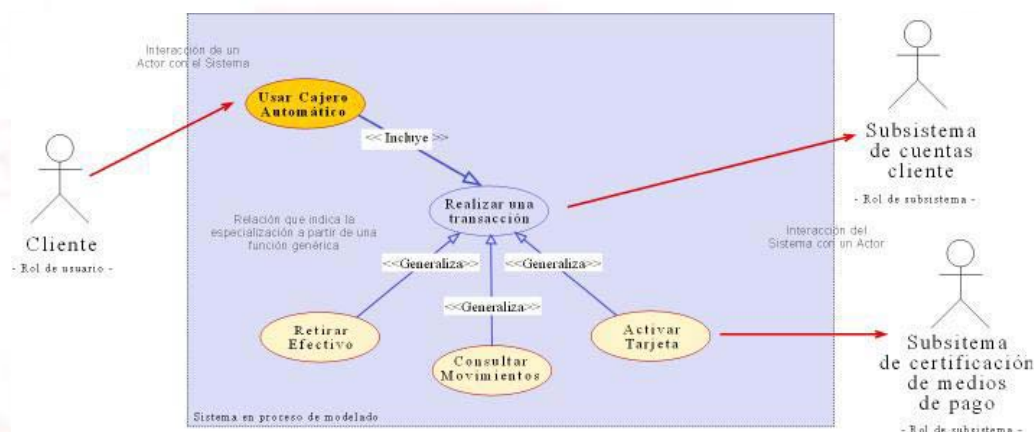
Qué es UML

- **Organizar.-** Establecer unos recursos, disponer un flujo de trabajo con un orden de responsabilidades y formalizar unas reglas de relación y actuación; todo ello orientado a conseguir el propósito de un proceso de negocio.



Qué es UML

- **Visualizar.-** Representar mediante artefactos y diagramas el contenido y la organización de los conceptos que configuran la naturaleza y complejidad de un sistema. Hacer visible su granularidad y entrelazamiento.



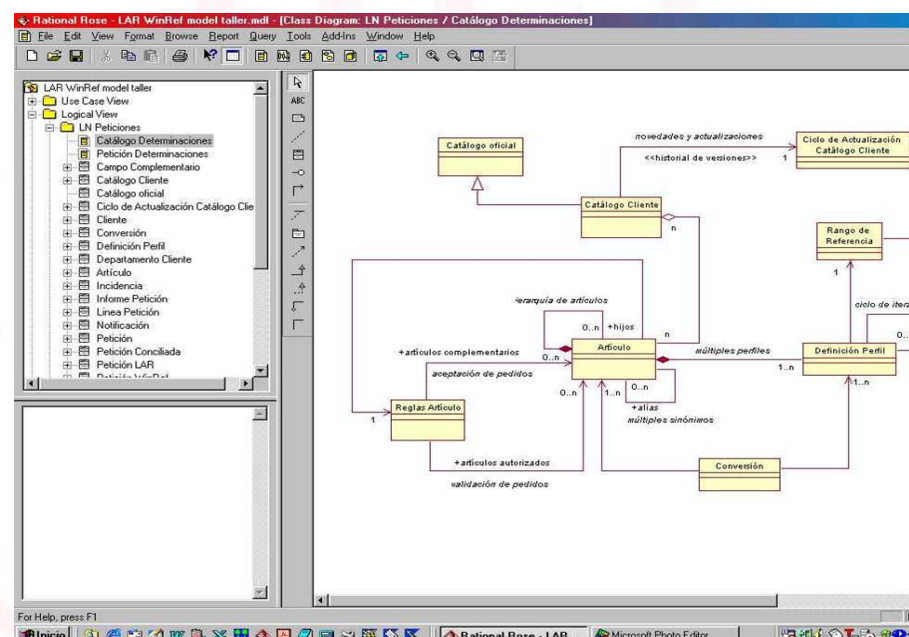
Qué es UML

- **Actuar.- Tomar decisiones de manera ágil y sistemática, siguiendo un método; éste a su vez, define el modo de actuar en base a la relación de un conjunto de actores, actividades, entregables y certificaciones posibles en un escenario concreto.**



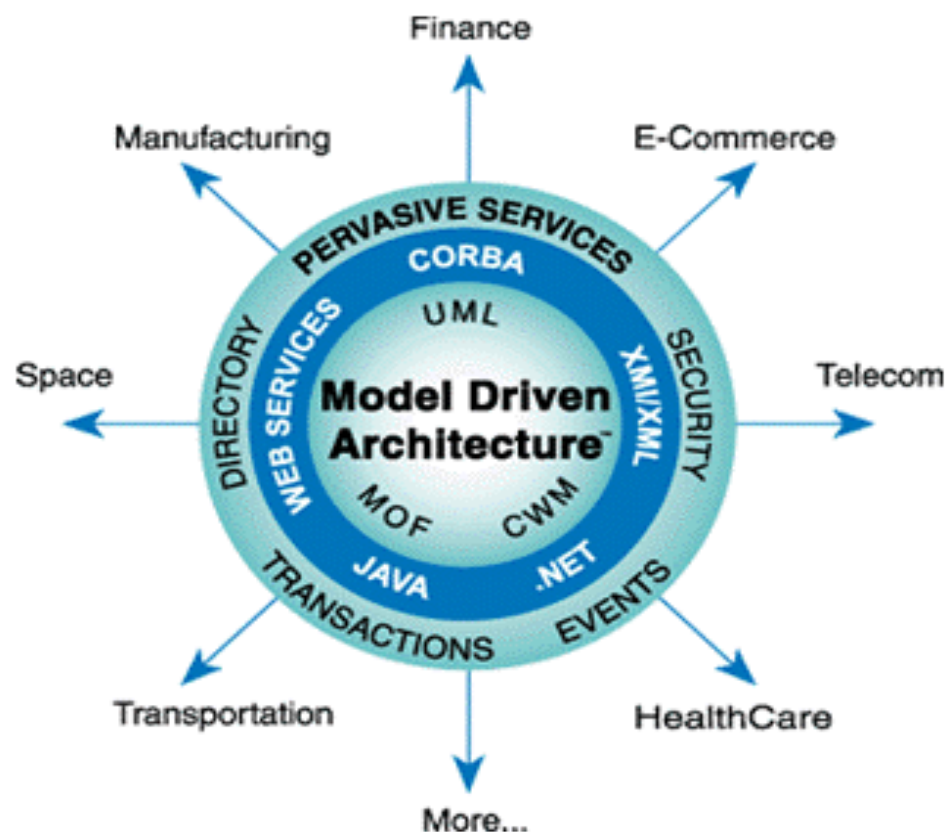
Qué es UML

- **Construir.- Realizar modelos que organizan la experiencia de los usuarios en patrones de conocimiento y procesar dichos modelos para generar una arquitectura de aplicaciones en base a componentes y procesos de negocio.**



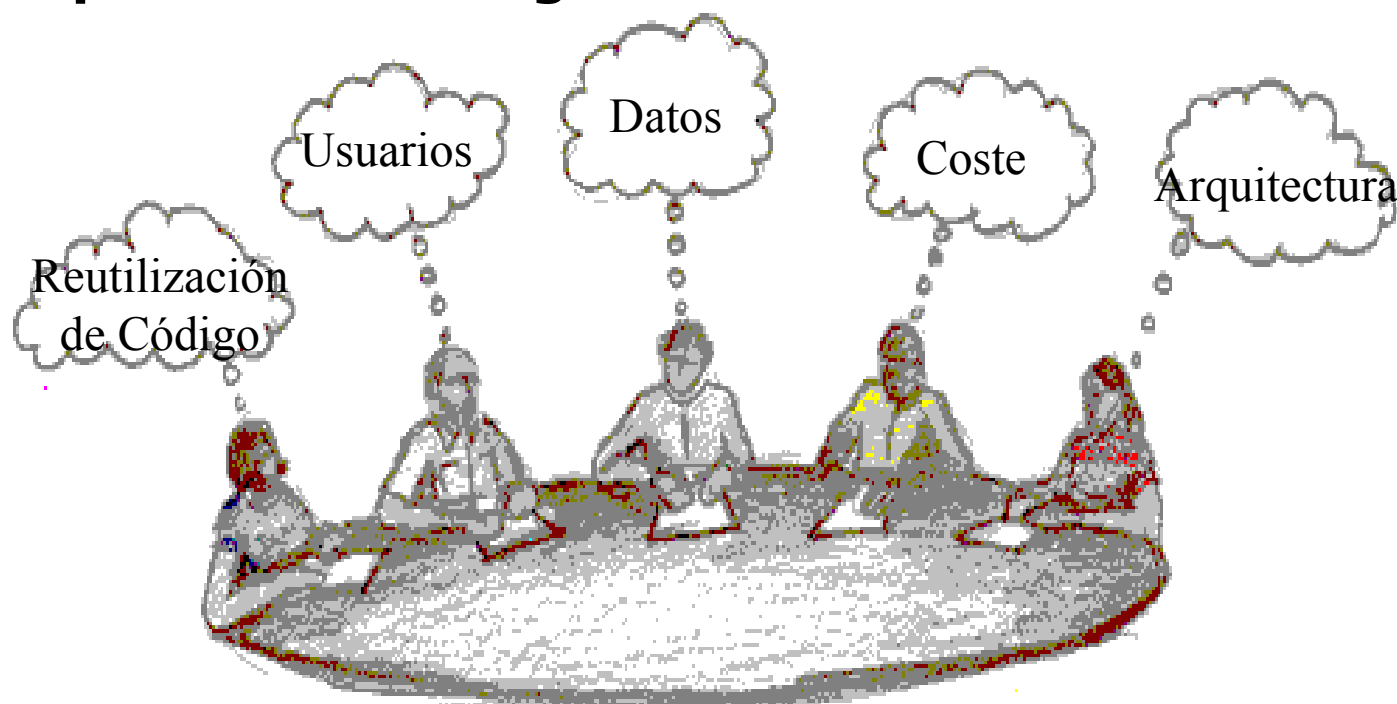
Qué es UML

- **Análisis y diseño independiente de plataforma**



Qué es UML

- **Documentar.-** Compartir el conocimiento con un vocabulario controlado que permite mantener la trazabilidad entre la concepción, la construcción, el despliegue, la implantación y la ejecución de los componentes de negocio.



Qué es UML

- **Certificar.- Comprobar de manera fehaciente que un entregable es coherente y completo en referencia al propósito para el que ha sido creado, y que es usable dentro de la cadena de valor de los Actores implicados e involucrados.**

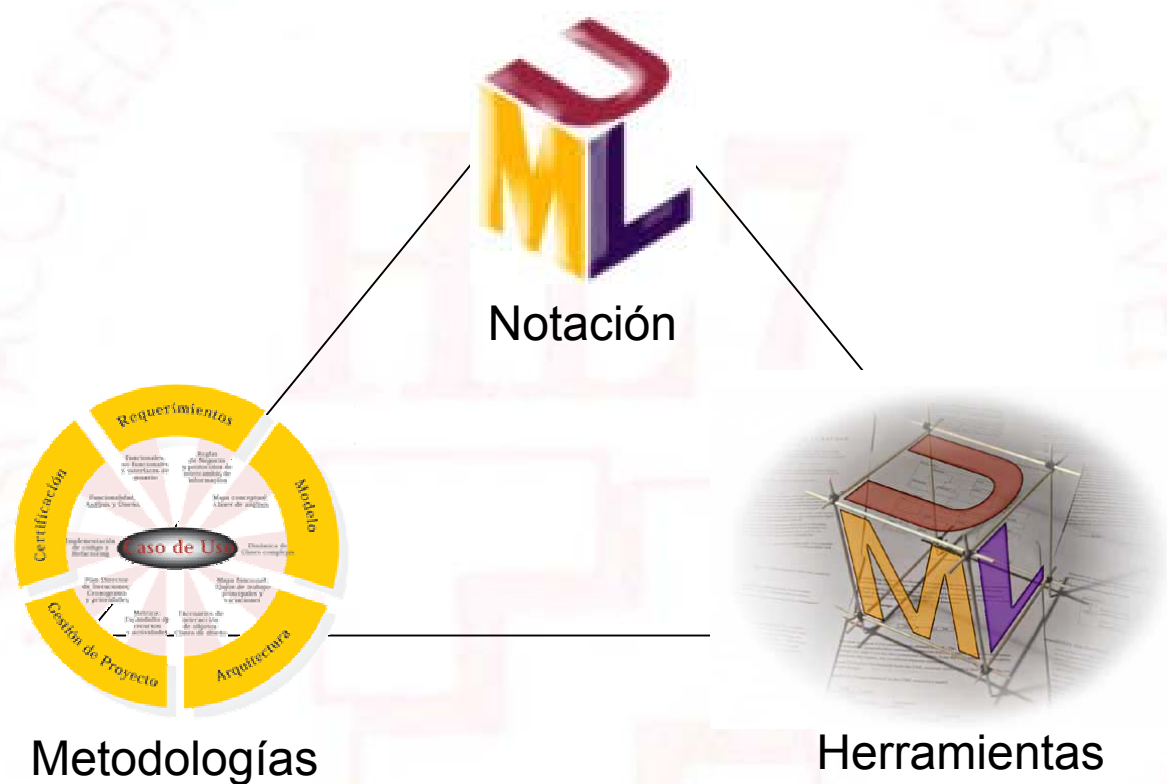


Qué es UML

El resultado, es una comprensión visual de las coordenadas **Espacio** – **Tiempo** – **Información**, que condicionan los dominios restringidos de una forma de vida organizativa.

- **Espacio**.- Existen unos OBJETOS que colaboran para cumplir un propósito.
- **Tiempo**.- Ocurren una serie de EVENTOS dentro de una secuencia acotada por escenarios.
- **Información**.- Los OBJETOS y EVENTOS producen HECHOS inteligibles, registrables y comunicables.

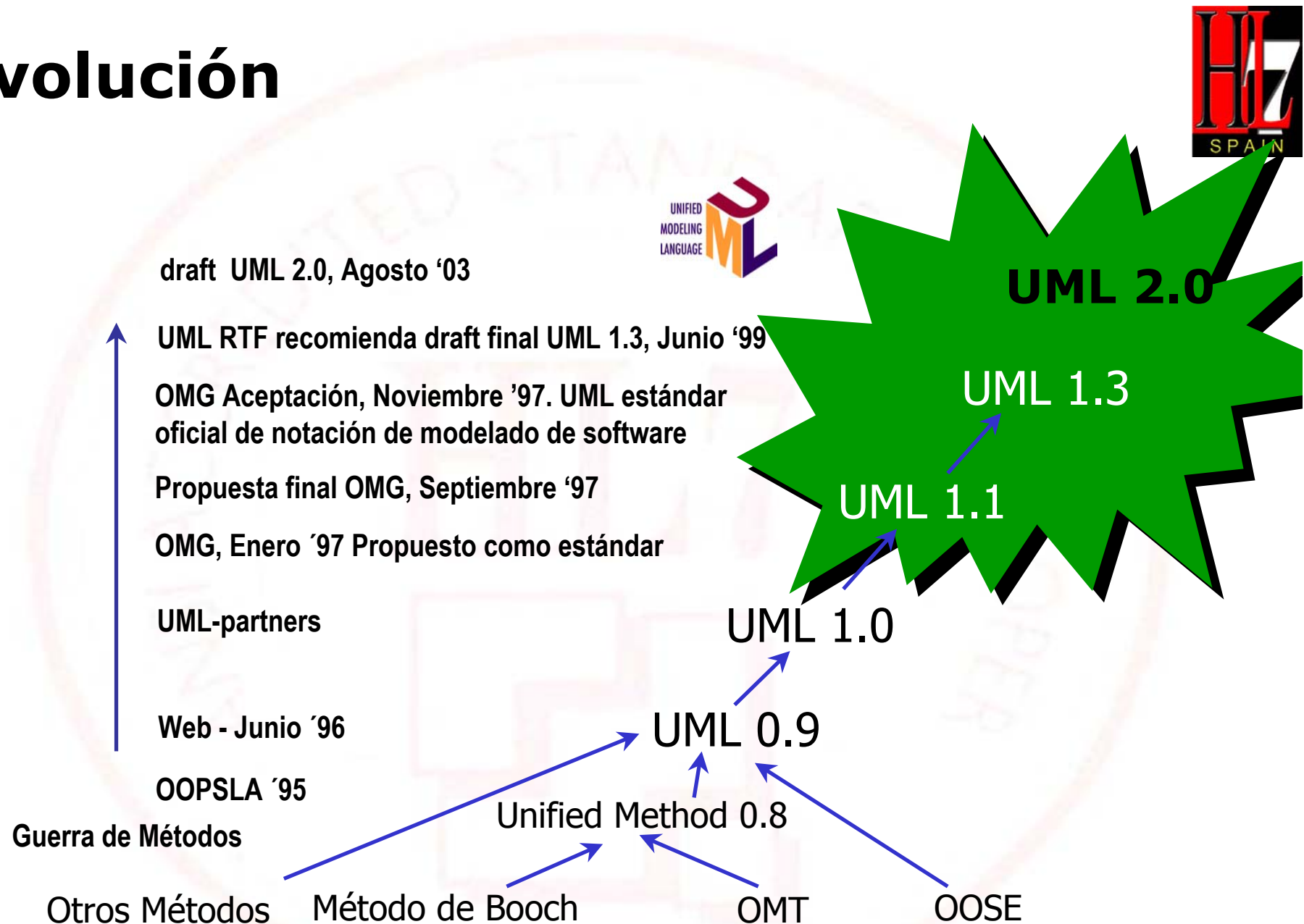
Qué es UML



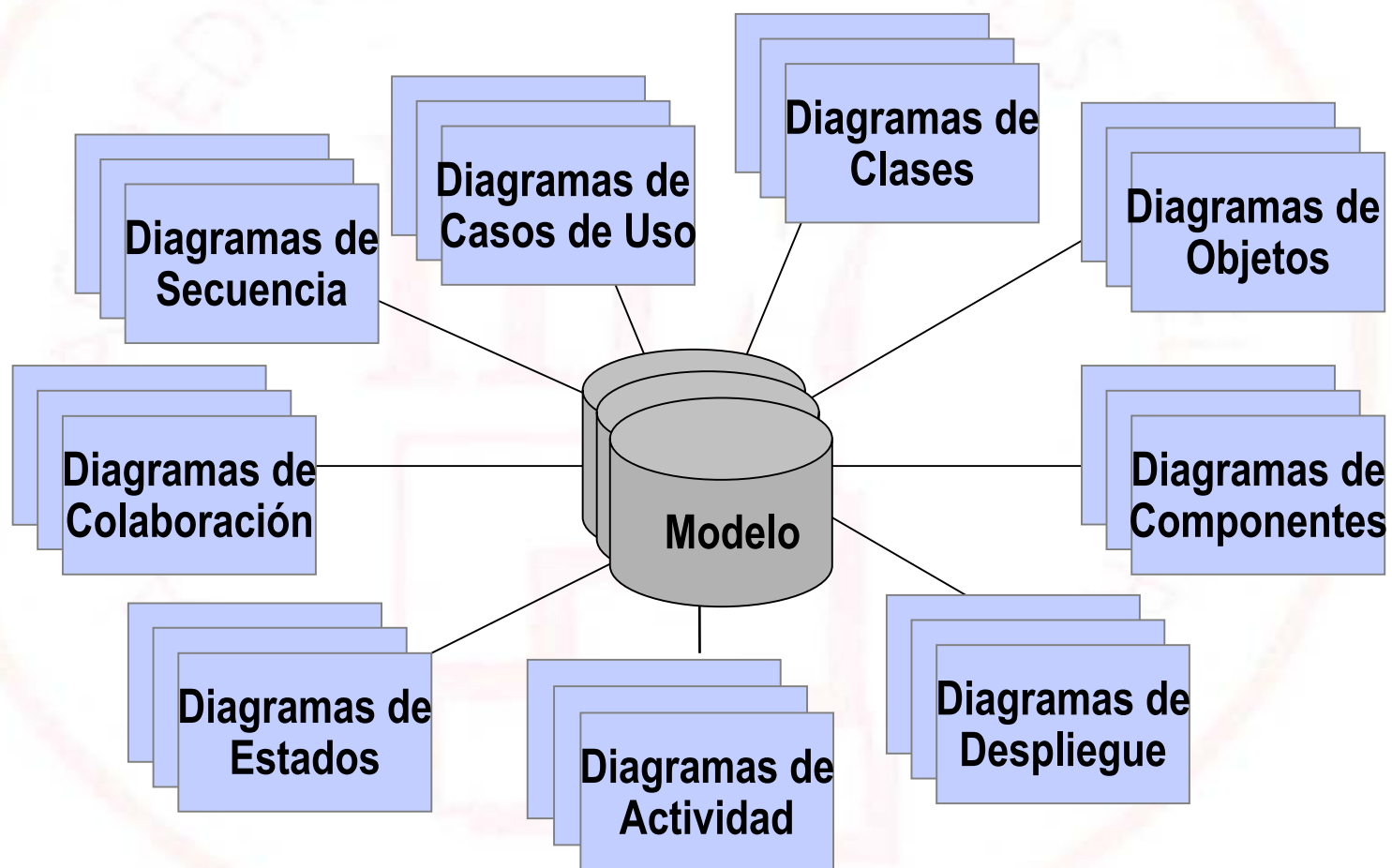
Cooperación y fusión de notaciones

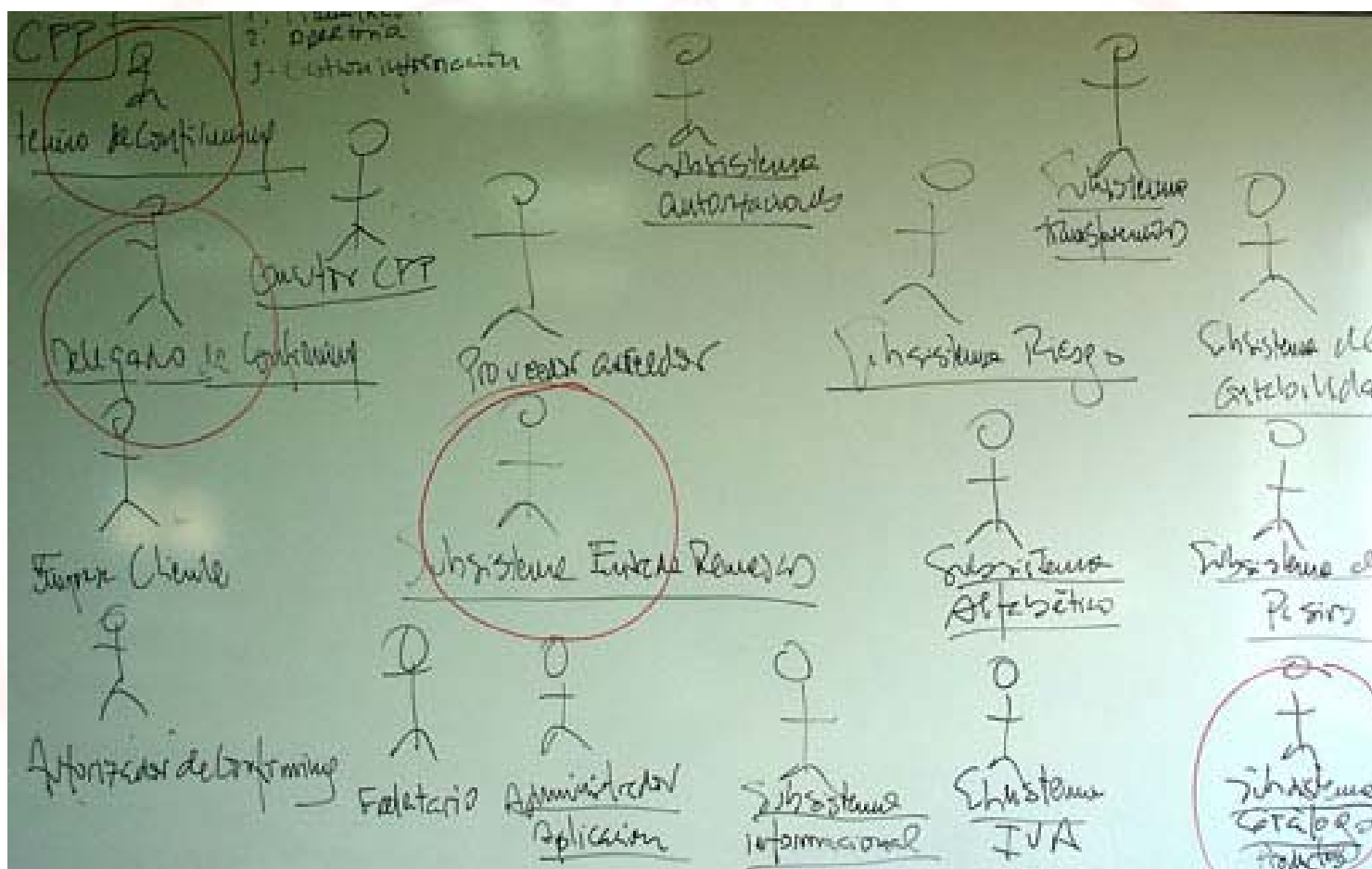


Evolución

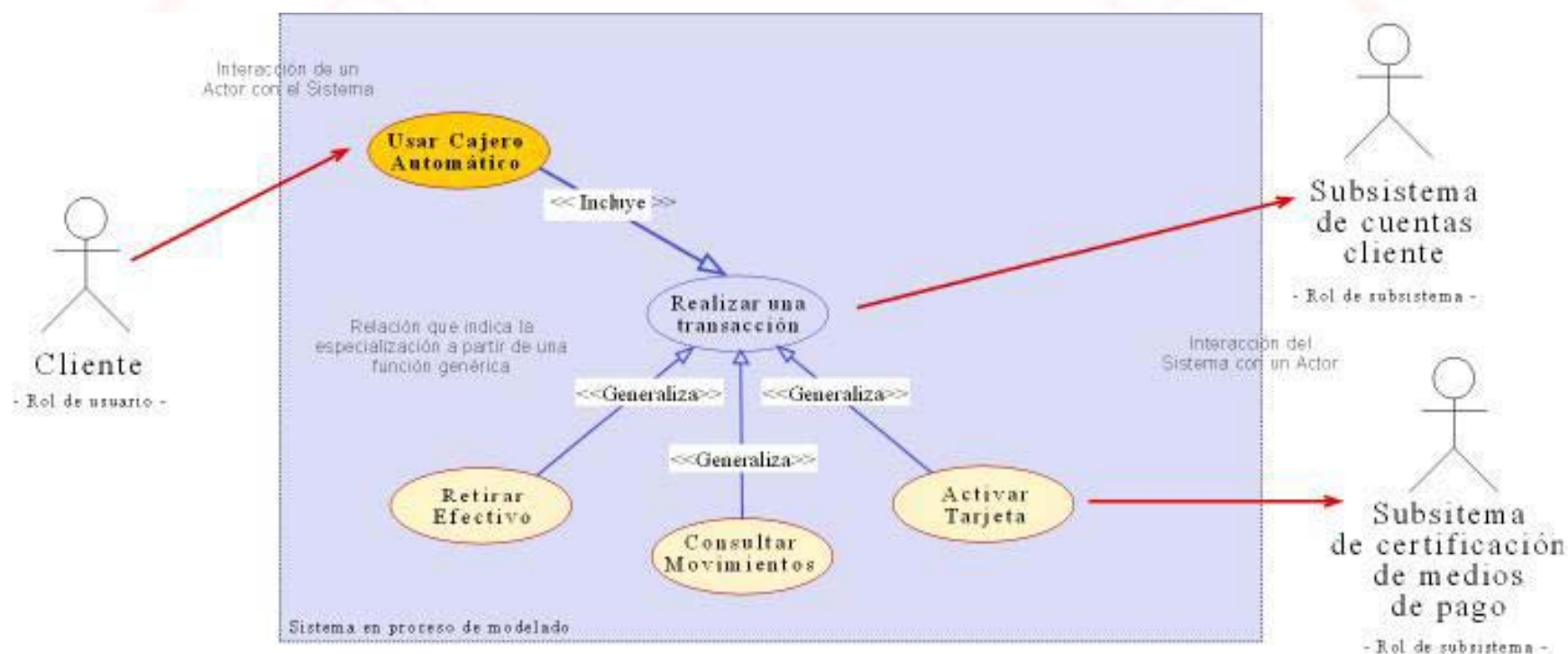


Múltiples perspectivas

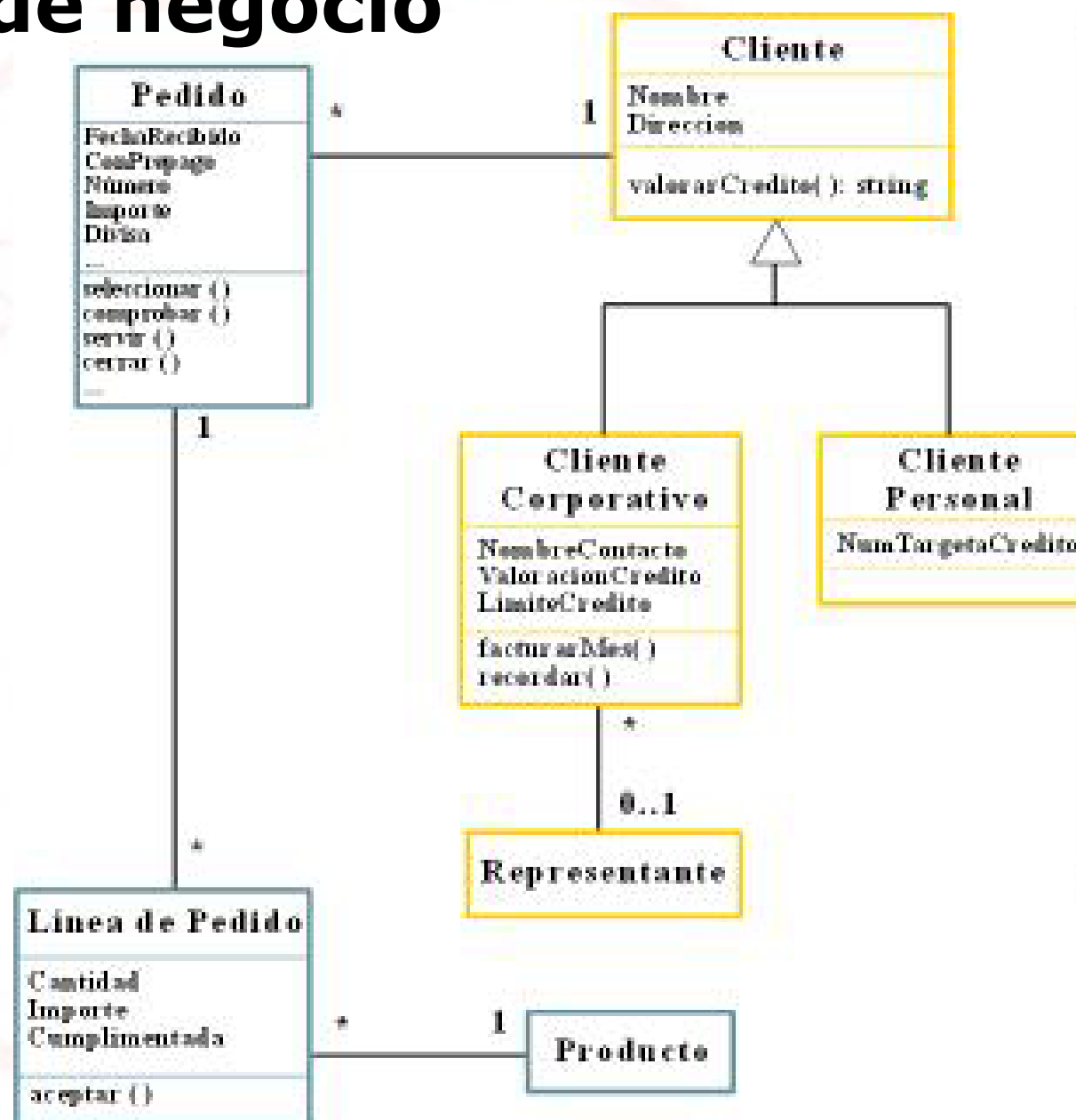




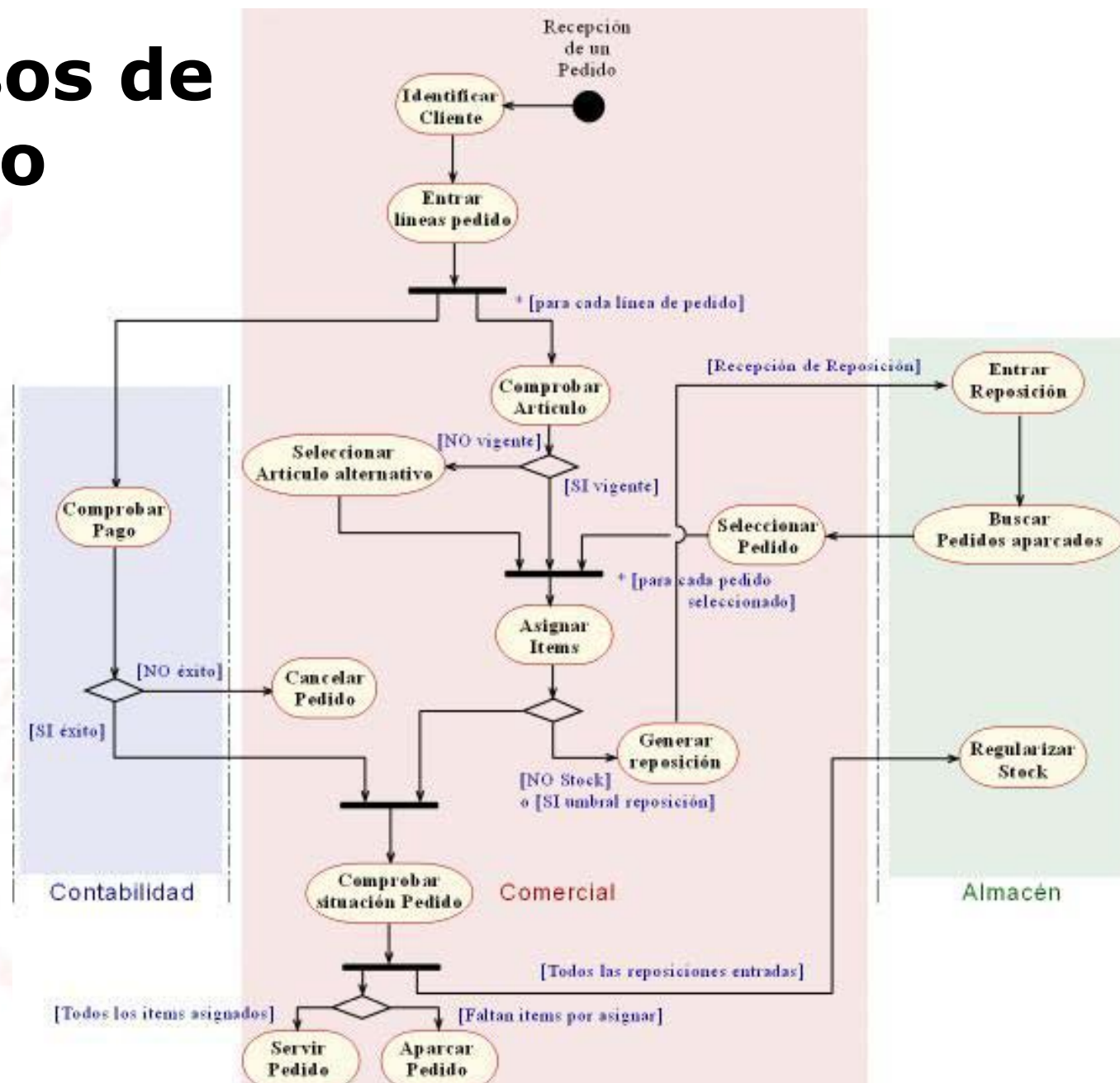
Escenarios de usabilidad



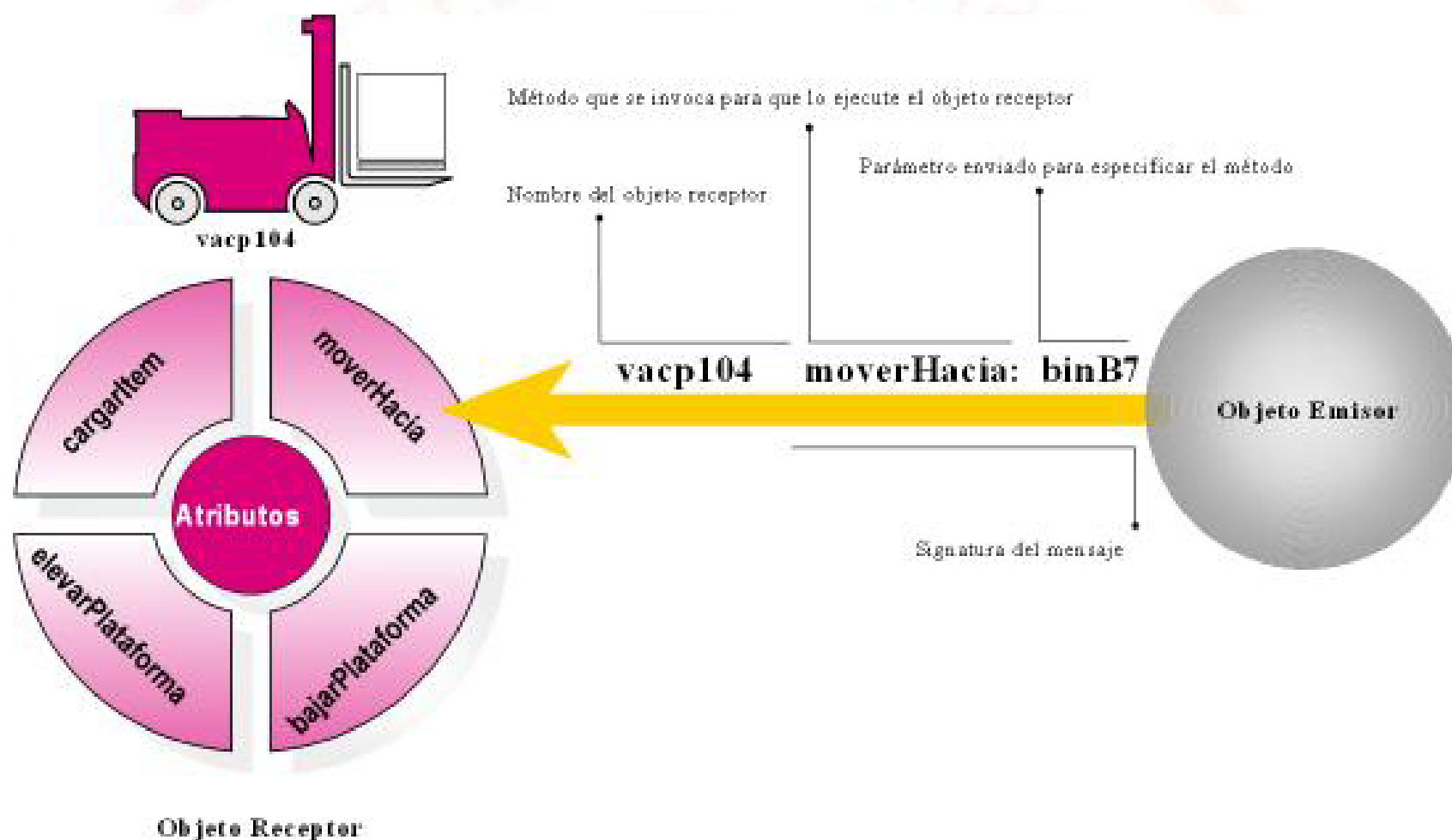
Modelos de negocio



Procesos de negocio



Orientación a objetos

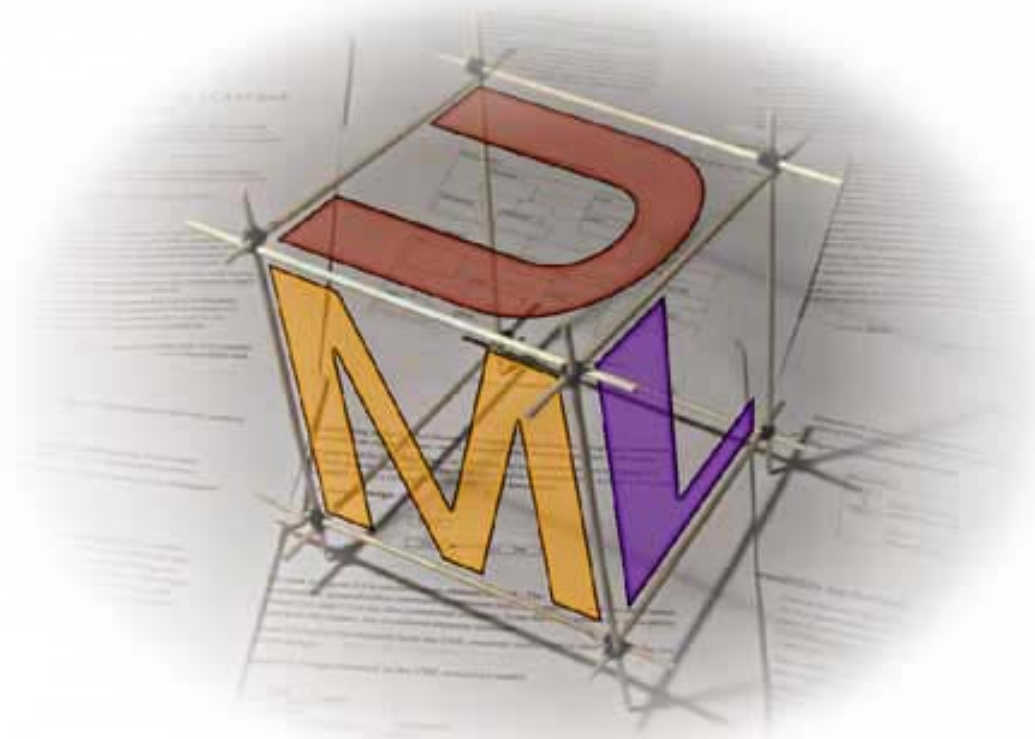


Metodologías asociadas



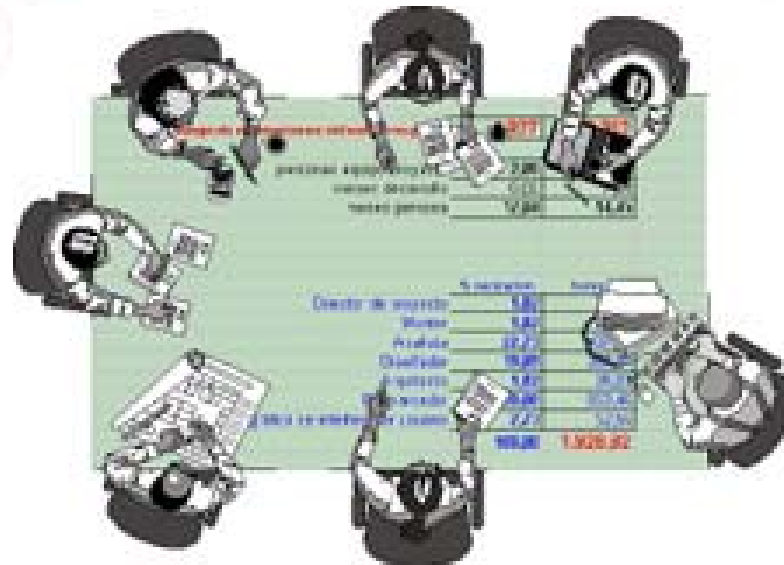
**Actuaciones, Entregables,
Participantes, Roles, Agentes
Plantillas de Proceso de Proyecto
Fases e hitos de Proyecto**

Herramientas de modelado



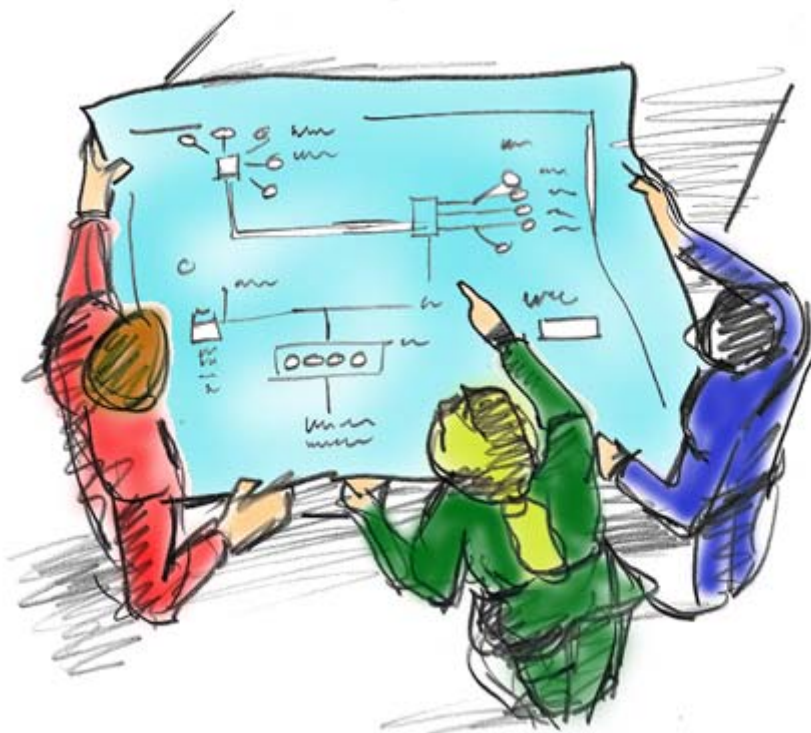
Desarrollo de software
Modelado de procesos de negocio
Dirección eficaz de proyectos

Desarrollo de software



Casting de roles
Repertorio de entregables UML
Casting de participantes
Esquema cooperativo

Modelado de procesos de negocio

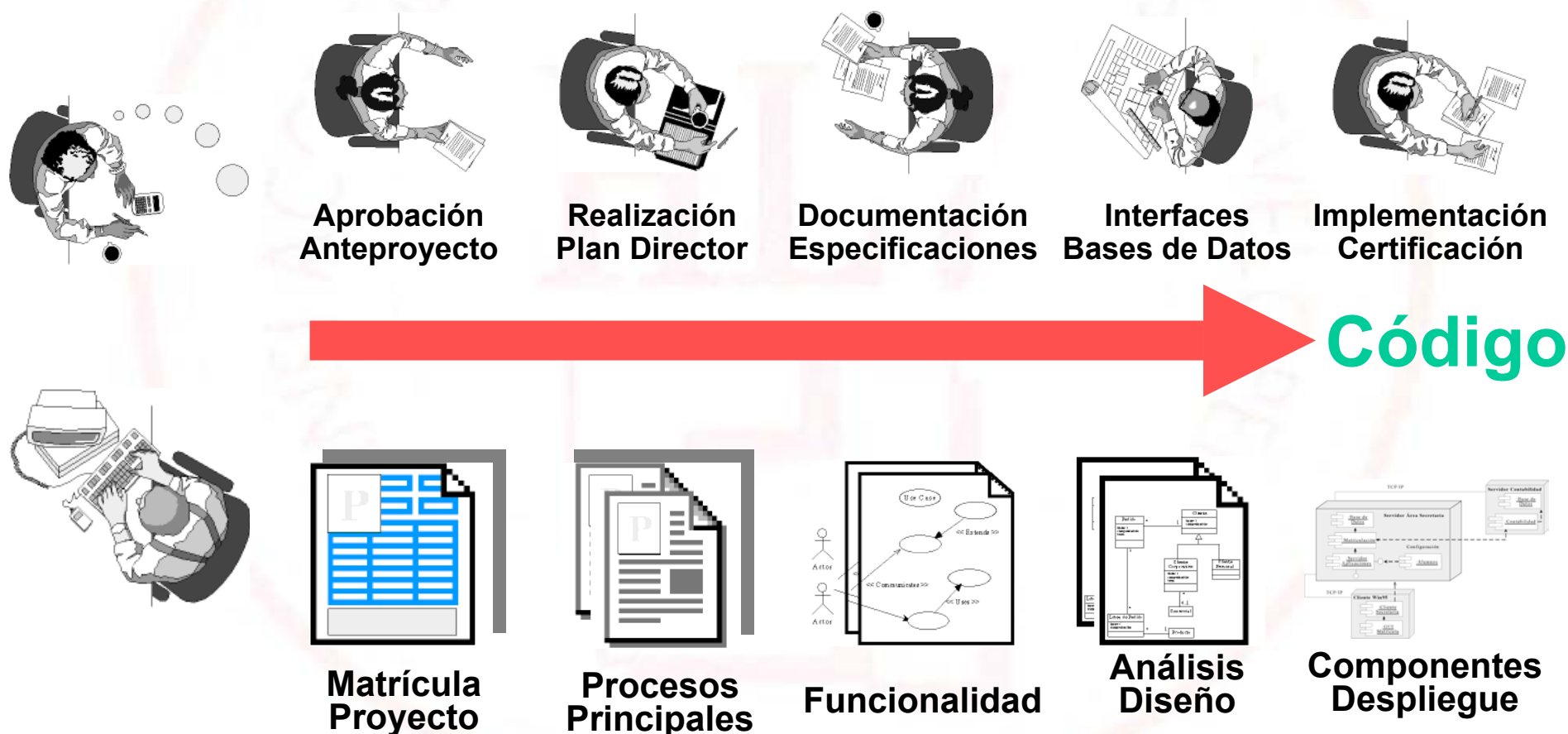


**Mapa de procesos
Reglas de Negocio**

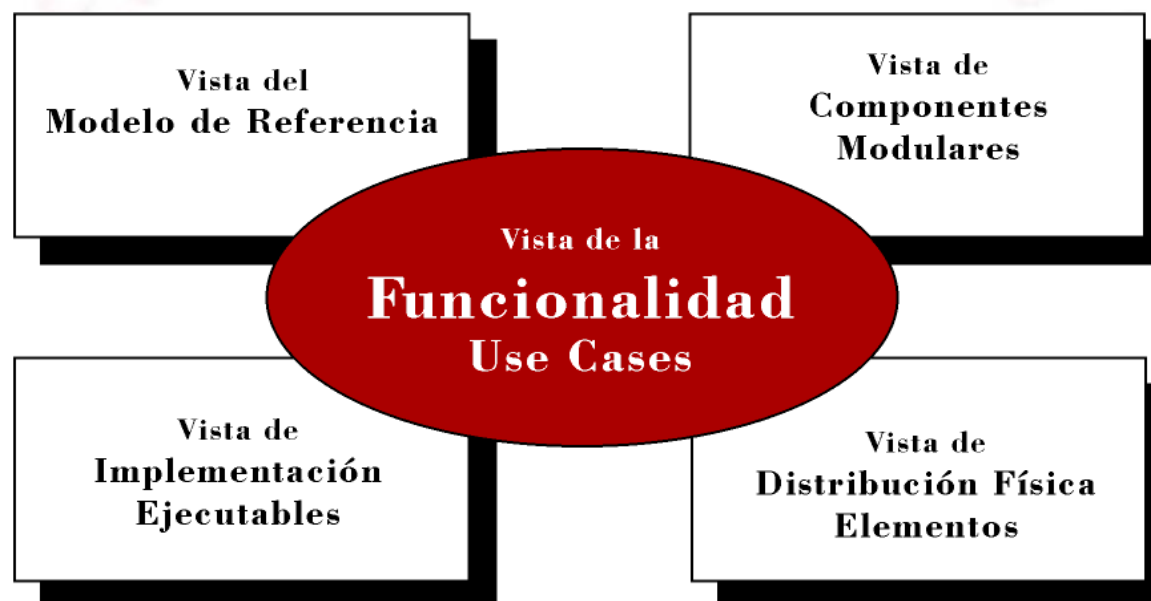
Dirección eficaz de proyectos



Dirección eficaz de proyectos



Vistas de arquitectura



Vista de la Arquitectura 4+1
Notación UML 1.3

Requerimientos y análisis

Modelo de Negocio



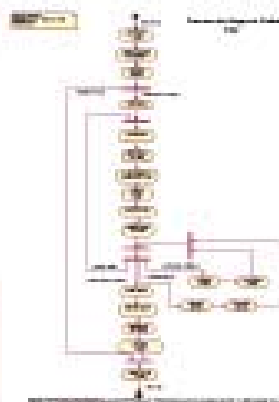
Representa, con un nivel muy alto de abstracción, las **entidades principales del negocio**, su cardinalidad y los roles de sus asociaciones más relevantes en un **diagrama de Clases de Análisis**. Es una visión estática de la estructura del sistema (visto desde dentro), con una granularidad básica; opcionalmente, pueden mostrarse algunos de los atributos más semánticos de cada Clase, pero sin tener en cuenta las operaciones. Tampoco es necesario precisar la fuerza de la asociación (agregación o composición), ni su navegabilidad.

Autor: Analista de Negocio.

Certificador: Gestor de Conocimiento & Interlocutor Cliente.

Requerimientos y análisis

Proceso de Negocio



Representa con un nivel muy alto de abstracción, y a una escala macro, el **comportamiento del sistema** visto desde fuera, a partir de los eventos producidos por los Actores involucrados e implicados. Con una granularidad básica, definimos un **diagrama de Actividad** que muestra los hilos de proceso de las distintas actividades. Su lectura permite identificar un flujo de trabajo principal, y a través de él, los **escenarios de usabilidad** del sistema en discusión más relevantes; siempre desde el punto de vista de los **Actores**. Puede especificarse con un mayor nivel de detalle, a una escala micro, a través de los Casos de Uso y los diagramas de Actividad de sus escenarios.

Autor: Analista de Negocio.

Certificador: Gestor de Conocimiento & Interlocutor Cliente.

Requerimientos y análisis

Modelo de Casos de Uso



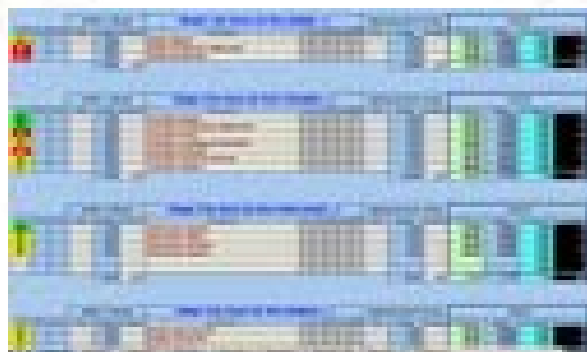
A partir de tres paquetes básicos: 1.- **Administrar aplicación**; 2.- **Realizar procesos de negocio**; 3.- **Consultar puntos de información**; organizamos los Casos de Uso en distintos diagramas que representan la combinación de piezas de funcionalidad que se activan para **resolver un proceso de negocio** (no hay que confundir este modelo con una descomposición funcional o un flujo de trabajo, en el modelo de Casos de Uso todas las piezas están desacopladas y pueden cohesionarse puntualmente en una red de funciones centrada en un proceso). Aplicamos un criterio de granularidad y un criterio de agrupación.

Autor: Analista de Negocio

Certificador: Gestor de Conocimiento & Líder Técnico

Requerimientos y análisis

Censo de Casos de Uso contractuales



Relación de **Casos de Uso** que han sido **certificados** como punto de partida de la funcionalidad de referencia para construir el software. Durante el proceso de Requerimientos y Análisis a lo largo de las distintas fases del proyecto, aparecerán nuevos Casos de Uso secundarios y auxiliares, pero sólo romperá el esquema contractual si aparece un nuevo **Caso de Uso Principal**. Este hecho implicaría que ha sido identificado un nuevo **Actor involucrado** y tendrá un impacto importante en la estimación de horas del proyecto que han sido previamente pactadas.

Autor: Analista de Negocio
Certificador: Coordinador de Proyecto

Requerimientos y análisis

Especificación de los escenarios de un CU



A través de un Caso de Uso describimos las distintas **interacciones** que se producen entre los Actores y el Sistema. Con este artefacto delimitamos un **flujo de eventos principal** y declaramos todas sus **variaciones posibles** y sus **excepciones probables**. Especificamos un Caso de Uso con los siguientes elementos: Nombre, Activador, Propósito, Precondiciones, Postcondiciones y Escenarios. Un Caso de Uso es una pieza de funcionalidad que con el objeto de cumplir su propósito (dar valor a un Actor), activa a otras piezas secundarias (incluidas en un Caso de Uso principal), o auxiliares (extensiones opcionales de otro Caso de Uso).

Autor: Analista de Negocio

Certificador: Gestor de Conocimiento & Líder Técnico

Requerimientos y análisis

Prototipos de interfaz GUI y de comunicación



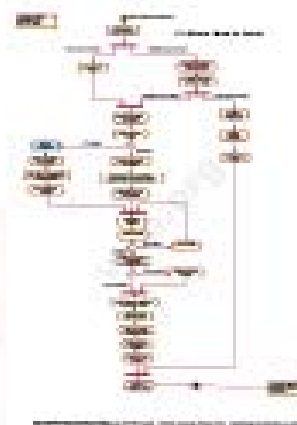
Un Caso de Uso principal es activado por un Actor involucrado y se desarrollan, a través de una interfaz gráfica de usuario (GUI), o una interfaz de comunicación, distintos **escenarios de usabilidad**, los cuales determinarán el comportamiento del sistema. Durante la especificación del flujo de eventos Actor-Sistema, se elabora un prototipo básico de la interfaz necesaria, sin caer en la trampa de abordar en ese momento la implementación de la lógica de presentación.

Autor: Analista de Negocio

Certificador: Gestor de Conocimiento & Líder Técnico

Requerimientos y análisis

Formalización de escenarios



Las fichas donde se mantiene la especificación de los Casos de Uso son reelaboradas **formalmente** en un **diagrama de Actividad** que muestra el hilo de proceso de su flujo de trabajo principal, la diversidad de escenarios y la claves de su sincronización o fusión con el flujo de eventos principal.

Denominamos “**formalización**” del Caso de Uso al proceso de revisión y representación de los **escenarios de usabilidad**: Actor-Sistema con un **diagrama de Actividad**.

Autor: Analista de Negocio

Certificador: Gestor de Conocimiento & Líder Técnico

Diseño

Normalización de escenarios con los objetos



El **diagrama de Secuencia** muestra con todo detalle el intercambio de mensajes que se establece entre los objetos que configuran los escenarios de un Caso de Uso. La visión de este diagrama es **temporal** a través de dos dimensiones: a).- una **horizontal** donde mostramos todos aquellos objetos que intercambian mensajes; y b).- una dimensión **vertical** donde aparecen los mensajes por orden de aparición siguiendo el ciclo de vida de los objetos.

Autor: Analista de Negocio & Programador
Certificador: Arquitecto & Líder Técnico

Diseño

Normalización de escenarios con los objetos



El **diagrama de Colaboración** muestra con todo detalle el intercambio de mensajes que se establece entre los objetos que colaboran dentro de un escenario de Caso de Uso. La visión de este diagrama es **espacial**. Es la otra cara del diagrama de secuencia que en este caso muestra la dinámica de intercambio de mensajes desde un esquema de colaboración, no desde un orden de secuencia en un marco temporal. Es muy útil para visualizar la acumulación de mensajes que recibe cada objeto.

Autor: Analista de Negocio & Programador

Certificador: Arquitecto & Líder Técnico

Diseño

Arquitectura de diseño



Con este diagrama disponemos de las Clases que configuran la **arquitectura de diseño** de la aplicación: Interfaces de Actor, Entidades de negocio persistentes, Actuaciones donde participan n Entidades, Interfaces de sistema que facilitan usar un conjunto básico de métodos a toda una colección de componentes. A través del polimorfismo, escribimos el código una sola vez, lo publicamos en los métodos de una interfaz, y su invocación por objetos externos producirá distintos comportamientos en los mismos.

Autor: Analista de Negocio & Programador

Certificador: Arquitecto & Líder Técnico

Diseño

Arquitectura de diseño

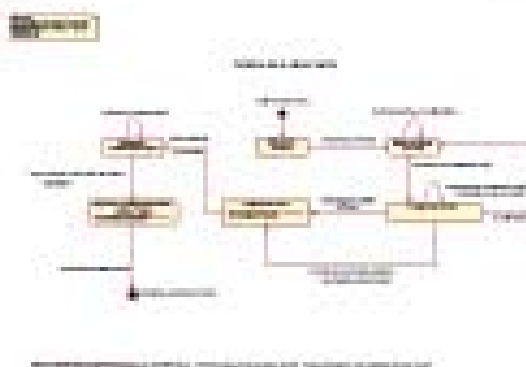


Con los **diagramas de Objetos** podemos evaluar la correspondencia entre la abstracción de las Clases de Diseño y los objetos que se instancian dentro del marco de un escenario de usabilidad cuando se ejecuta la aplicación.

Autor: Analista de Negocio & Programador
Certificador: Arquitecto & Líder Técnico

Diseño

Máquina de estados



El **diagrama de Estados-Transición** muestra el ciclo de vida de los objetos que instancia una Clase. Todos los objetos disponen de un ciclo de estados pero no todos son candidatos a la elaboración de este tipo de diagrama. Normalmente escogemos sólo aquel Objeto que representa el eje central del sistema en discusión. Tan importante es averiguar todos los estados posibles (empezando por identificar el primero y el último), como todos aquellos eventos, acciones y condiciones que permiten la transición de un estado al otro.

Autor: Analista de Negocio & Programador
Certificador: Arquitecto & Líder Técnico

Diseño

Modelos de datos

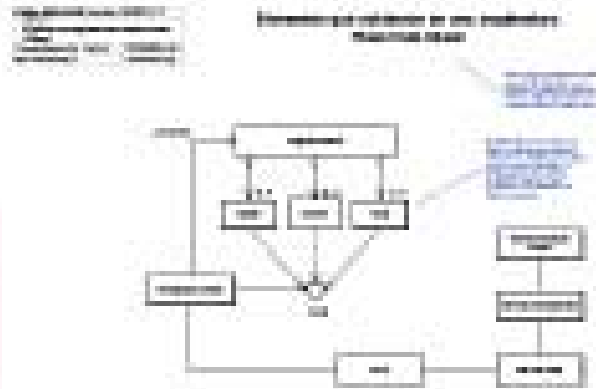


Con el propósito de manejar todos los objetos que requieren persistencia, a partir del diagrama de Clases de diseño, generamos un **esquema lógico de datos** que mapeará en tablas o contenedores la estructura de las asociaciones entre las Clases. El modelo incluye la descripción de los datos persistentes utilizados en la aplicación o sistema. A partir del modelo lógico de datos se generan todas las sentencias SQL que definen el **modelo físico** ajustado a sistema gestor de Base de Datos relacional.

Autor: Analista de Negocio & Programador
Certificador: Arquitecto & Líder Técnico

Diseño

Modelos de arquitecturas



Una arquitectura configura un todo autosuficiente de componentes interdependientes. Hemos de distinguir tres entornos de arquitectura: **conceptual, desarrollo y ejecución**. El primero define la granularidad de los elementos que configuran el sistema en discusión; el segundo, nos permitirá construir dicho sistema dentro de un entorno de desarrollo centrado en componentes; el tercero, nos permitirá instalar y desplegar la aplicación desarrollada durante el proceso de implantación.

Autor: Líder Técnico & Programador
Certificador: Arquitecto

Diseño

Organización de arquitecturas

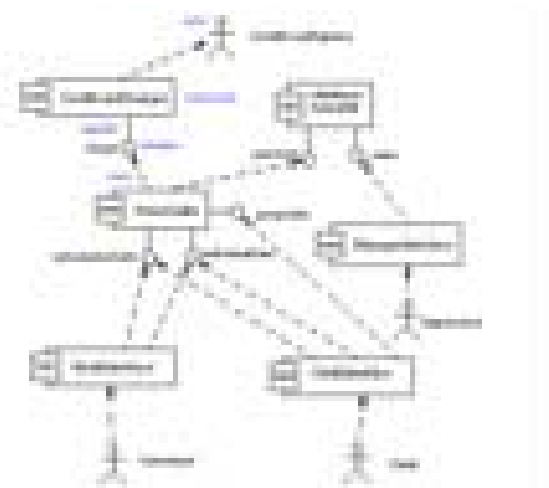


El **diagrama de Paquetes** es un artefacto de propósito general que nos permite agrupar otros artefactos UML, en concreto: Casos de Uso, Clases y Componentes. Podemos estructurar una jerarquía de paquetes anidados e indicar su dependencia con otros paquetes. El paquete principal contendrá la arquitectura de todos los elementos del modelo.

Autor: Líder Técnico & Programador
Certificador: Arquitecto

Implementación

Componentes

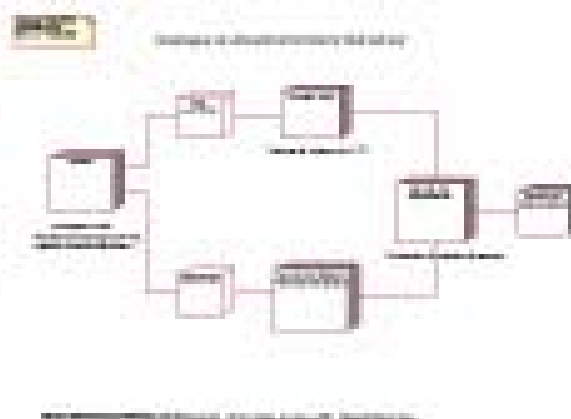


El **diagrama de Componentes** representa una vista física del sistema y su propósito es hacer visible las dependencias entre los distintos componentes de software (Librerías, APIs, GUI, conectores a bases de datos, etc.) . Un Componente es una unidad de composición que ofrece su funcionalidad a través de la especificación contractual de sus interfaces. El diagrama muestra una vista estática de todos los componentes y subsistemas a implementar con una clara indicación de sus interfaces y dependencias.

Autor: Líder Técnico & Programador
Certificador: Arquitecto

Implementación

Despliegue

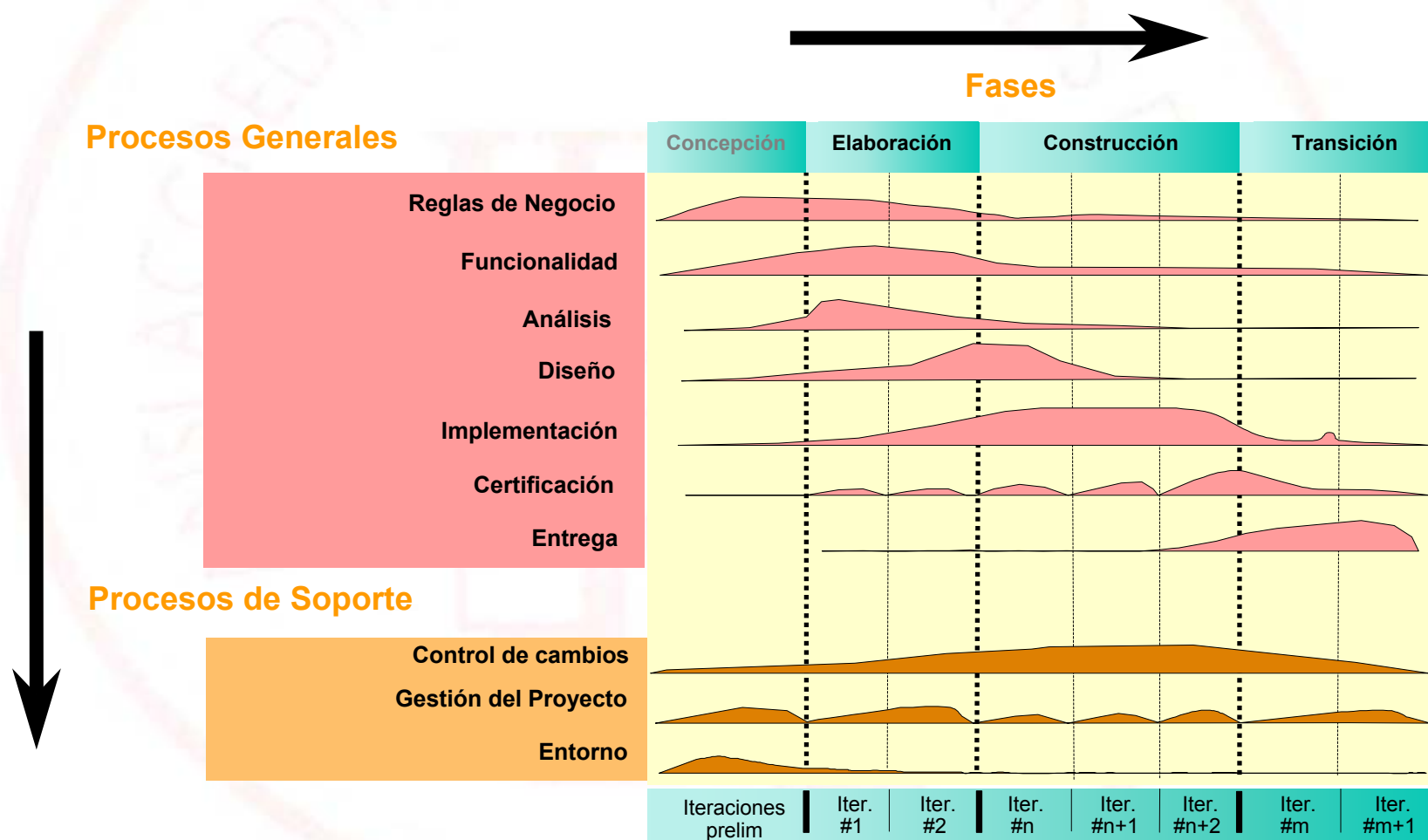


El **diagrama de Despliegue** Muestra de que manera vamos a desplegar físicamente el sistema que hemos construido dentro de un entorno de ejecución. Su propósito es mostrar donde se ubicarán los distintos nodos del sistema (máquinas físicas o virtuales), y cómo se comunicarán entre ellos. Es una documentación muy útil para el equipo de implantación ya que la correcta ubicación de los distintos nodos en la red condicionará críticamente la mayoría de nuestros requerimientos no funcionales: seguridad, rendimiento, escalabilidad, etc.

Autor: Líder Técnico & Programador
Certificador: Arquitecto

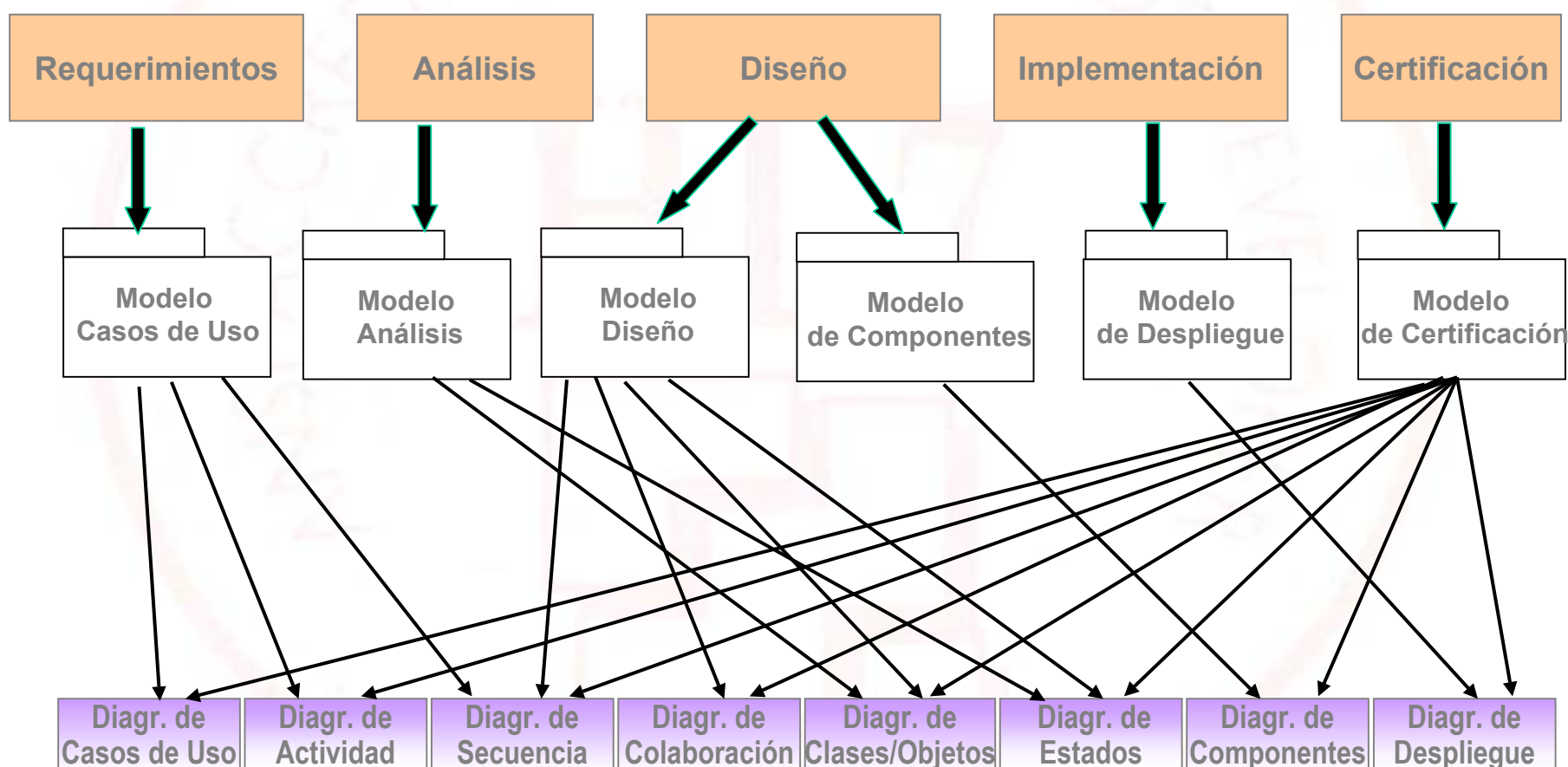
Plan Director de Iteraciones

Fases, Procesos y Entregables

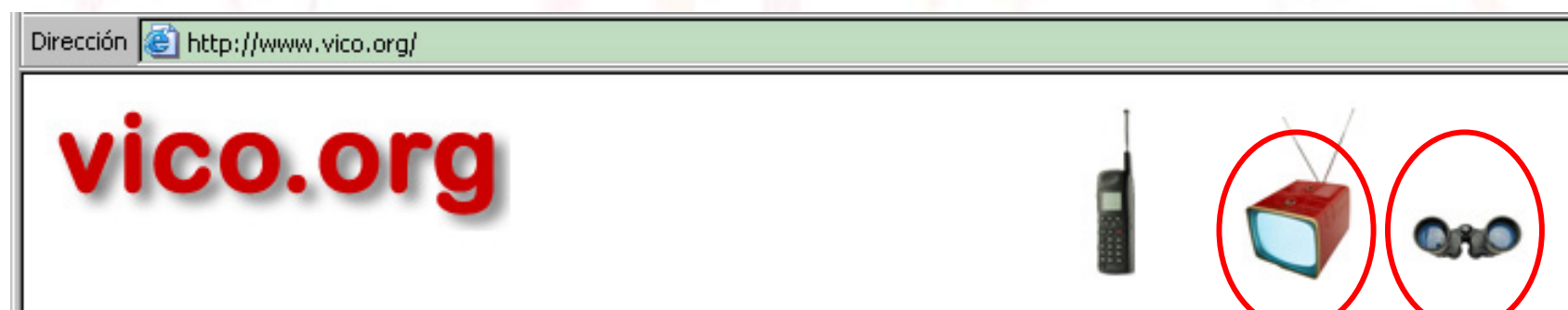


Metodología de desarrollo

Certificación de Entregables



Donde localizar recursos UML



Donde localizar recursos UML



Con la colaboración de:

ORACLE®

IBM®



jvilalta@vico.org