

Desarrollo de Software Orientado a Objeto usando UML

Contenido

- introducción modelado de sw
- claves en desarrollo de si
- abstracción - modelado visual
- notación (visual) - Beneficios
- introducción: UML
- que es UML?.
- Historia del UML.
- Participantes en UML 1.0
- UML "aglutina" enfoques OO.
- Perspectivas de UML.

Contenido

- Proceso de desarrollo de SW basado en UML.
- Que es un proceso de desarrollo de sw?.
- Rational Unified Process (RUP).
- Fases e Hitos (Milestones).
- Elementos en RUP.
- Características Esenciales de RUP.
- Proceso dirigido por los Casos de Uso.
- Proceso Iterativo e Incremental.
- Proceso Centrado en la Arquitectura.
- Fases del ciclo de vida.
- Esfuerzo respecto de las Workflows.
- Esfuerzo respecto de las Fases.

Contenido

- Diagramas de UML.
- Modelos y diagramas.
- Diagramas de UML.
- Organización de los modelos
- Diagrama de actividad
- Diagrama de Casos de Uso.
- Fully Use Case.
- Diagrama de clases.

Contenido

- Diagramas de UML.
- clases
- clases : notación grafica
- clases : encapsulación.
- Relaciones entre clases
- Asociación.
- Agregación.
- Composición.

Contenido

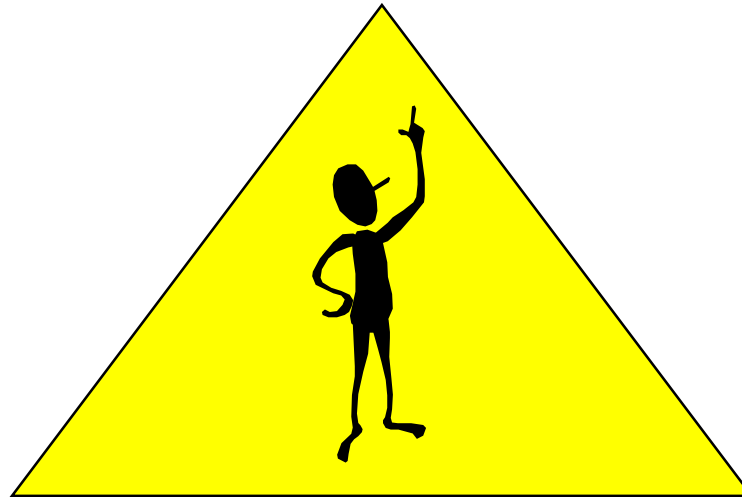
- Diagramas de UML.
- Clases y Objetos.
- Generalización.
- Polimorfismo.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de componentes.
- Diagrama de despliegue.
- Resumen.

Introducción

Modelado de SW

Claves en Desarrollo de SI

Notación

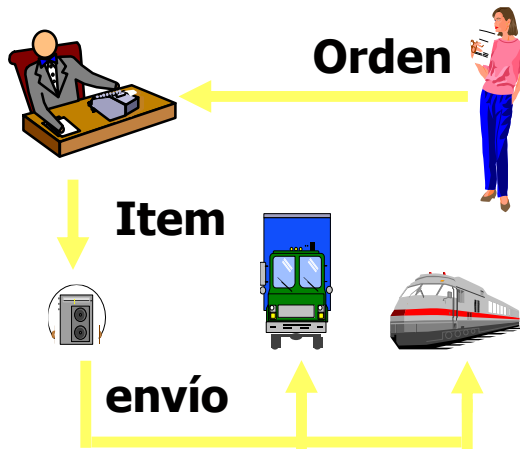


Herramientas

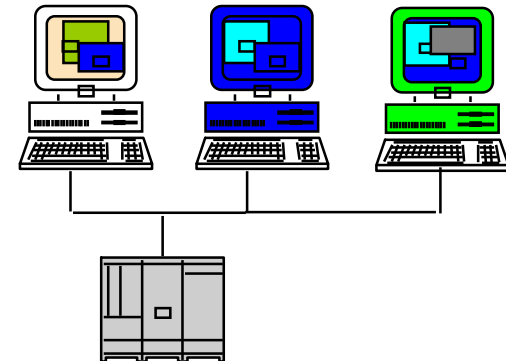
Proceso

Abstracción - Modelado Visual (MV)

"El modelado captura las partes esenciales del sistema"



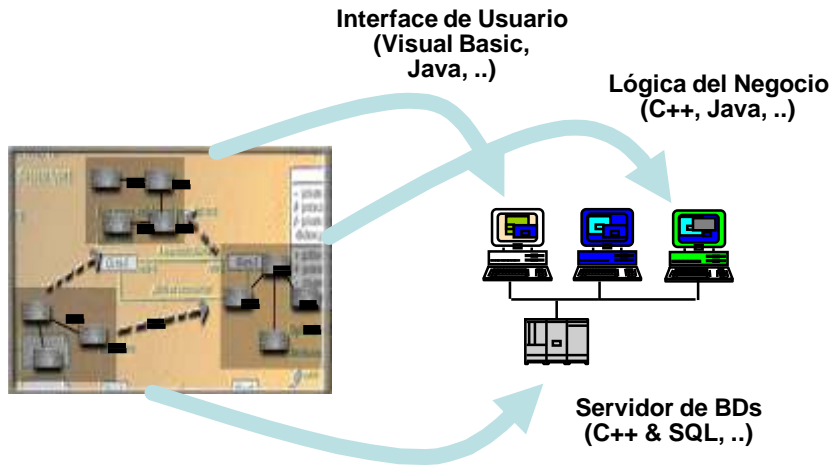
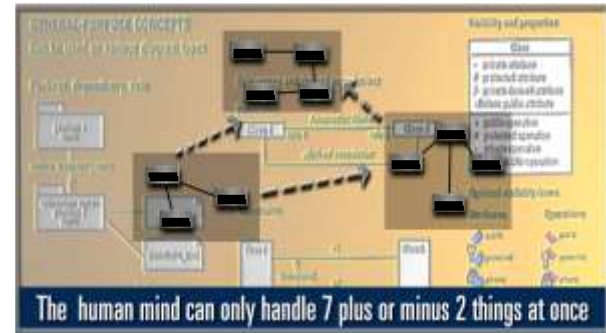
Proceso de Negocios



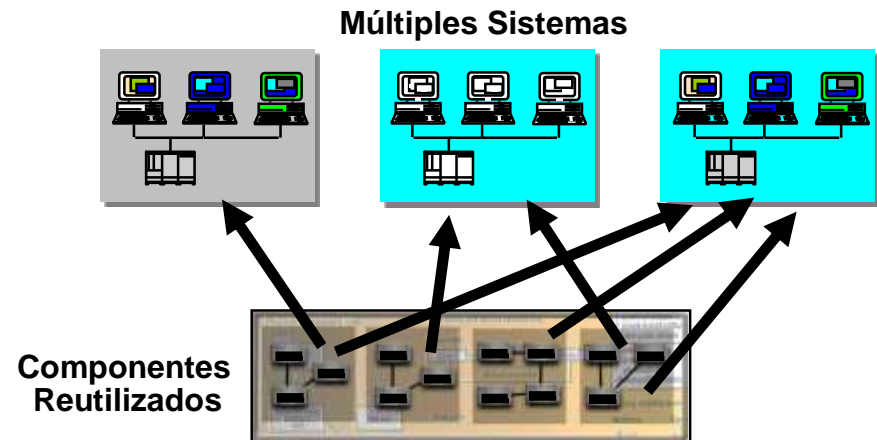
Sistema Computacional

II. Notación (Visual) - Beneficios

Manejar la complejidad



“Modelar el sistema independientemente del lenguaje de implementación”



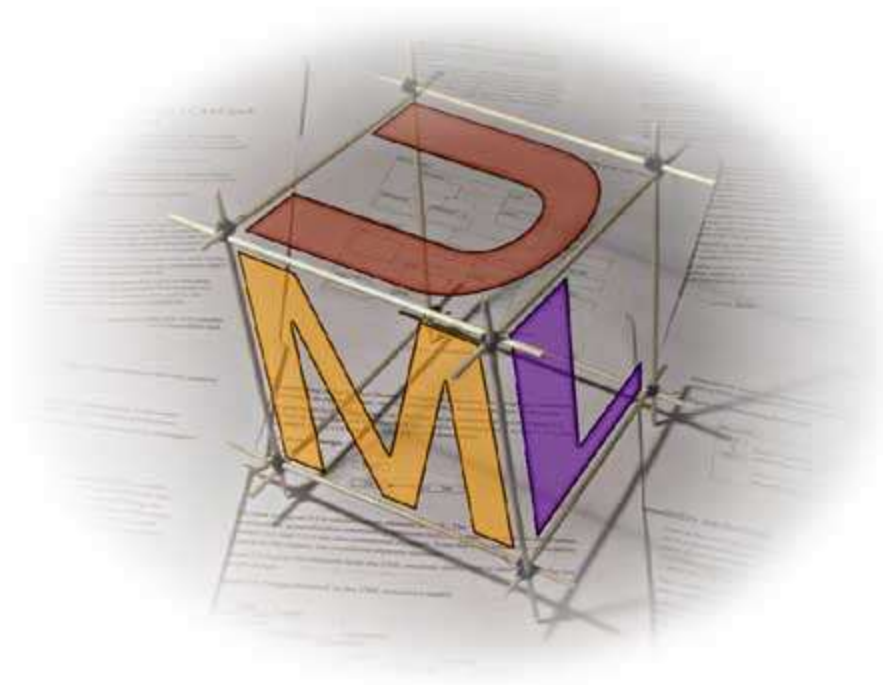
Promover la Reutilización

Introducción: UML

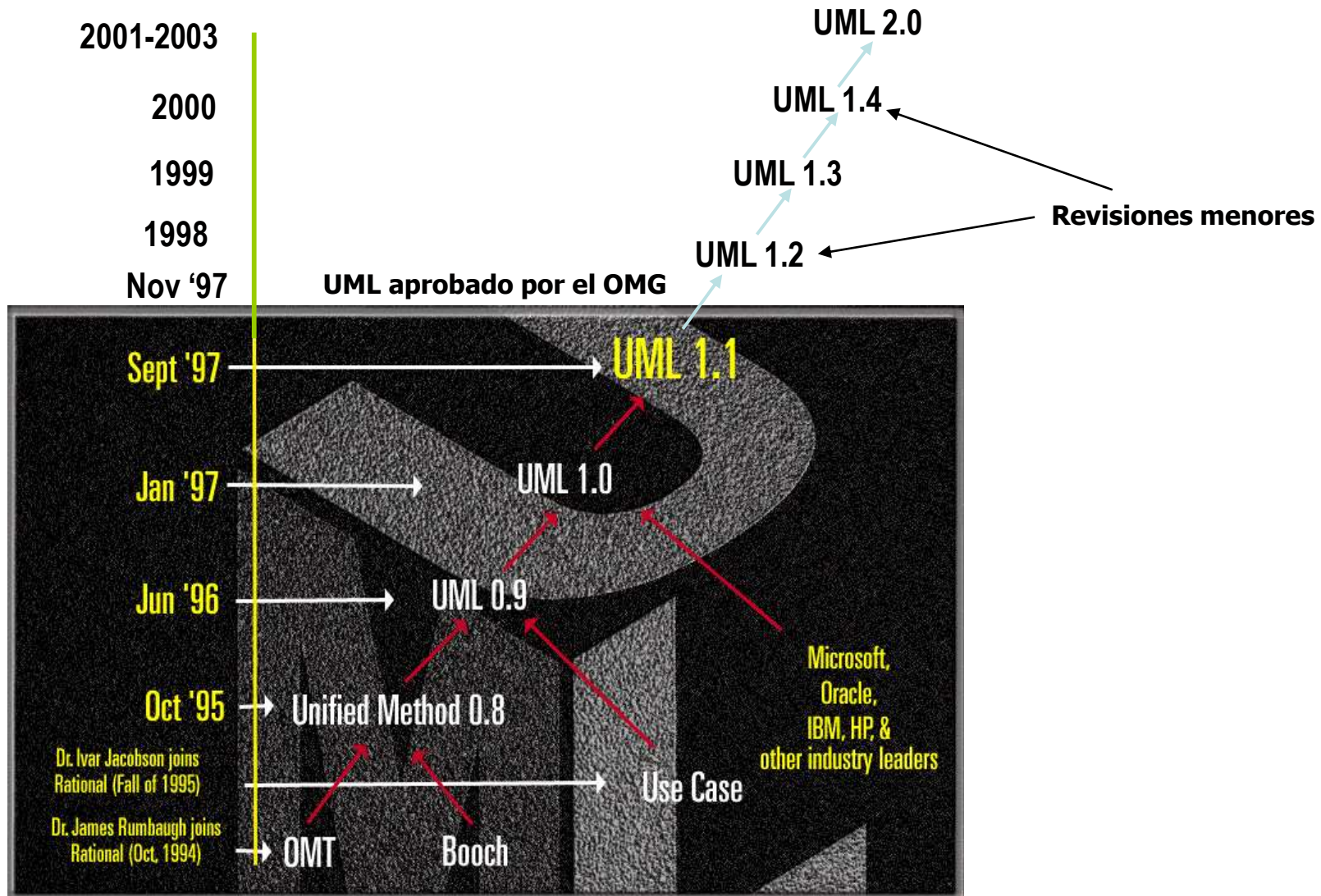
¿Qué es UML?

- UML = Unified Modeling Language
- Un lenguaje de propósito general para el modelado orientado a objetos
- Documento “OMG Unified Modeling Language Specification”
- UML combina notaciones provenientes desde:
 - Modelado Orientado a Objetos
 - Modelado de Datos
 - Modelado de Componentes
 - Modelado de Flujos de Trabajo (Workflows)

Historia de UML



Historia de UML



Proceso de Desarrollo de SW basado en UML

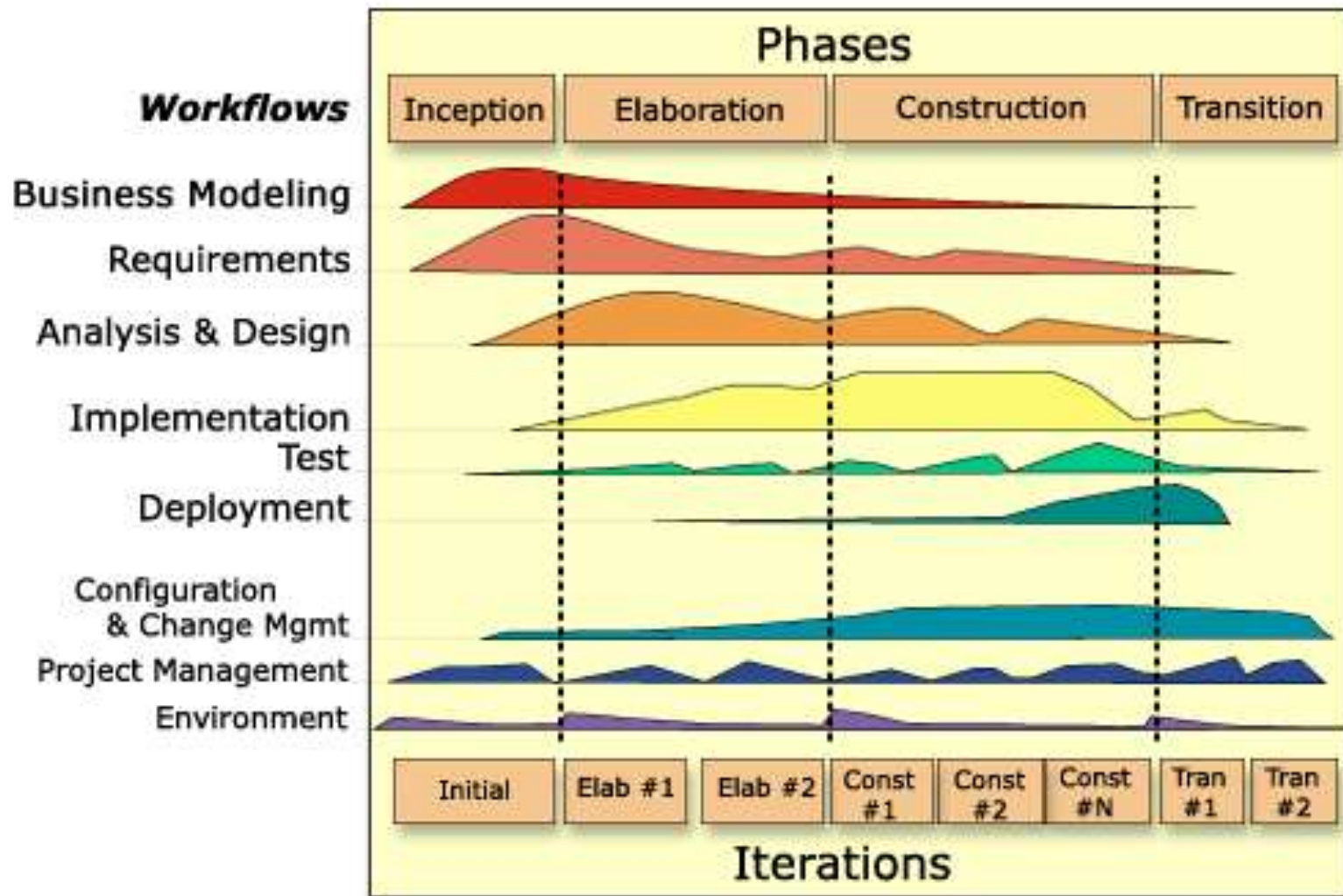
¿Qué es un Proceso de Desarrollo de SW?

- Define **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo

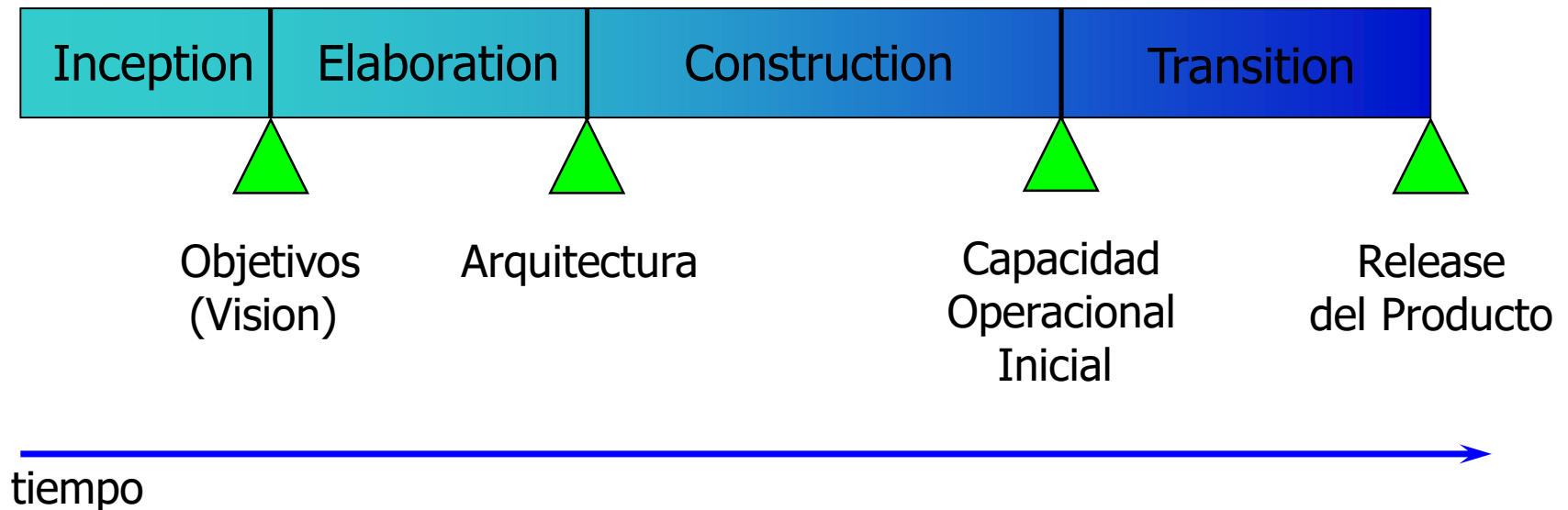


No existe un proceso de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable

Dos Dimensiones



Fases e Hitos (Milestones)



Elementos en RUP

- Workflows (Disciplinas)

Workflows Primarios

- Business Modeling (Modelado del Negocio)
- Requirements (Requisitos)
- Analysis & Design (Análisis y Diseño)
- Implementation (Implementación)
- Test (Pruebas)
- Deployment (Despliegue)

Workflows de Apoyo

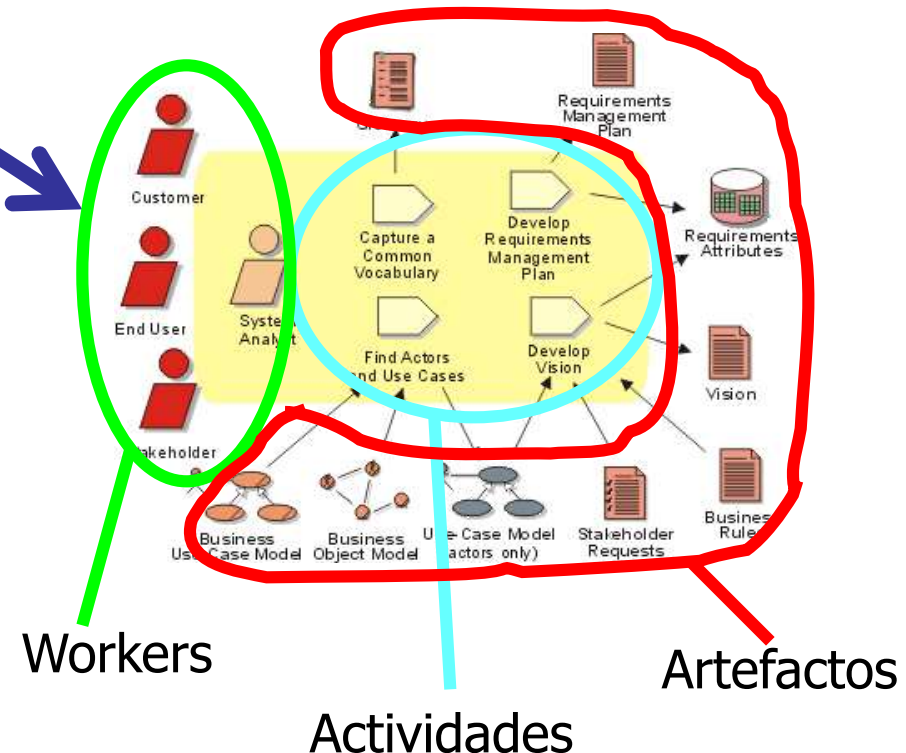
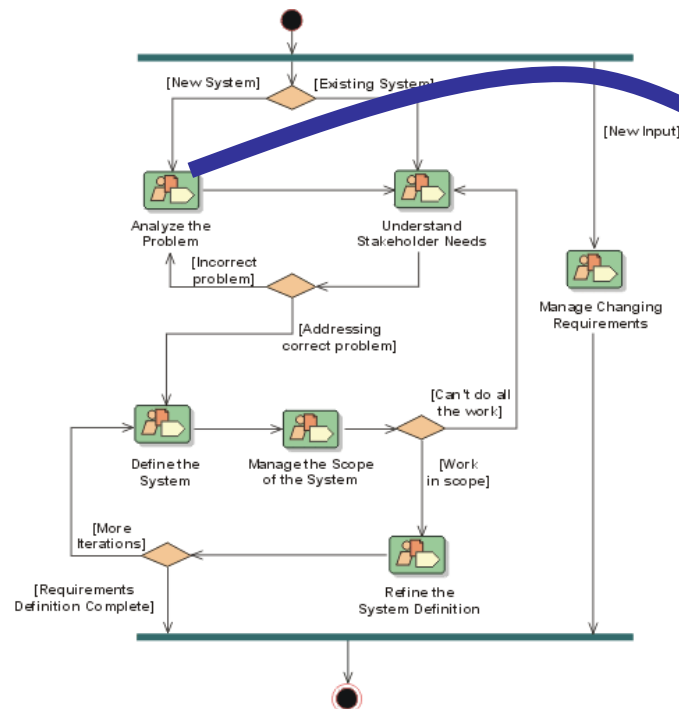
- Environment (Entorno)
- Project Management (Gestión del Proyecto)
- Configuration & Change Management (Gestión de Configuración y Cambios)

... Elementos en RUP

Workflow, Workflow Detail , Workers, Actividades y Artefactos

Ejemplo Workflow: Requirements

Workflow Detail:Analyse the Problem



... Elementos en RUP

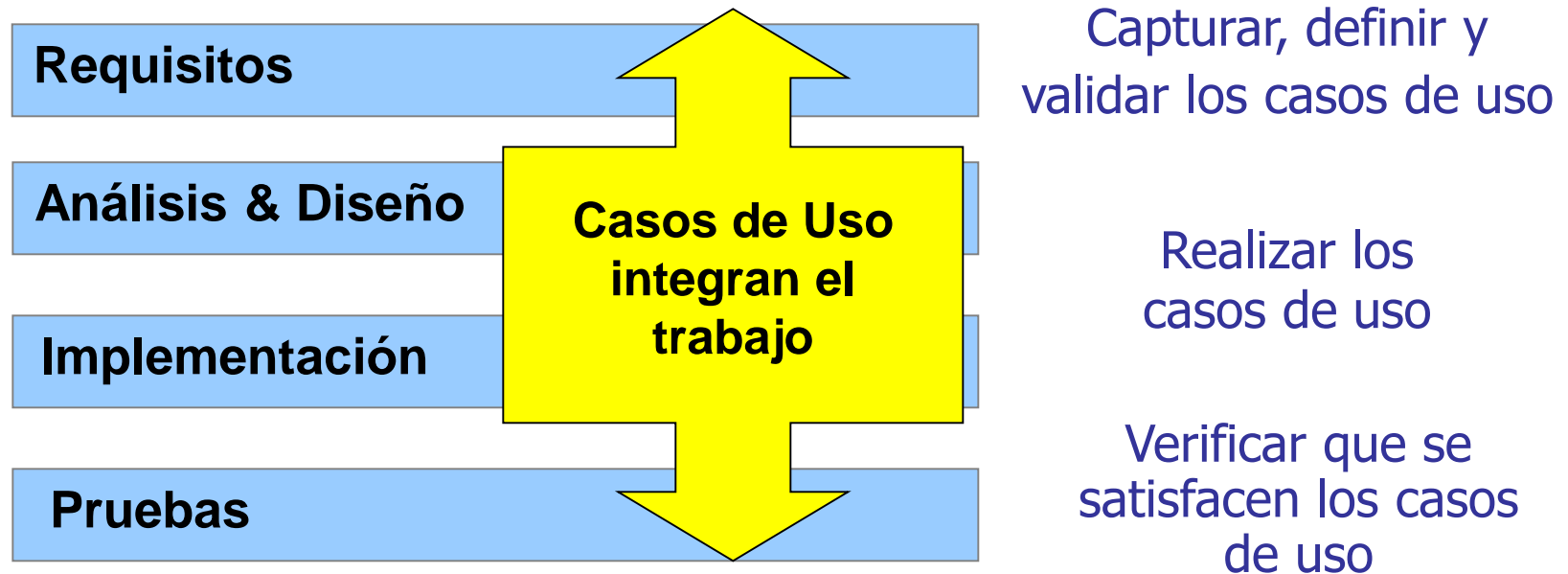
Artefactos

- Resultado parcial o final que es producido y usado durante el proyecto. Son las entradas y salidas de las actividades
- Un artefacto puede ser un documento, un modelo o un elemento de modelo
- Conjuntos de Artefactos
 - Business Modeling Set
 - Requirements Set
 - Analysis & Design Set
 - Implementation Set
 - Test Set
 - Deployment Set
 - Project Management Set
 - Configuration & Change Management Set
 - Environment Set

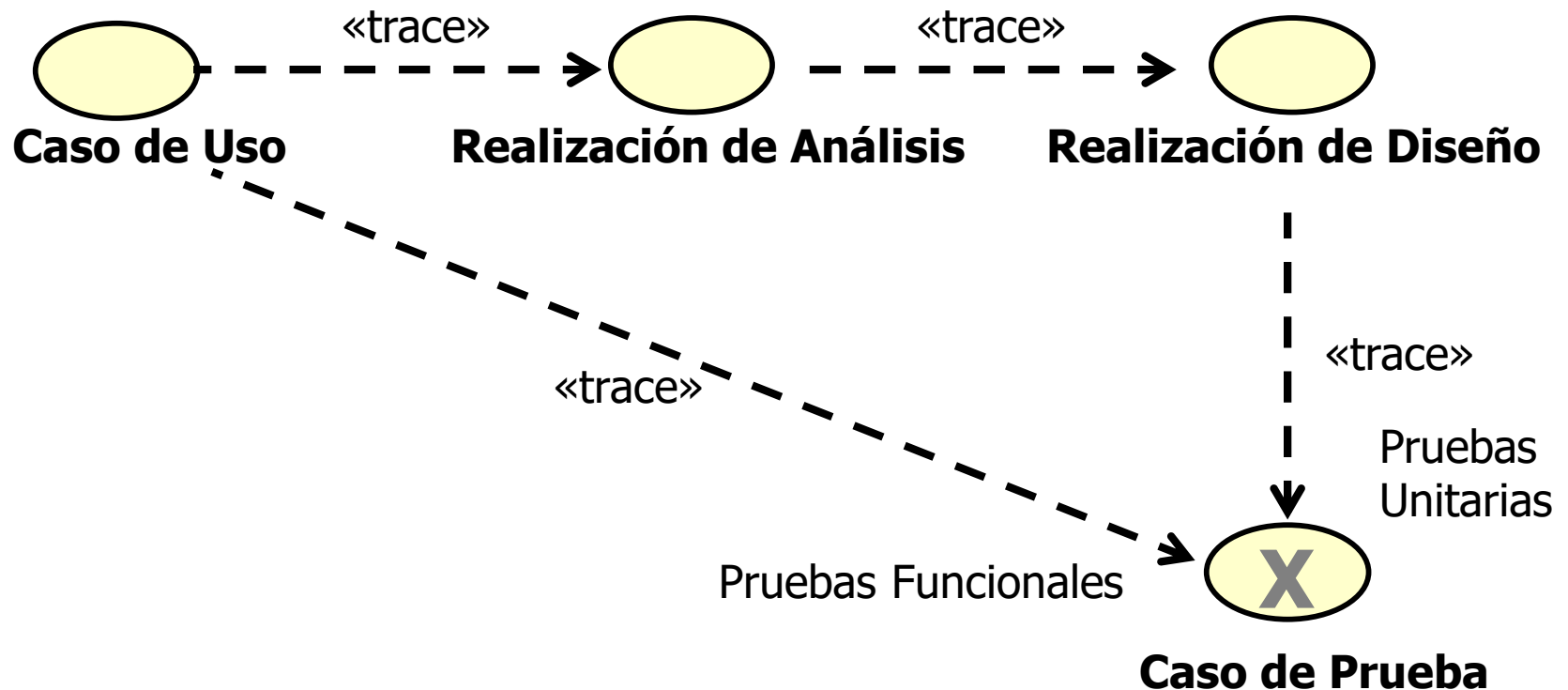
Características Esenciales de RUP

- Proceso Dirigido por los Casos de Uso
- Proceso Iterativo e Incremental
- Proceso Centrado en la Arquitectura

Proceso dirigido por los Casos de Uso



... Proceso dirigido por los Casos de Uso



... Proceso dirigido por los Casos de Uso

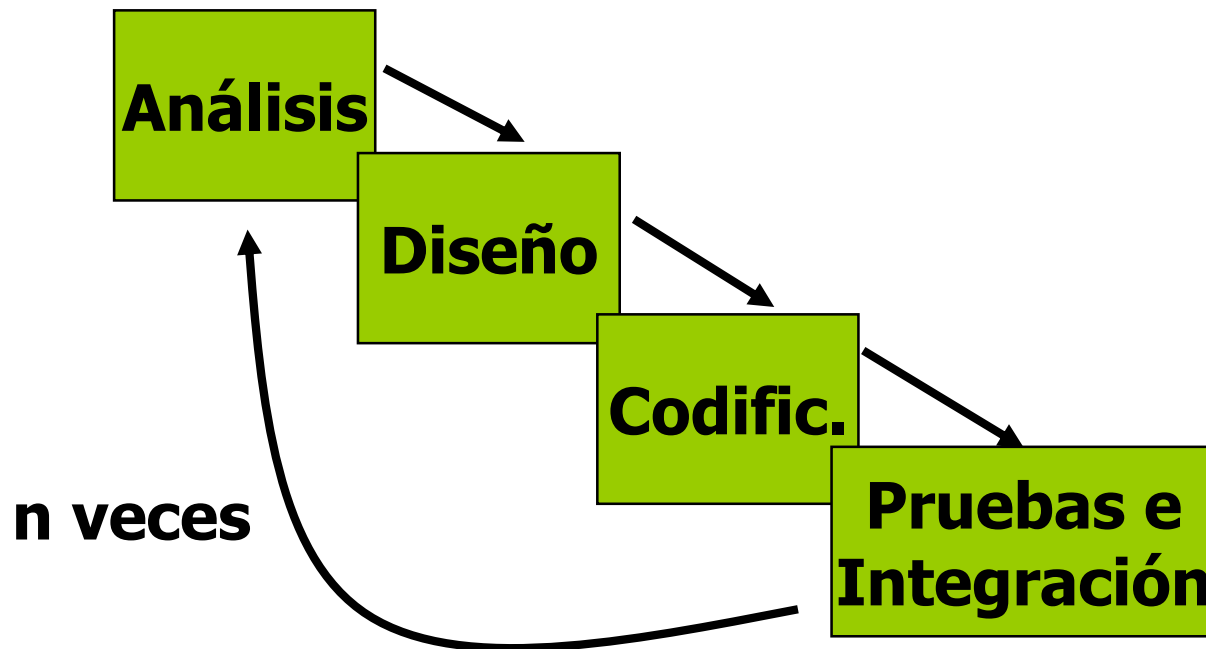
Estado de aspectos de los Casos de Uso al finalizar cada fase

	Modelo de Negocio Terminado	Casos de Uso Identificados	Casos de Uso Descritos	Casos de Uso Analizados	Casos de Uso Diseñados, Implementados y Probados
Fase de Concepción	50% - 70%	50%	10%	5%	Muy poco, puede que sólo algo relativo a un prototipo para probar conceptos
Fase de Elaboración	Casi el 100%	80% o más	40% - 80%	20% - 40%	Menos del 10%
Fase de Construcción	100%	100%	100%	100%	100%
Fase de Transición					

The Unified Software Development Process. I. Jacobson, G. Booch y J. Rumbaugh. página 358. Addison-Wesley, 1999.

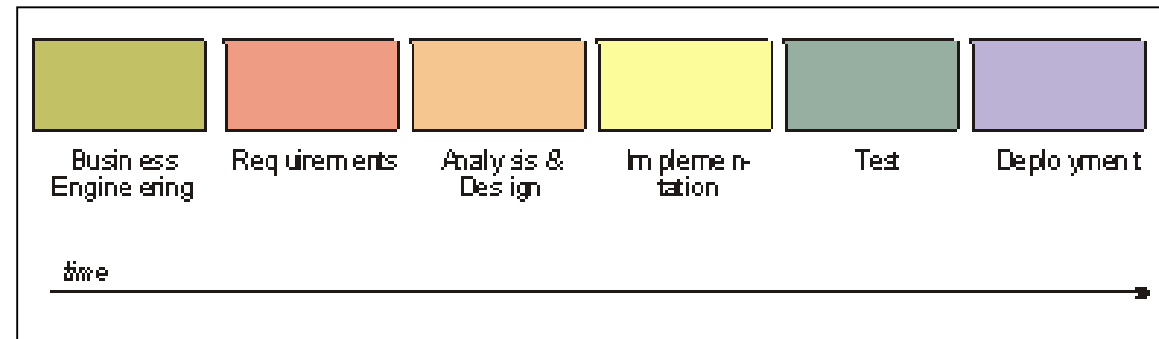
... Proceso Iterativo e Incremental

- Las actividades se encadenan en una mini-cascada con un alcance limitado por los objetivos de la iteración

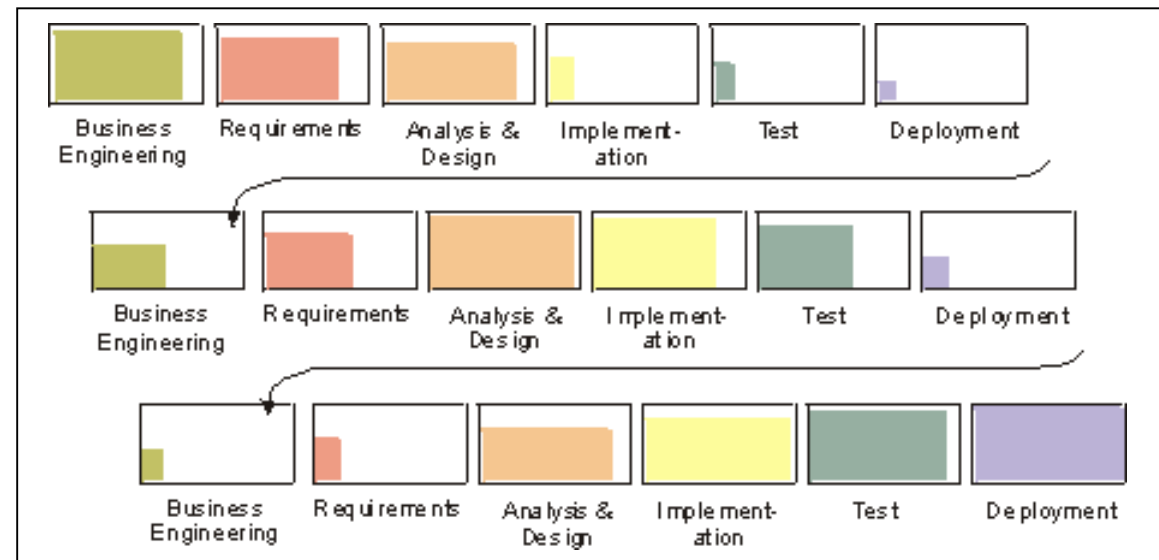


Proceso Iterativo e Incremental

Enfoque Cascada

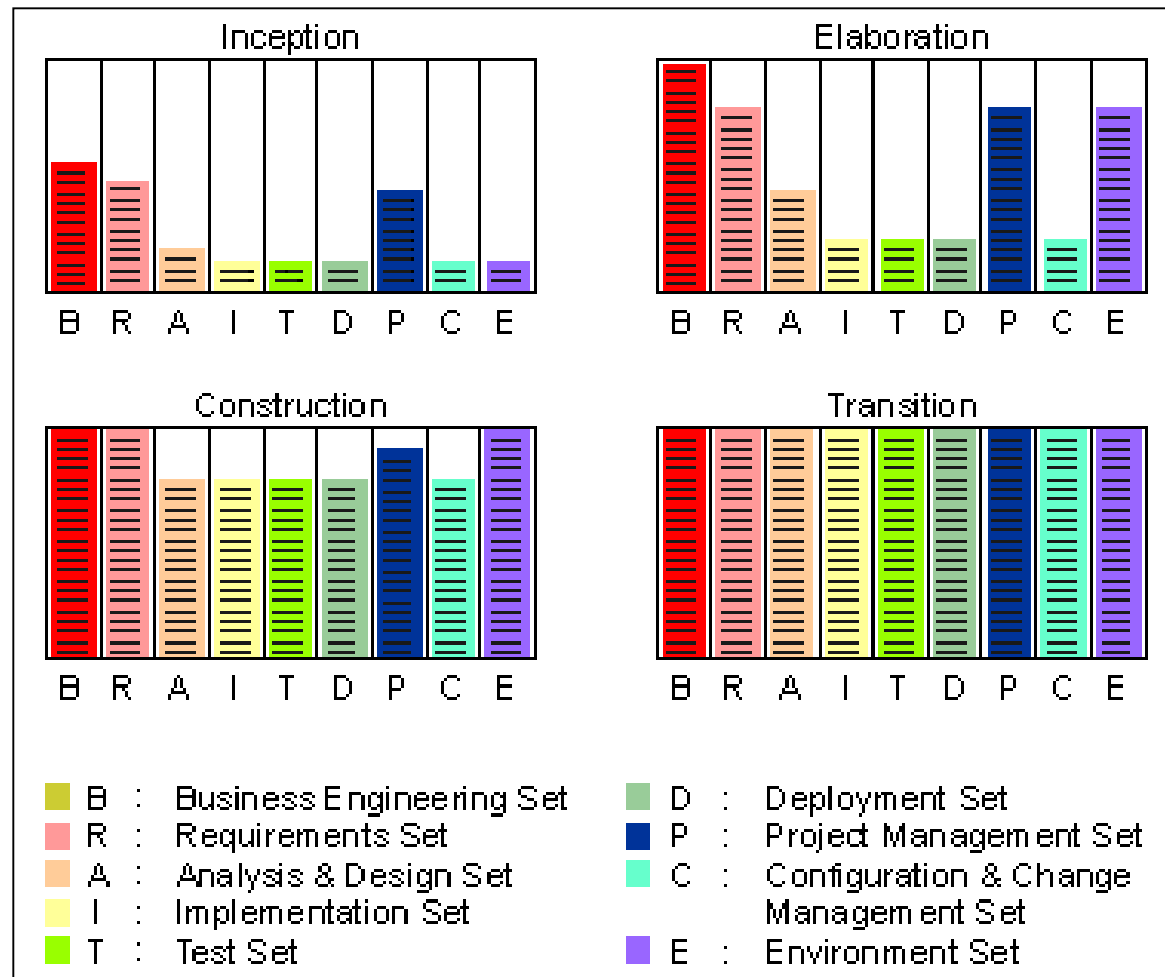


Enfoque Iterativo e Incremental



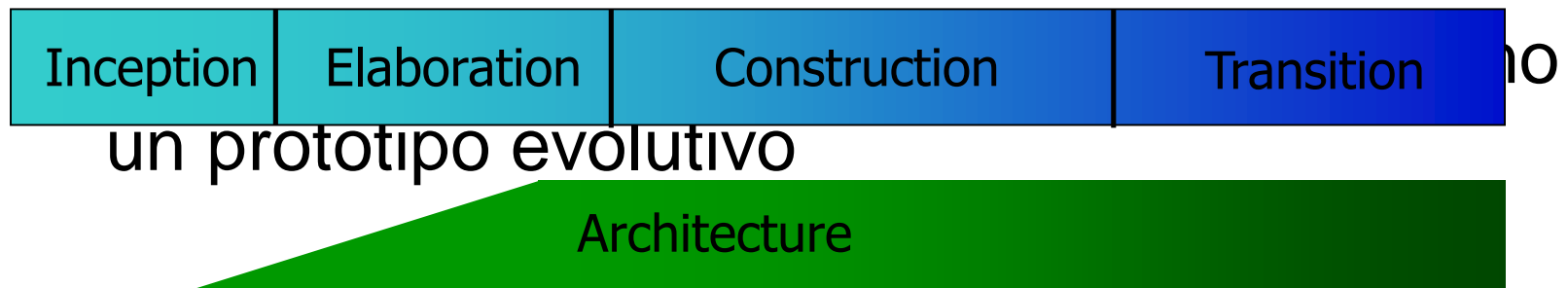
... Proceso Iterativo e Incremental

Grado de Finalización de Artefactos

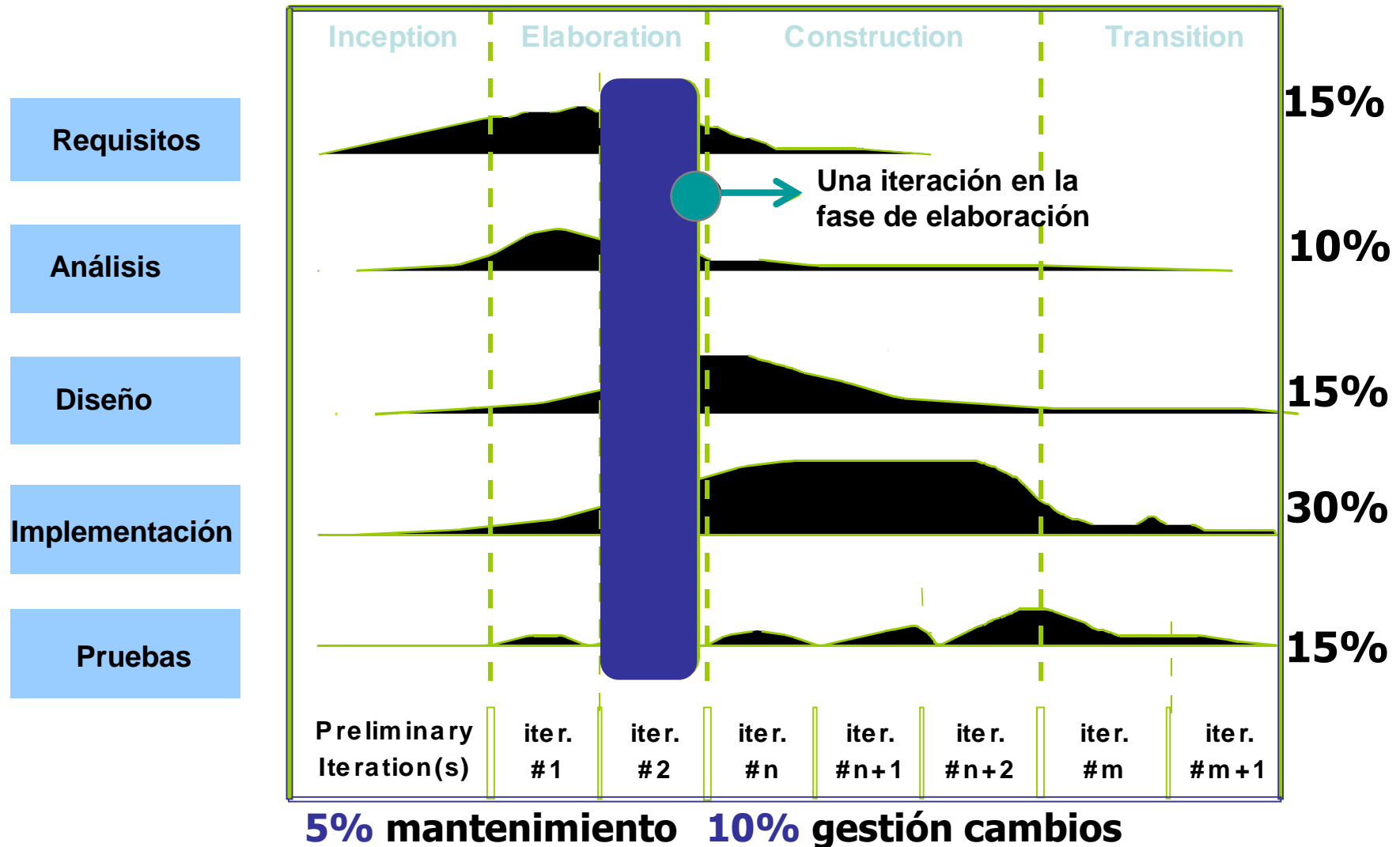


Proceso Centrado en la Arquitectura

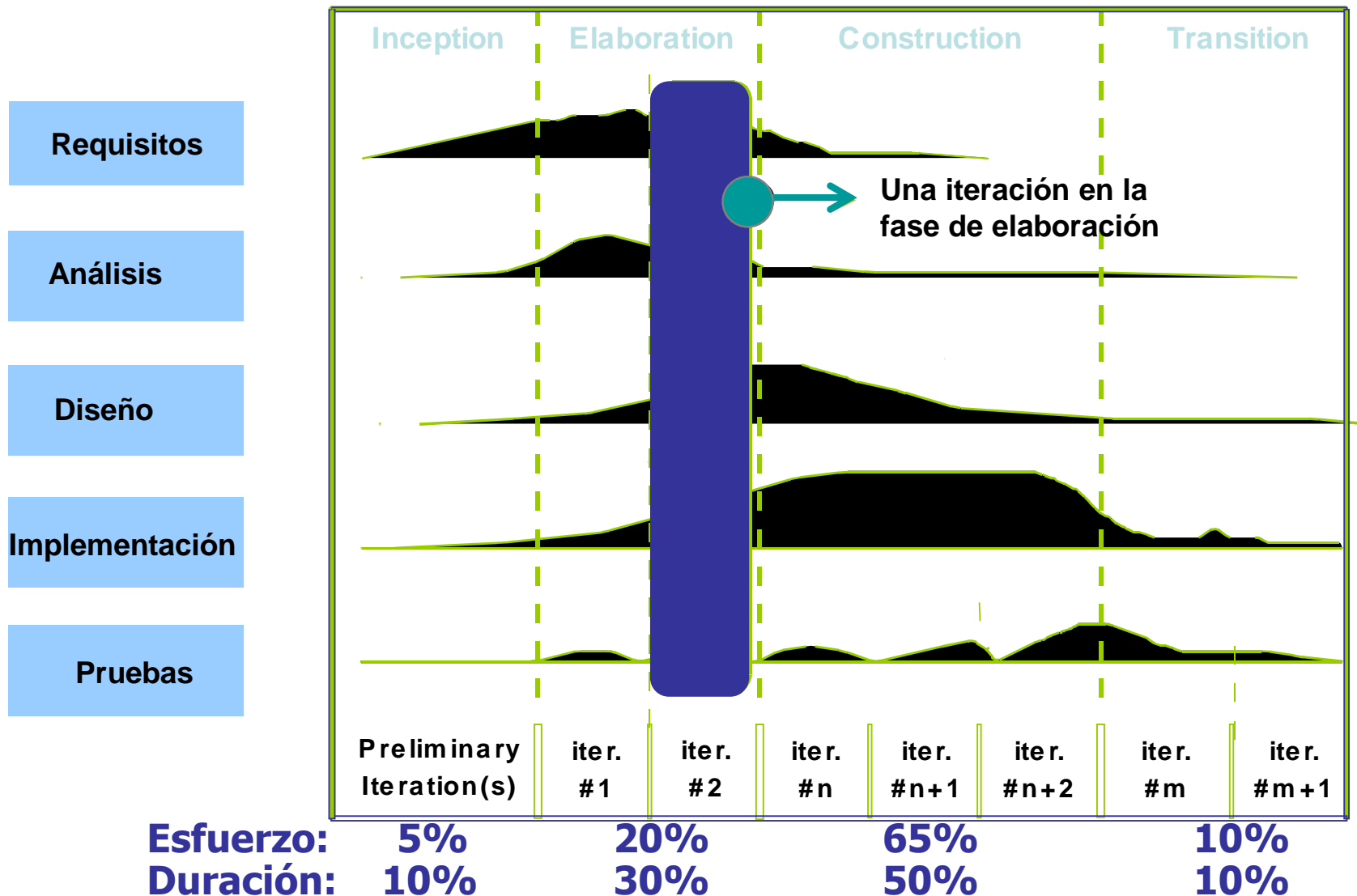
- Arquitectura de un sistema es la organización o estructura de sus partes más relevantes
- Un arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades
- RUP establece refinamientos sucesivos de



Esfuerzo respecto de las Workflows



...Esfuerzo respecto de las Fases



Diagramas de UML

... Diagramas de UML

Los diagramas expresan gráficamente partes de un modelo

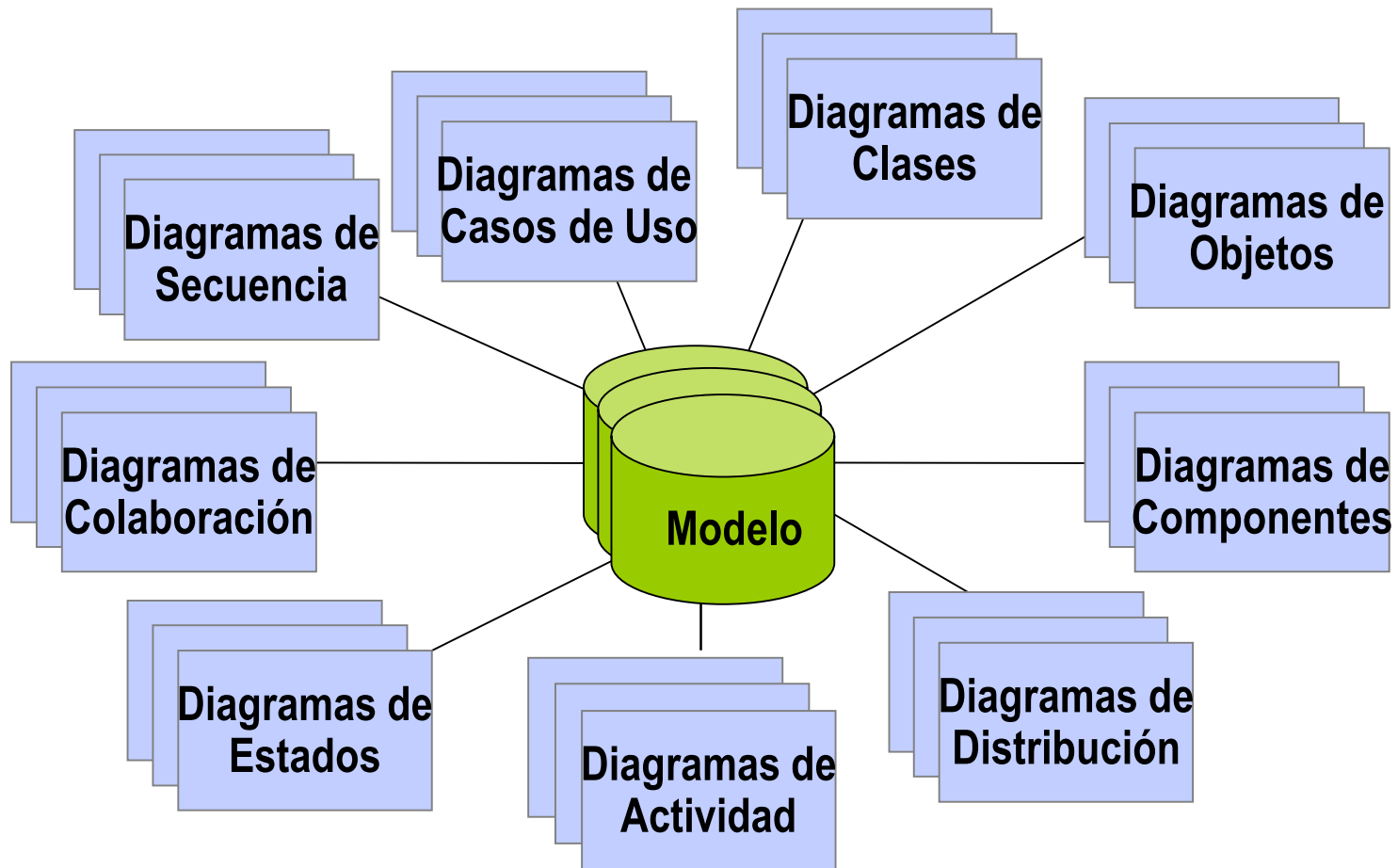
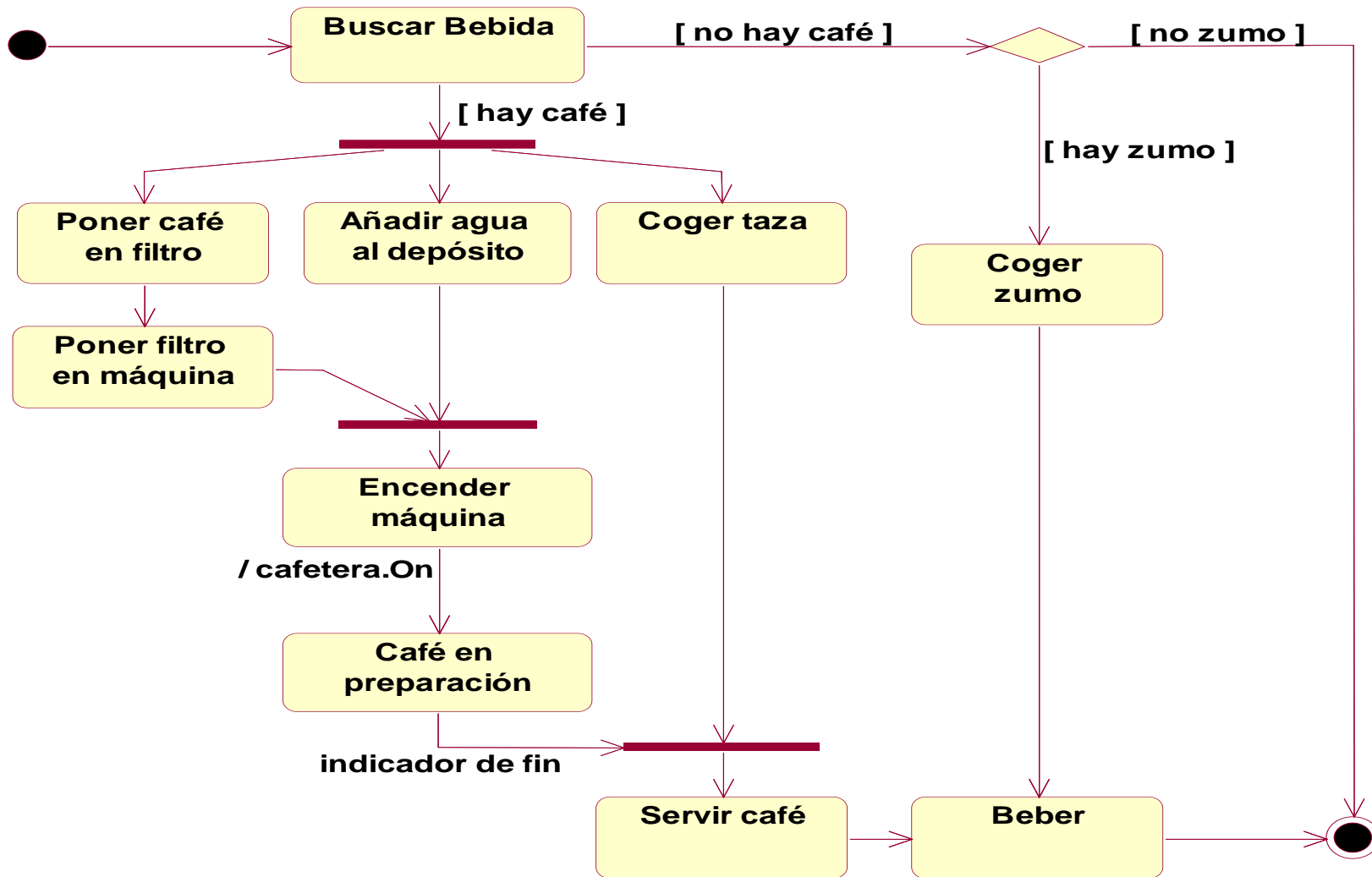


Diagrama de Actividad



... Otro Ejemplo (con *swim lines*)

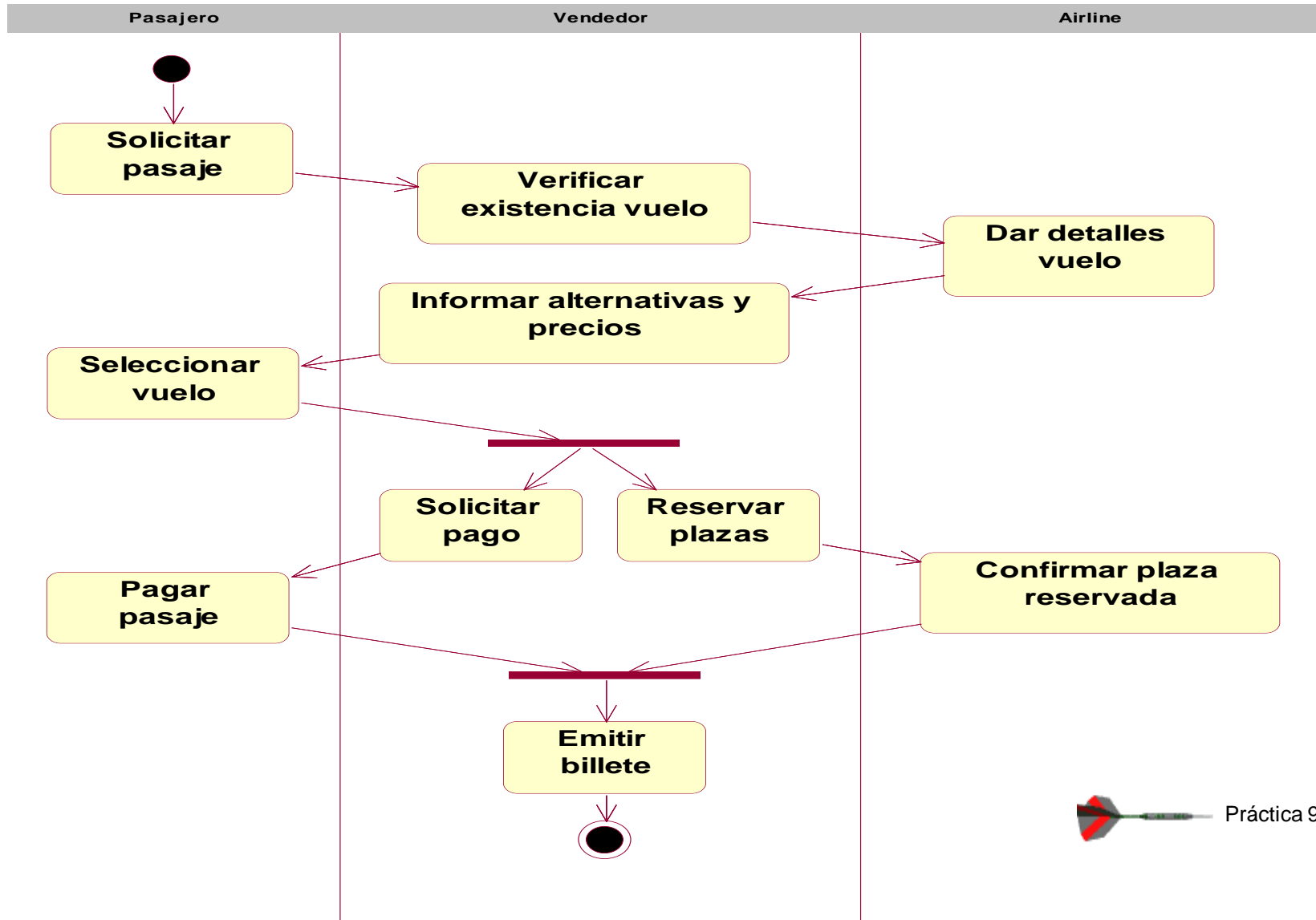
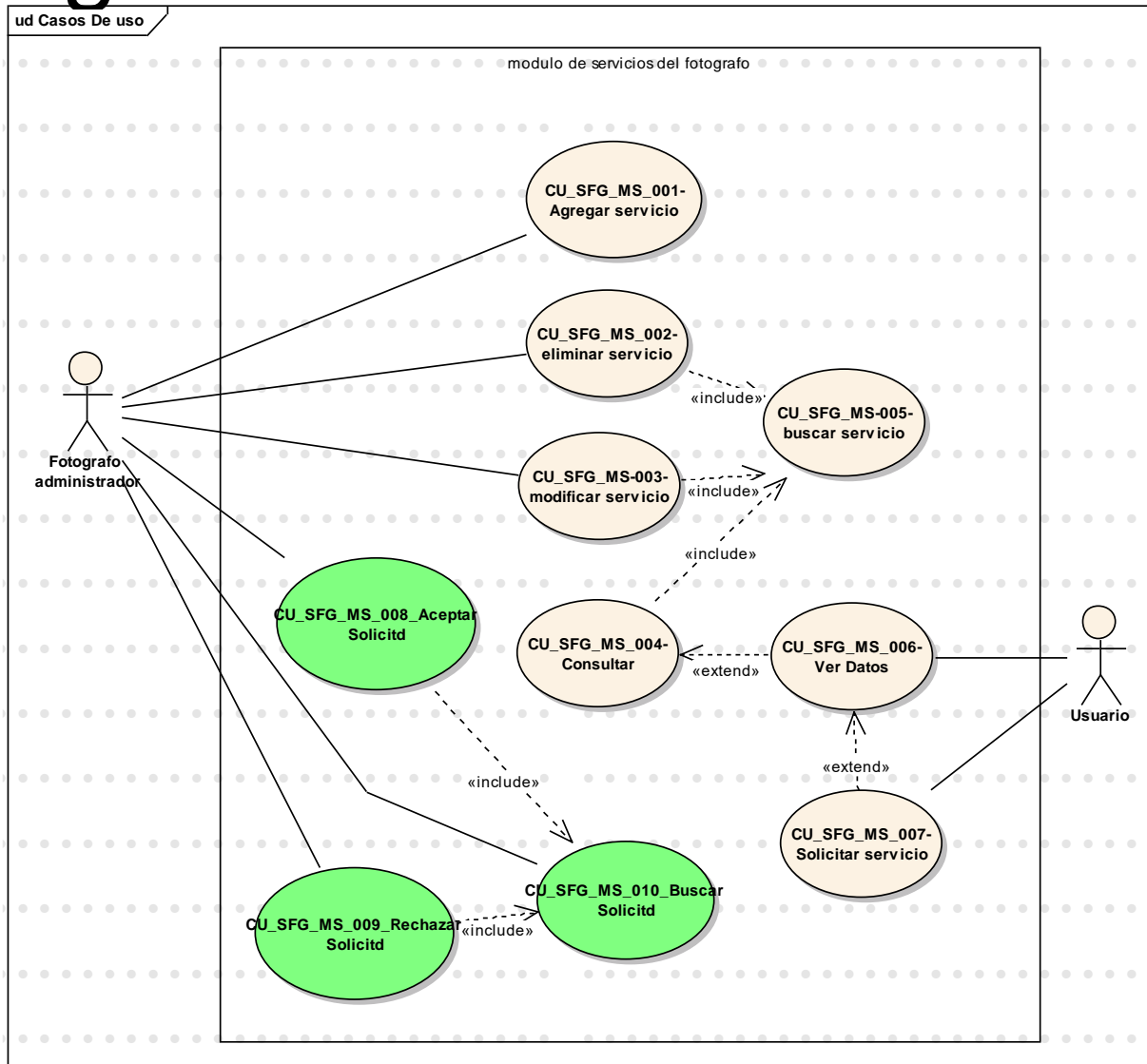


Diagrama de Casos de Uso

- Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje
- No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos

Diagrama de Casos de Uso



Fully use case

- Es un documento escrito que dispone de la explicación detallada del caso de uso
- Esto lo logra organizando la información de una determinada forma
 - Nombre
 - Breve descripción
 - Actores
 - Pre condiciones
 - Post condición
 - Flujo principal
 - Flujos alternativos
 - Condiciones especiales.
- Existe una amplia variedad de modelos de fully use case.

Diagrama de Clases

- Método del sustantivo
 - Determinar todos los sustantivos
 - Eliminar sustantivos de acuerdo a los siguientes criterios:
 - Redundantes
 - Irrelevantes
 - Vagos
 - Operaciones
 - Constructores de implementación
 - Meta lenguajes
 - Atributos
 - Establecer relaciones
 - Asociaciones
 - Cardinalidades
 - generalizaciones

Diagrama de clases

Persona

- nombre: String
- fechaDeNacimiento: Date
- # direccion: String
- + telefono: String

- + calcularEdad(Date) : void
- + setNombre(String) : void
- + getNombre() : String

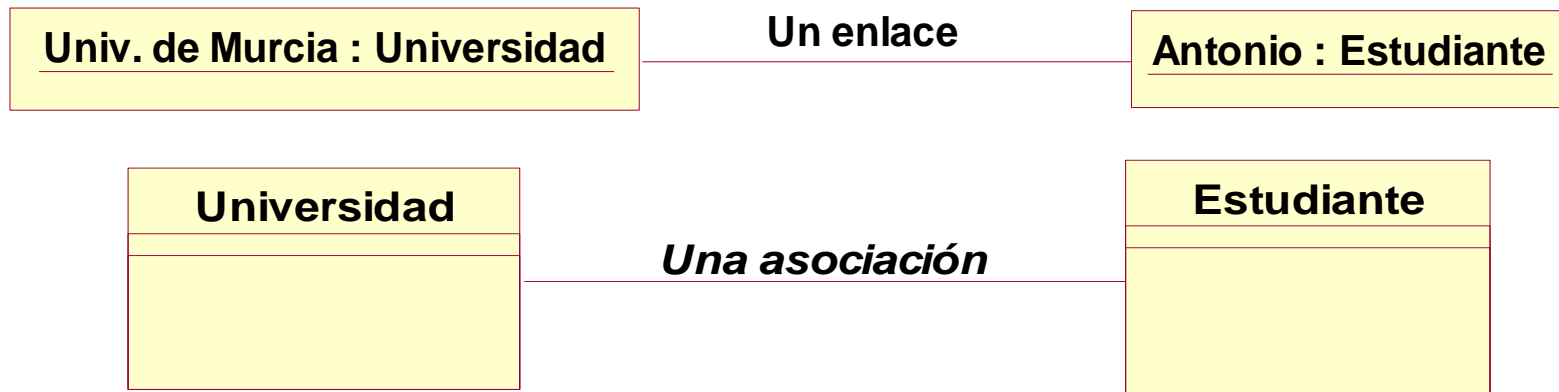
Persona

- nombre: String
- fechaDeNacimiento: Date
- # direccion: String
- + telefono: String

- + calcularEdad(Date) : void
- + setNombre(String) : void
- + getNombre() : String

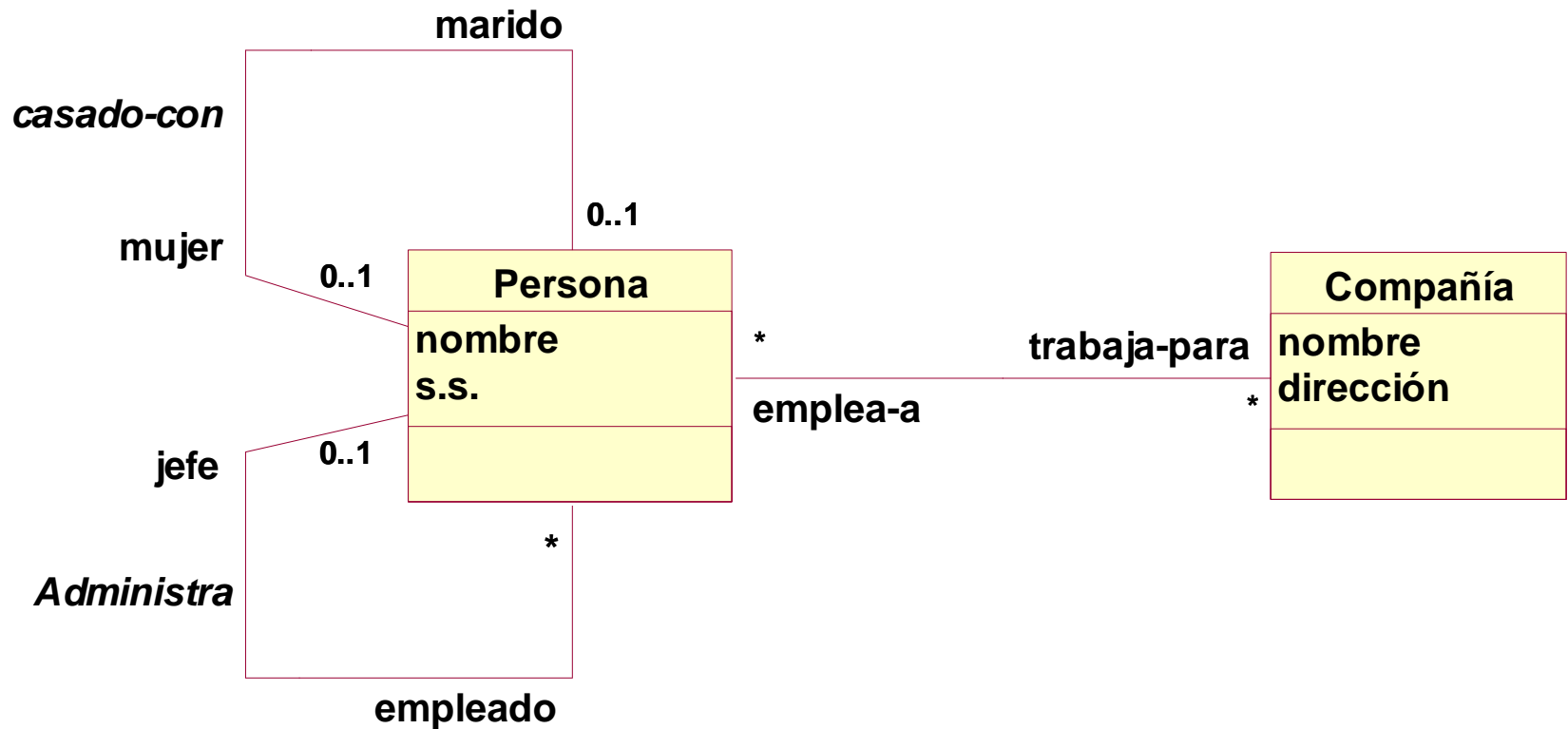
Asociación

- La asociación expresa una conexión bidireccional entre objetos
- Una asociación es una abstracción de la relación existente en los enlaces entre los objetos



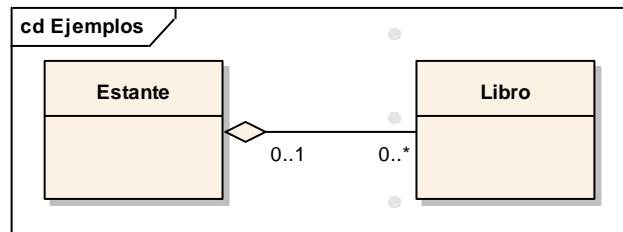
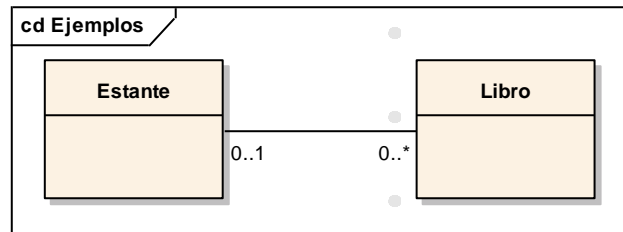
... Asociación

- Ejemplo:



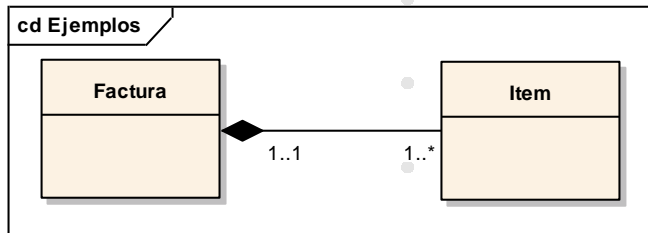
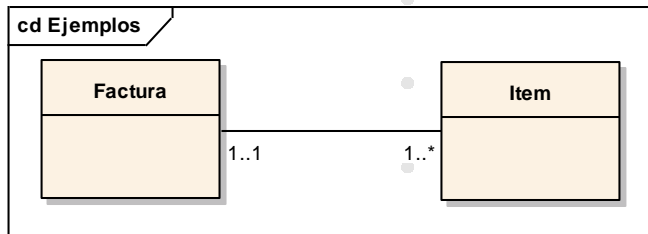
Agregación

- La agregación representa una relación en que cada uno puede existir por separado
- Un ejemplo de esto es estante libro

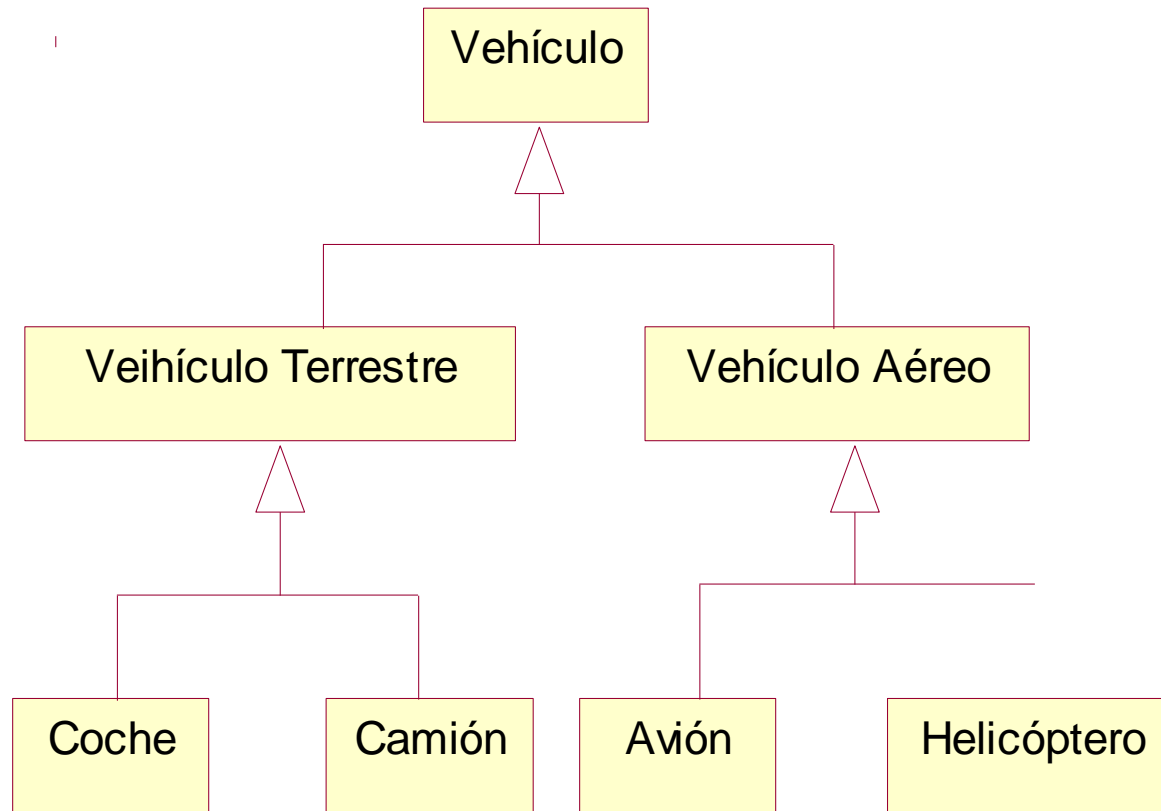


Composición

- La composición representa el caso que un objeto no puede existir sin el otro
- En este caso podemos ir al ejemplo de Factura – ítem

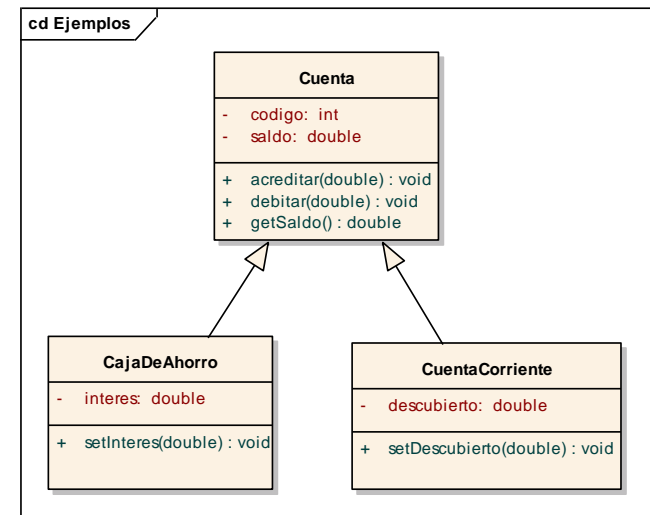
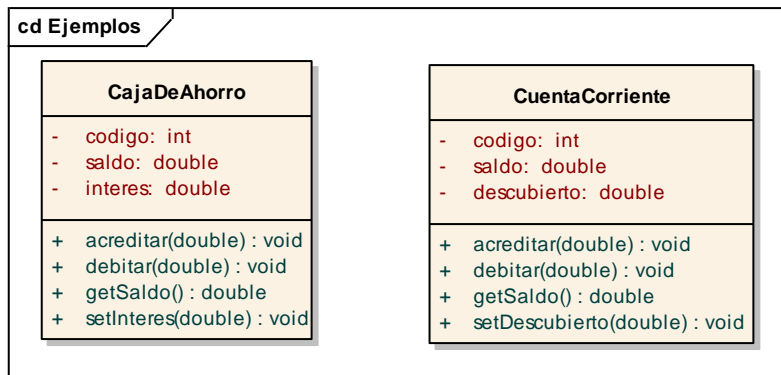


... Generalización



... Generalización

- Esta relación se da cuando existe un conjunto de atributos que se repiten en la clases y lleva a la generación de un padre
- Por esa razón las clases hijas tienen todo lo que tiene el padre mas las particularidades



Polimorfismo

- El término polimorfismo se refiere a que una característica de una clase puede tomar varias formas
- El polimorfismo representa en nuestro caso la posibilidad de desencadenar operaciones distintas en respuesta a un mismo mensaje
- Cada subclase hereda las operaciones pero tiene la posibilidad de modificar localmente el comportamiento de estas operaciones

Polimorfismo

- Ejemplo: toda cuenta debita de forma distinta, la caja de ahorro debita hasta llegar a 0 y la cuenta corriente debita hasta llegar al descubierto

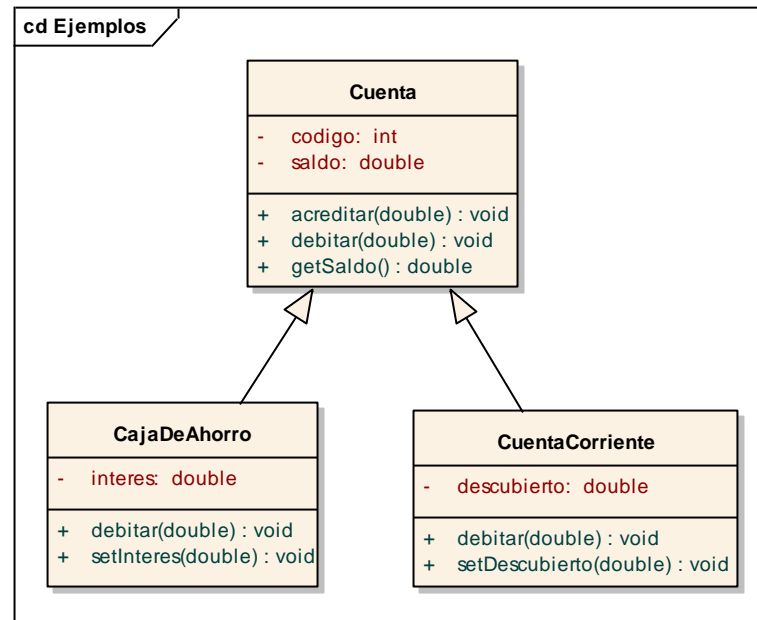


Diagrama de Secuencia

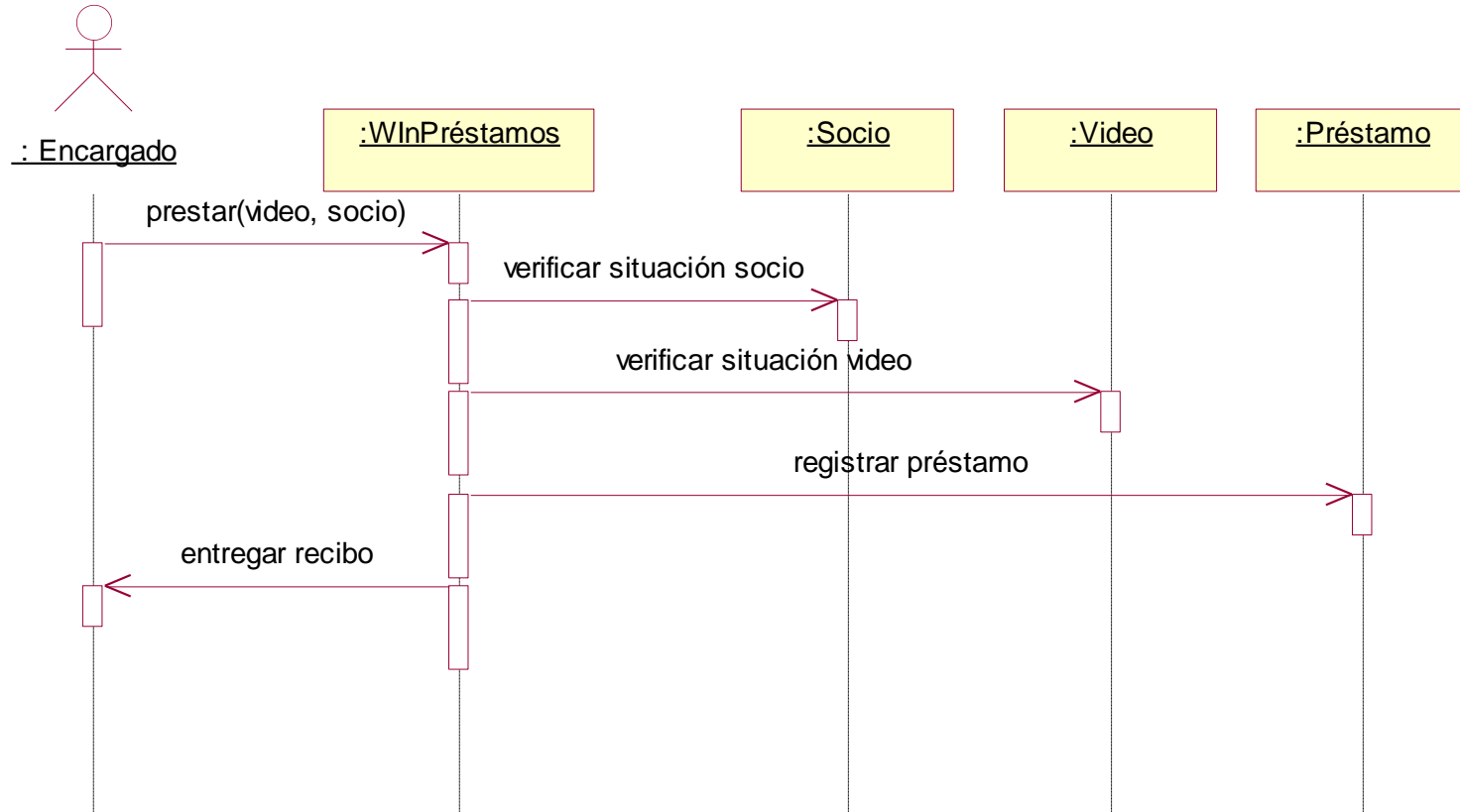


Diagrama de Colaboración

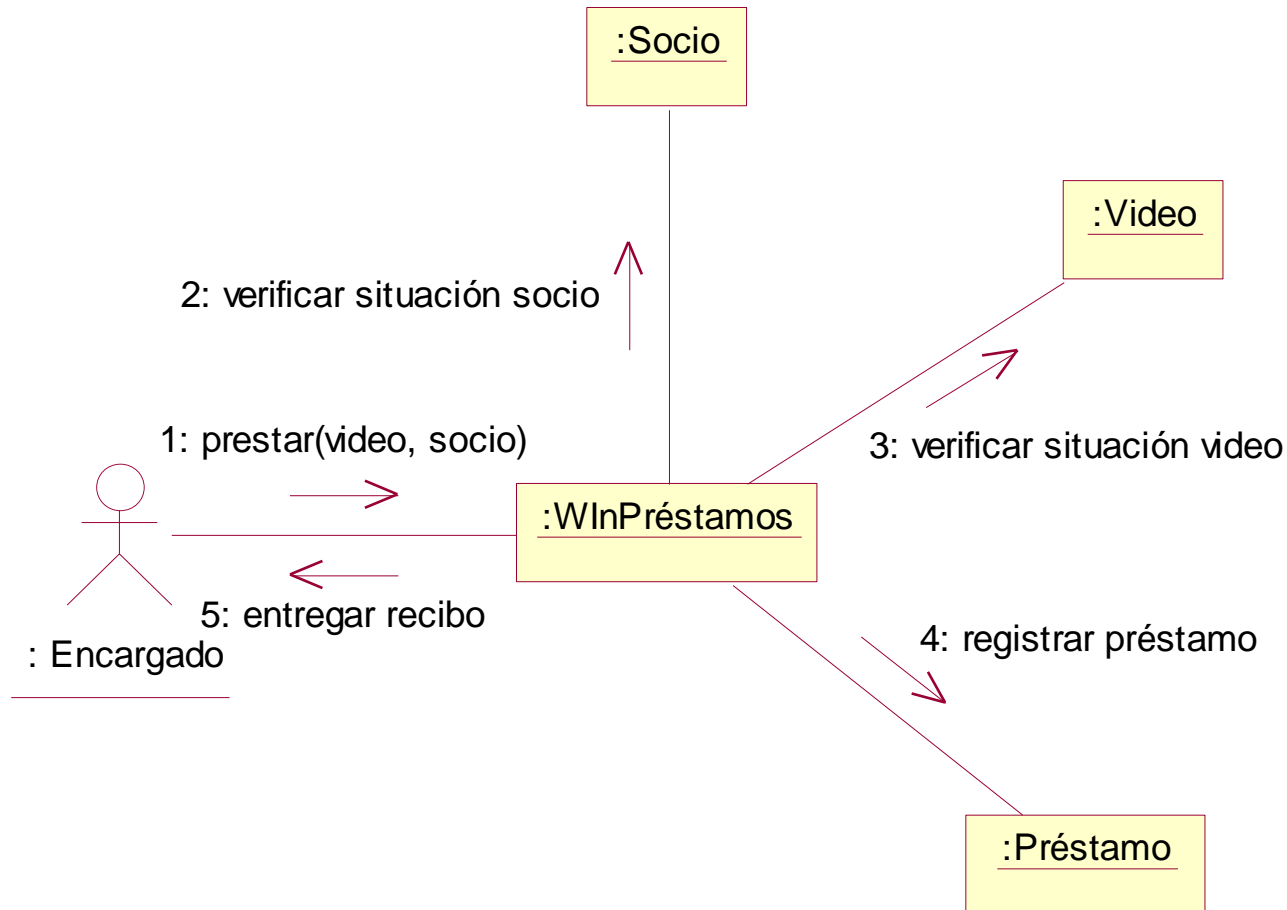


Diagrama de Estados

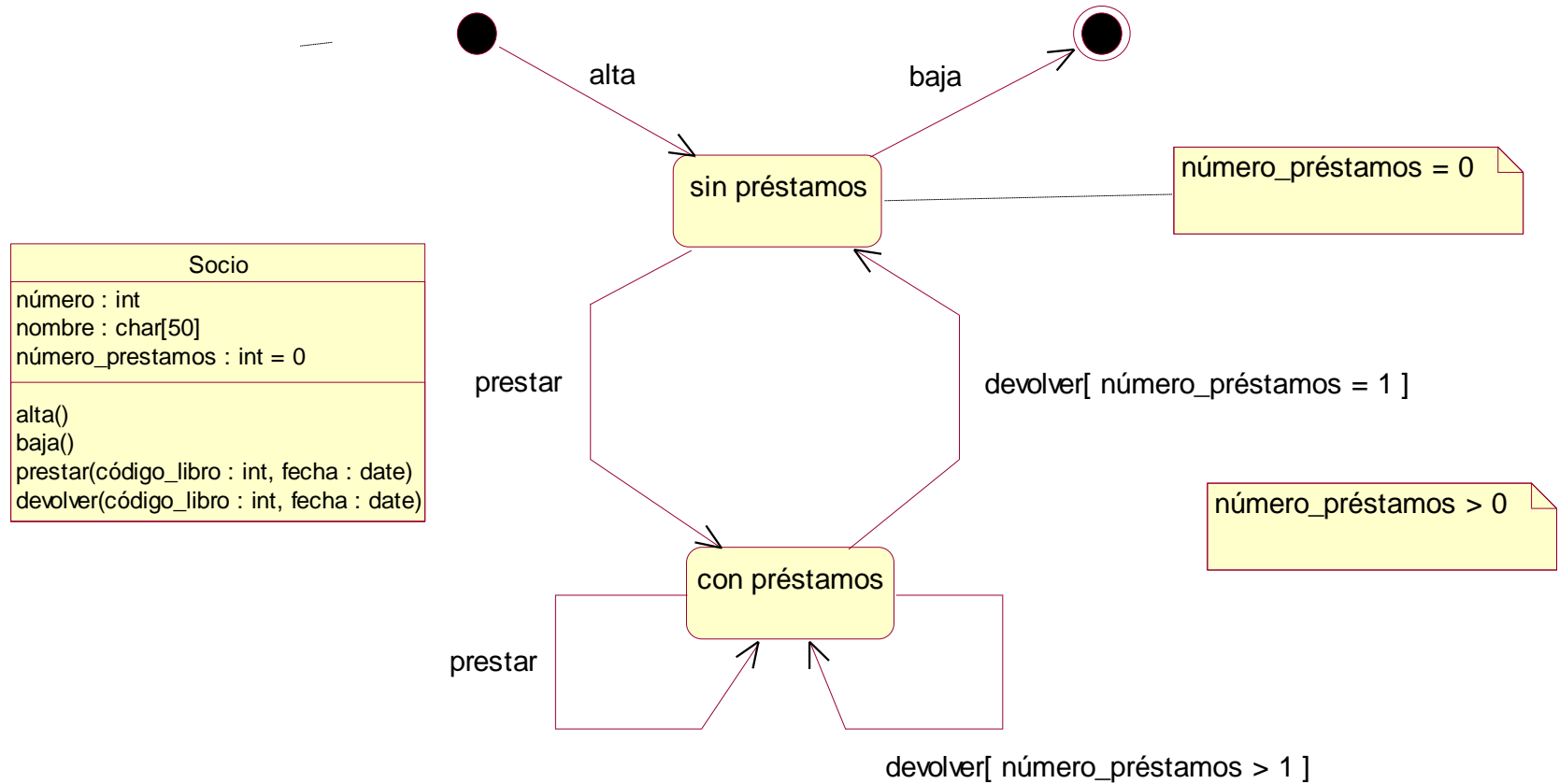


Diagrama Componentes

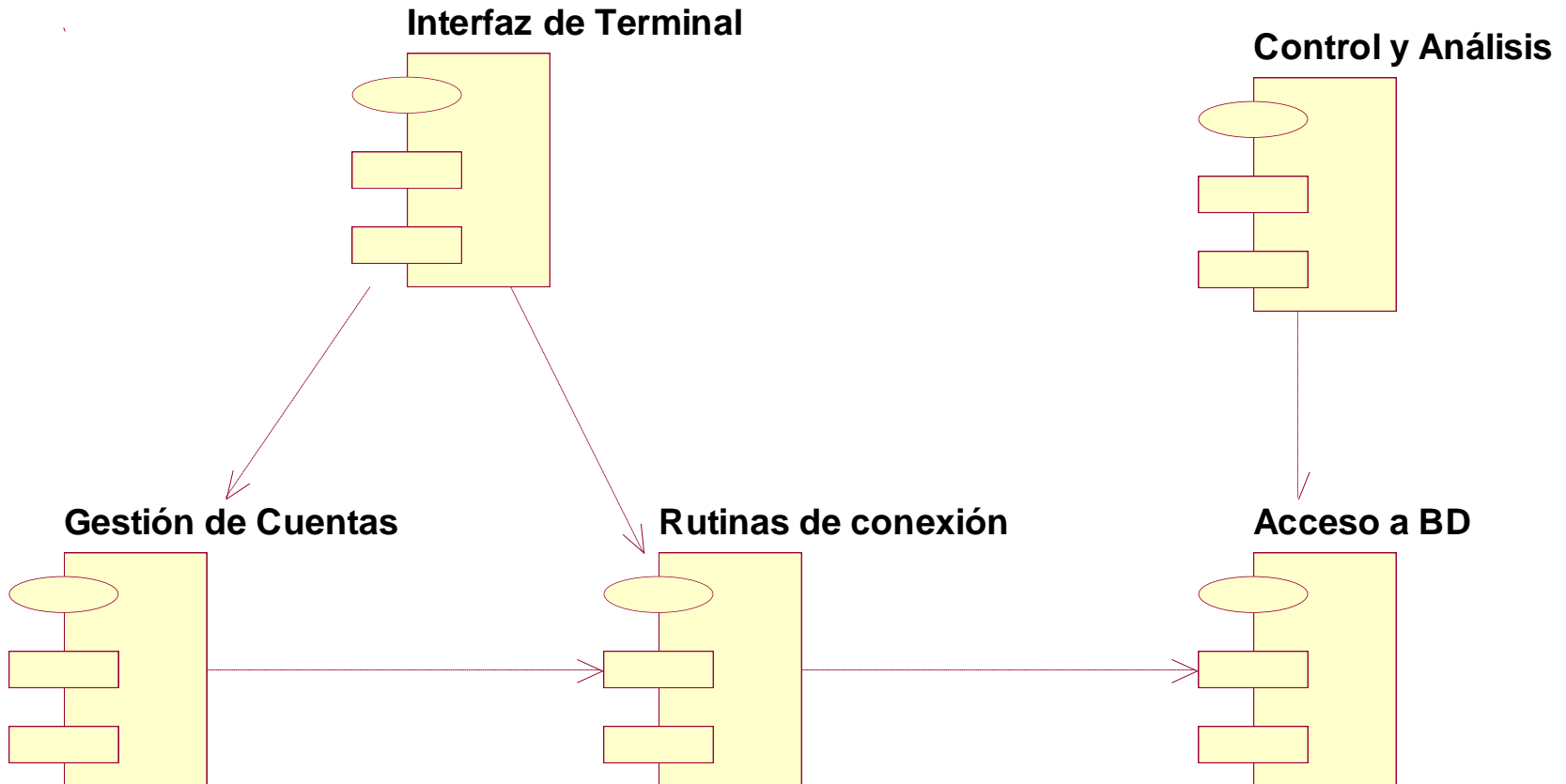
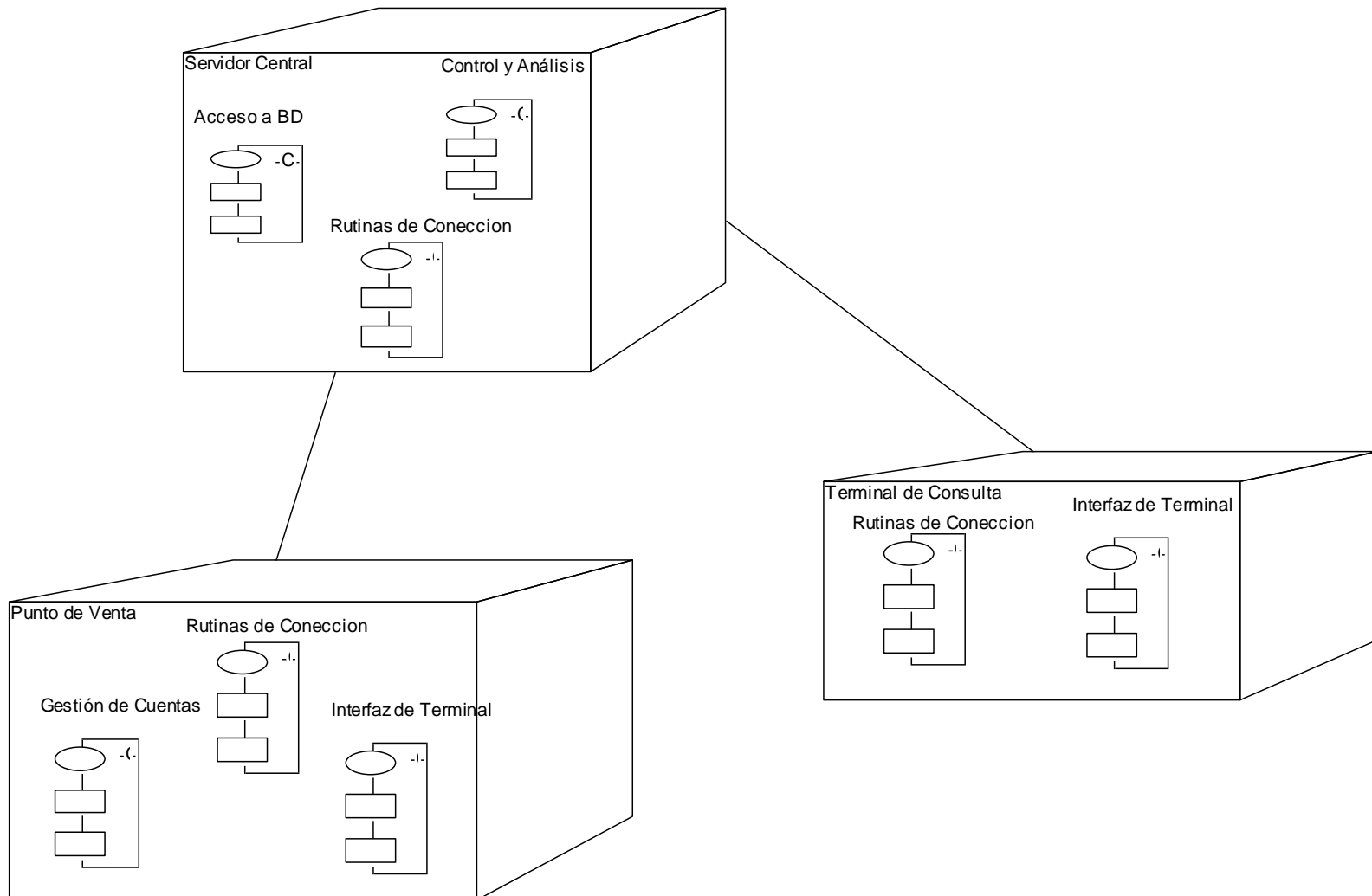


Diagrama de Despliegue



Resumen

- UML define una notación que se expresa como diagramas sirven para representar modelos/subsistemas o partes de ellos
- *El 80 por ciento de la mayoría de los problemas pueden modelarse usando alrededor del 20 por ciento de UML-- Grady Booch*