

Análisis y Diseño Orientado a Objetos

The slide features a title centered in the upper half. Below the title, there are five circles of varying shades of light purple. Two circles are solid, and three are hollow outlines. They are arranged in a way that some overlap the text area.

Sistema

- Un sistema puede ser definido como un complejo de **elementos** interactuantes. Interacción significa que elementos, p , están en relaciones, R , de suerte que el comportamiento de un elemento p en R es diferente de su comportamiento en otra relación R' . Si los comportamientos en R y R' no difieren, no hay interacción, y los elementos se comportan independientemente con respecto a las relaciones R y R' .

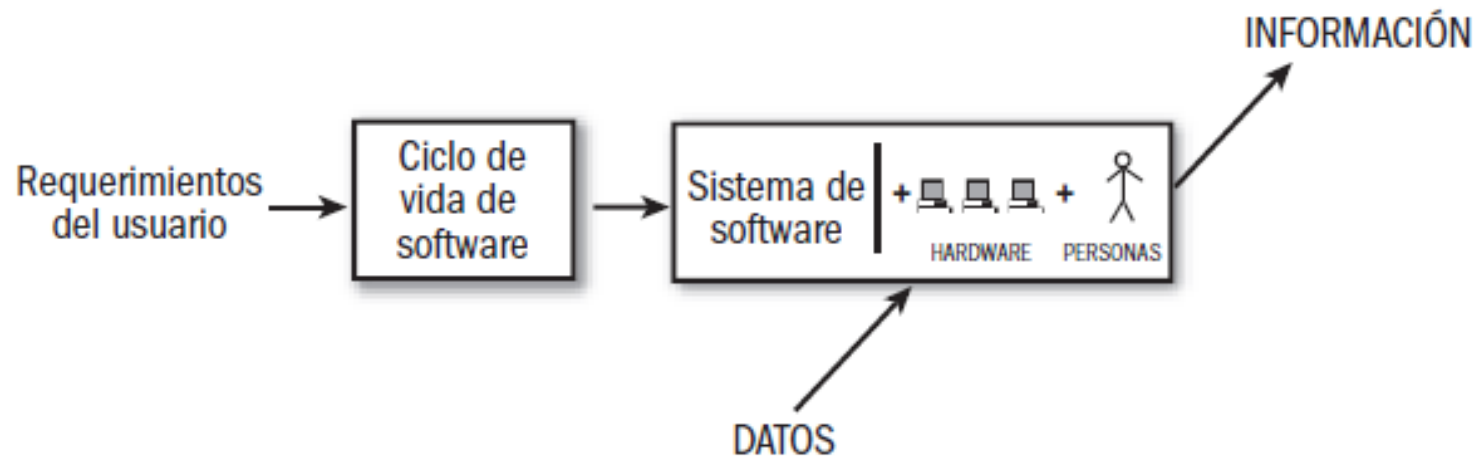
Software



- Software es lo que se denomina producto en Ingeniería de Software.
- El estándar IEEE 610.12-1990 define producto software:

software product. (1) The complete set of computer programs, procedures, and possibly associated documentation and data designated for delivery to a user.
(2) Any of the individual items in (1).

Sistema informático



Proceso de desarrollo (creación) de software

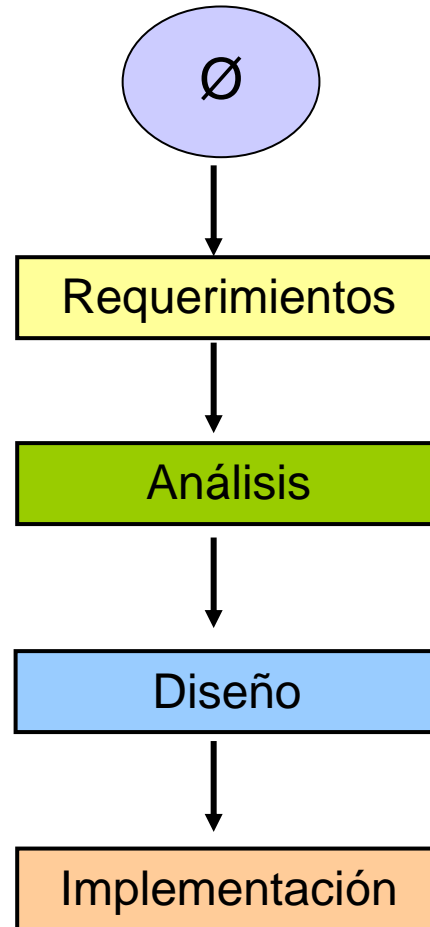
- Proceso: conjunto ordenado de pasos a seguir para llegar a la solución de un problema u obtención de un producto, en este caso particular, para lograr un producto software que resuelva un problema específico.
- Los procesos de desarrollo de software poseen reglas preestablecidas.



Ciclo de vida del software

- ISO/IEC 12207:2008

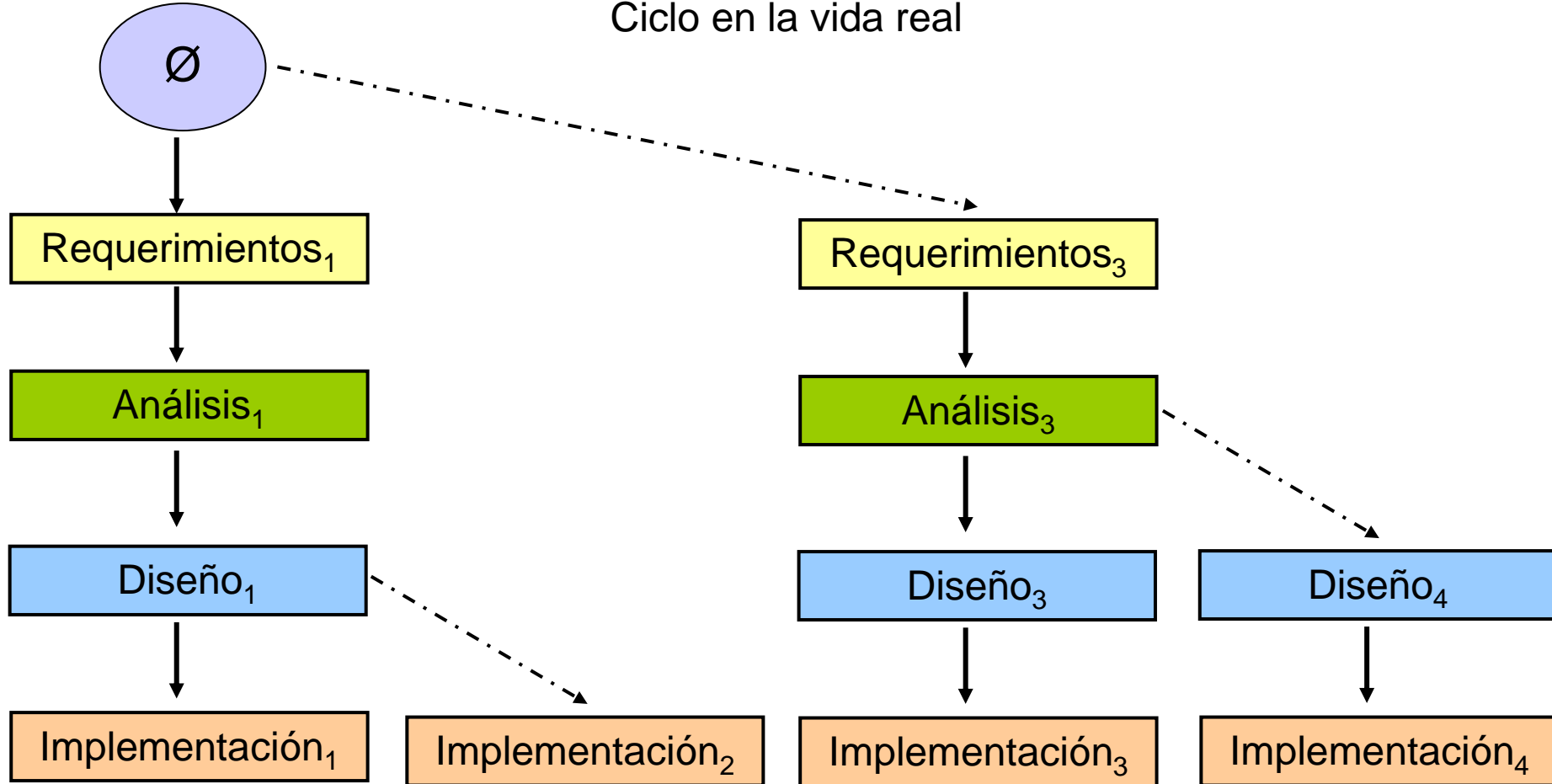
Modelos del ciclo de vida del SW



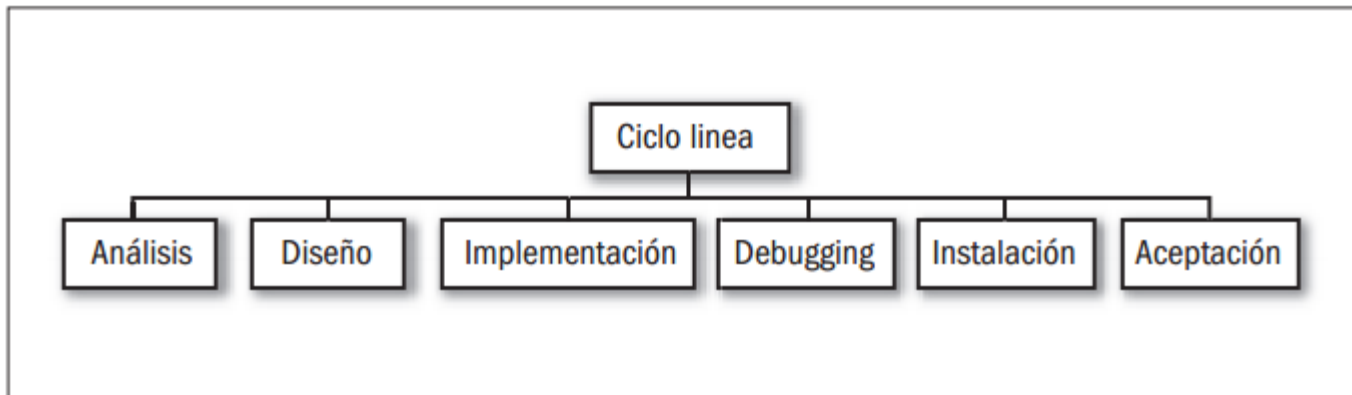
Ciclo idealizado

Modelos del ciclo de vida del SW

Ciclo en la vida real

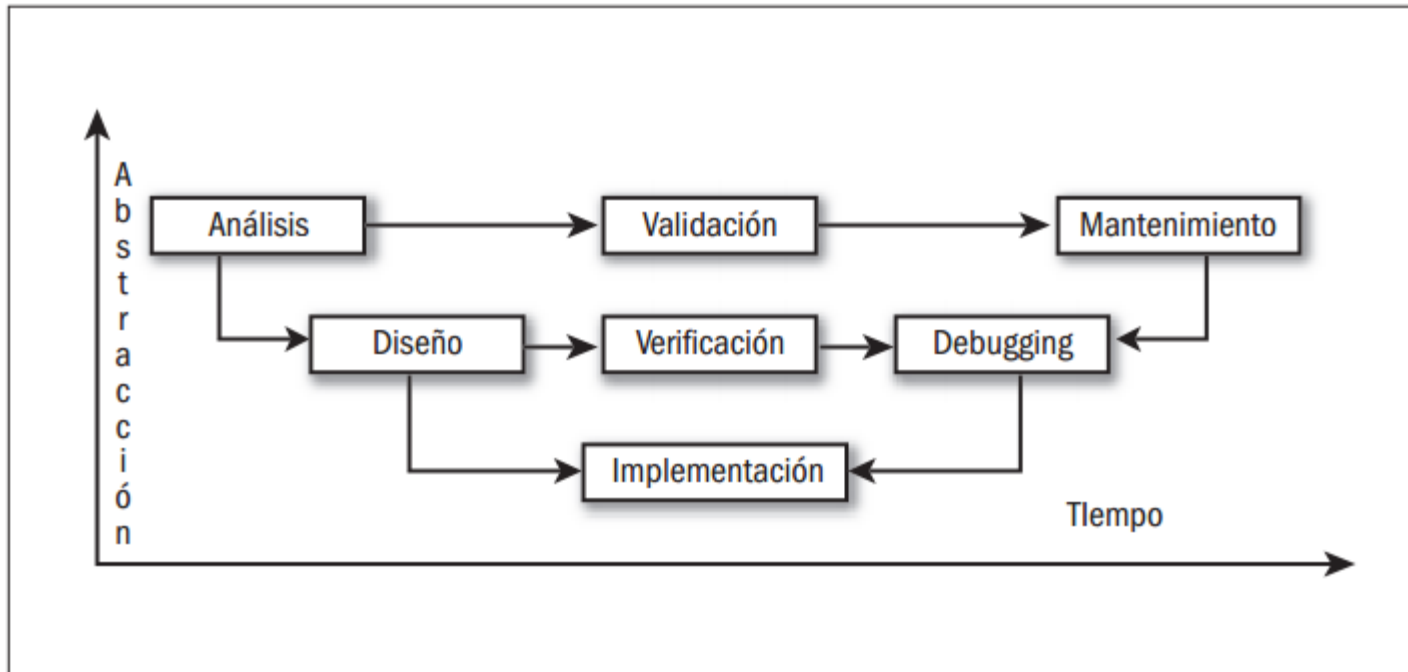


Modelo lineal

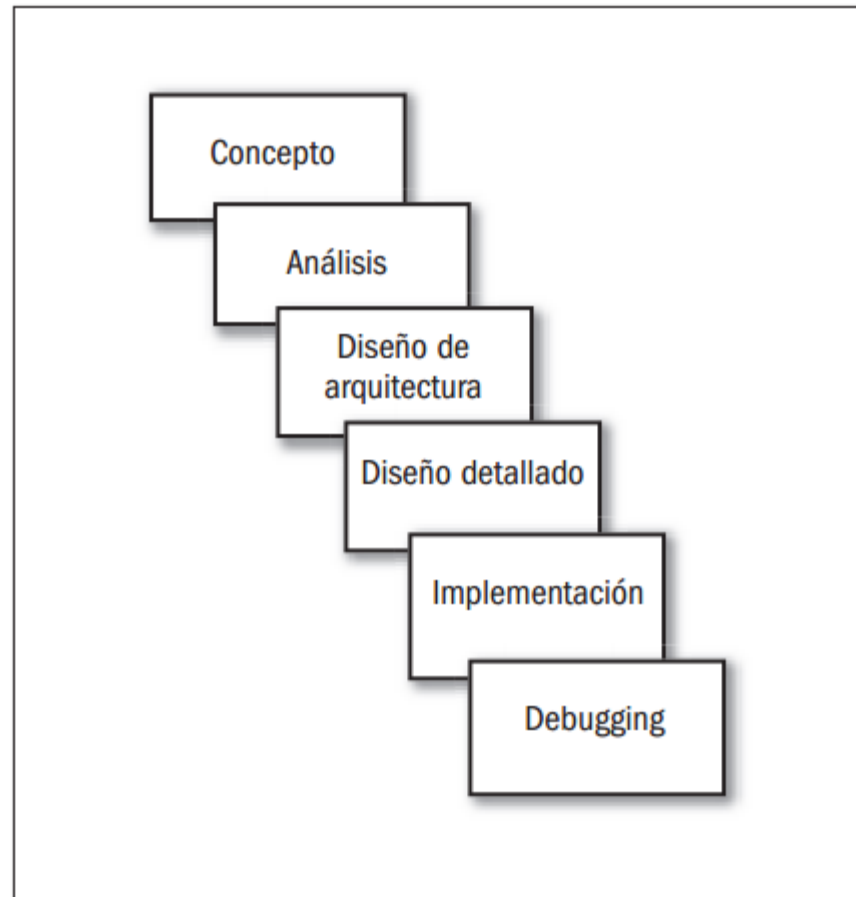




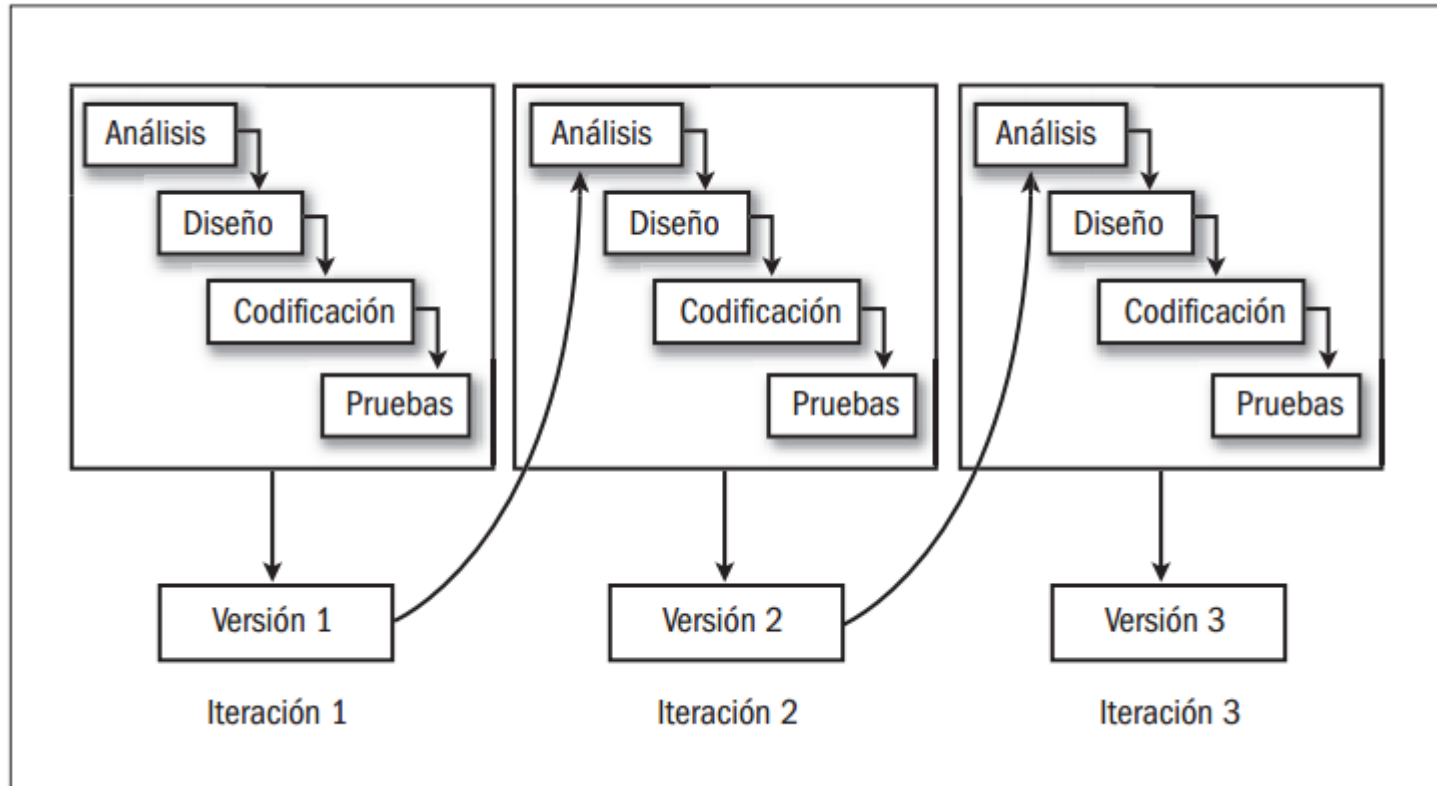
Modelo en V



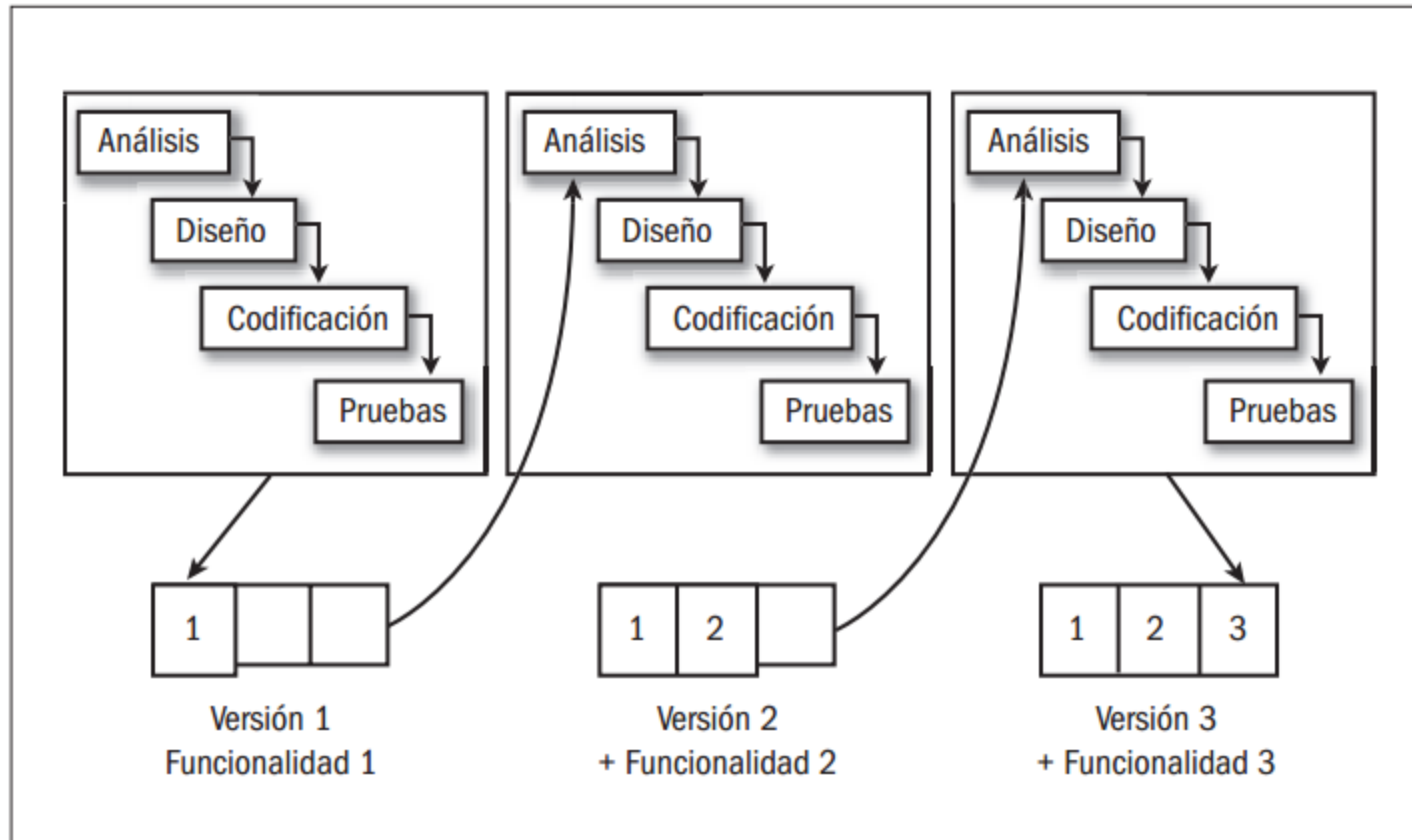
Modelo en sashimi



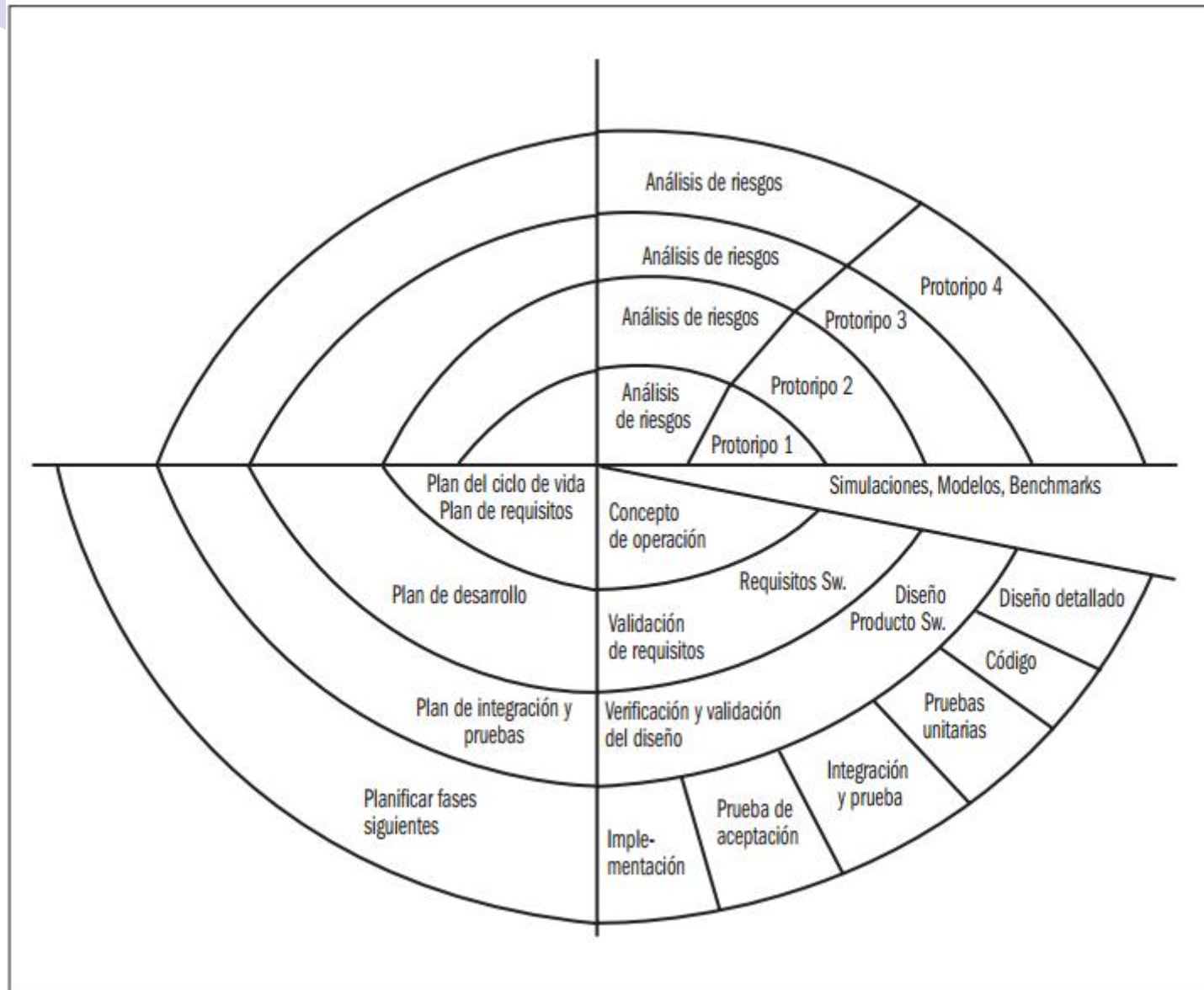
Modelo iterativo



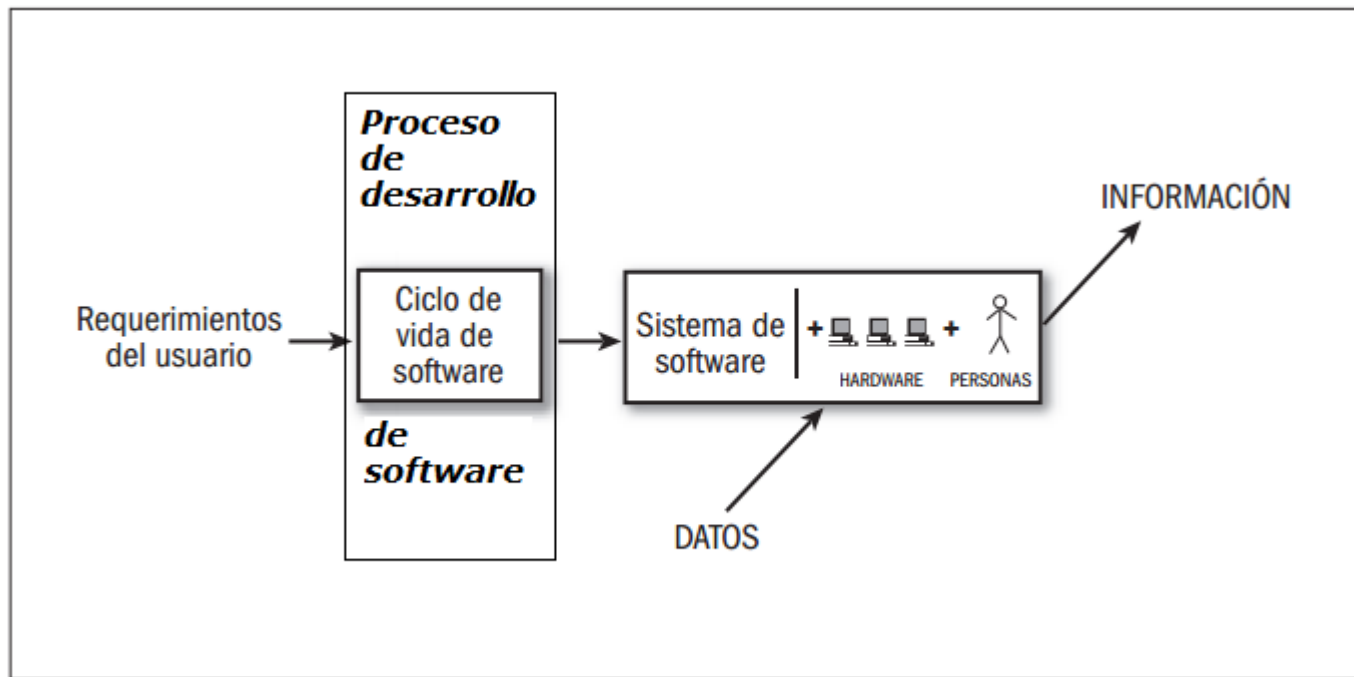
Modelo incremental



Modelo en espiral



El proceso del software



El proceso del software



- Es el modo en que se produce software.
- Incorpora una metodología y su modelo del ciclo de vida de software que la sustenta, además de las personas (roles), técnicas y herramientas necesarias para llevarlo a cabo.
- Es el conjunto de actividades necesarias para transformar los requerimientos de un usuario en un sistema software.

El proceso del software



- Proporciona una guía para ordenar las actividades de un equipo.
- Dirige las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifica los artefactos que deben desarrollarse.
- Ofrece criterios para el control y la medición de los productos y actividades del proyecto.

El Proceso Unificado



- Principal metodología orientada a objetos.
- Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

El Proceso Unificado



- Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.
- Utiliza el UML para preparar todos los esquemas de un sistema software. UML es una parte esencial del PU ya que sus desarrollos fueron paralelos.
- Es dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

El Proceso Unificado

EL PROCESO UNIFICADO: DIRIGIDO POR CASOS DE USO, CENTRADO EN LA ARQUITECTURA... 5

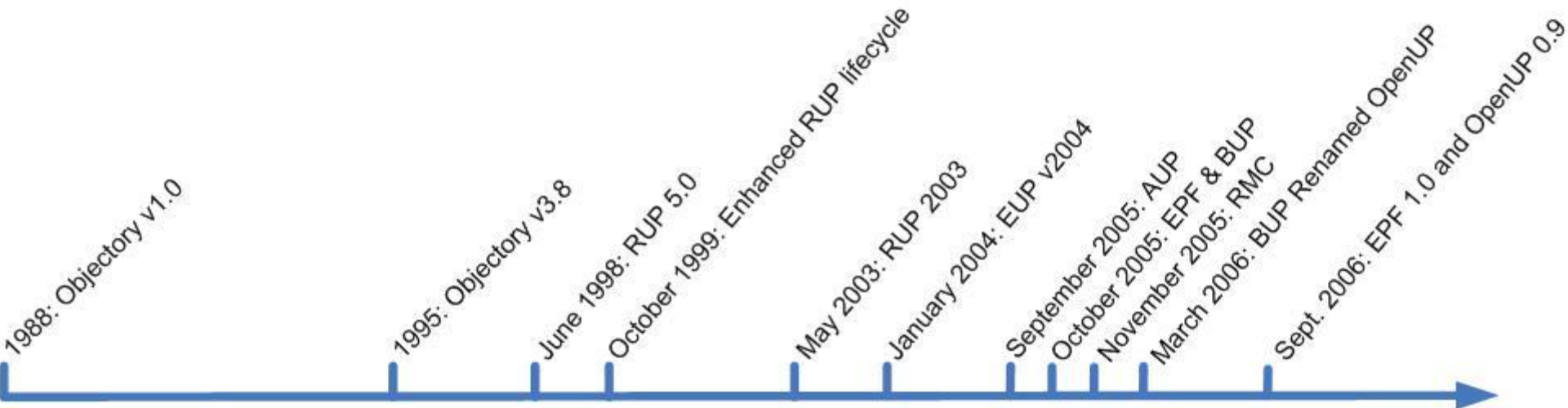
1.2. El Proceso Unificado está dirigido por casos de uso

Un sistema software ve la luz para dar servicio a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean.

El Proceso Unificado

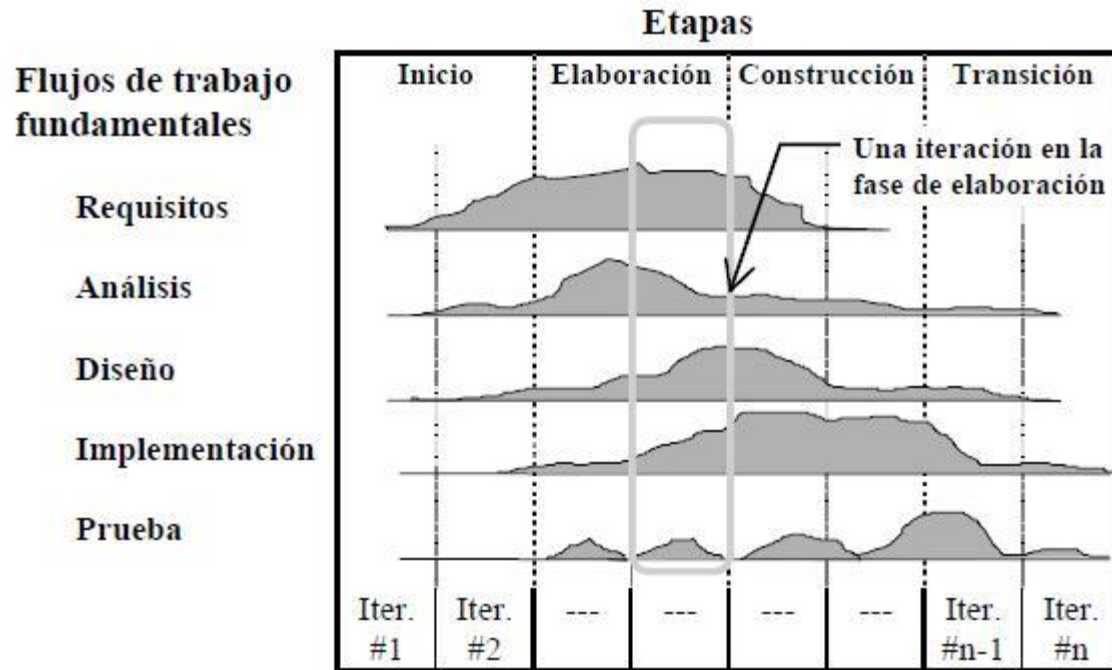
Una interacción de este tipo es un **caso de uso** (Apéndice A; véase también el Capítulo 3). Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el **modelo de casos de uso** (Apéndice B; véase también la Sección 2.3), el cual describe la funcionalidad total del sistema. Puede decirse que una especificación funcional contesta a la pregunta: ¿Qué debe hacer el sistema?. La estrategia de los casos de uso puede describirse añadiendo tres palabras al final de esta pregunta: *¿...para cada usuario?* Estas tres palabras albergan una implicación importante. Nos fuerzan a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno tener. Sin embargo, los casos de uso no son sólo una herramienta para especificar los requisitos de un sistema. También guían su diseño, implementación, y prueba; esto es, *guían el proceso de desarrollo*. Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que llevan a cabo los casos de uso. Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conformes al modelo de casos de uso. Los ingenieros de prueba prueban la implementación para garantizar que los componentes del modelo de implementación implementan correctamente los casos de uso. De este modo, los casos de uso no sólo inician el proceso de desarrollo sino que le proporcionan un hilo conductor. Dirigido por casos de uso quiere decir que el proceso de desarrollo sigue un hilo —avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan, y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba.

Proceso Unificado

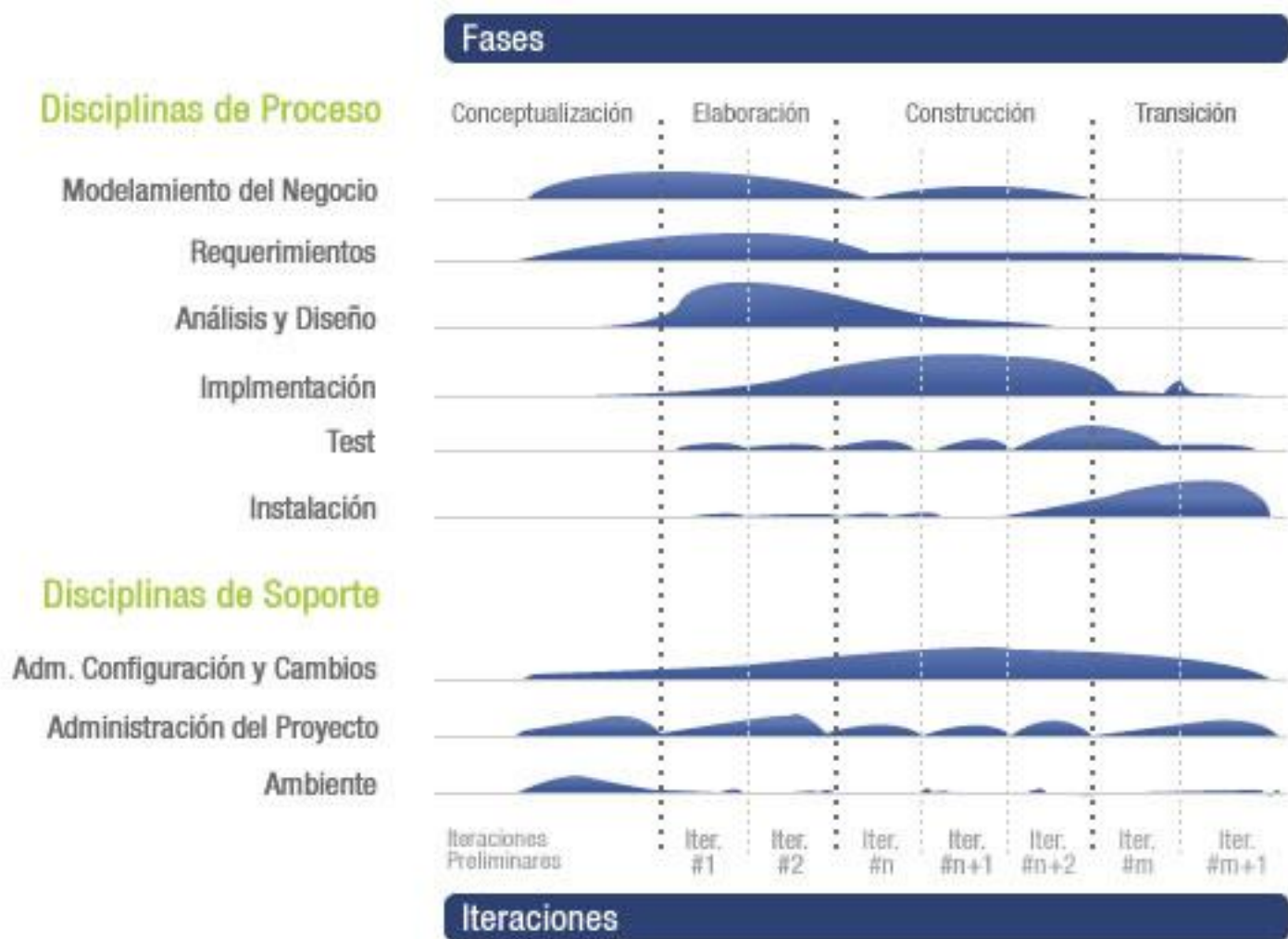


Copyright 2005-2006 Scott W. Ambler

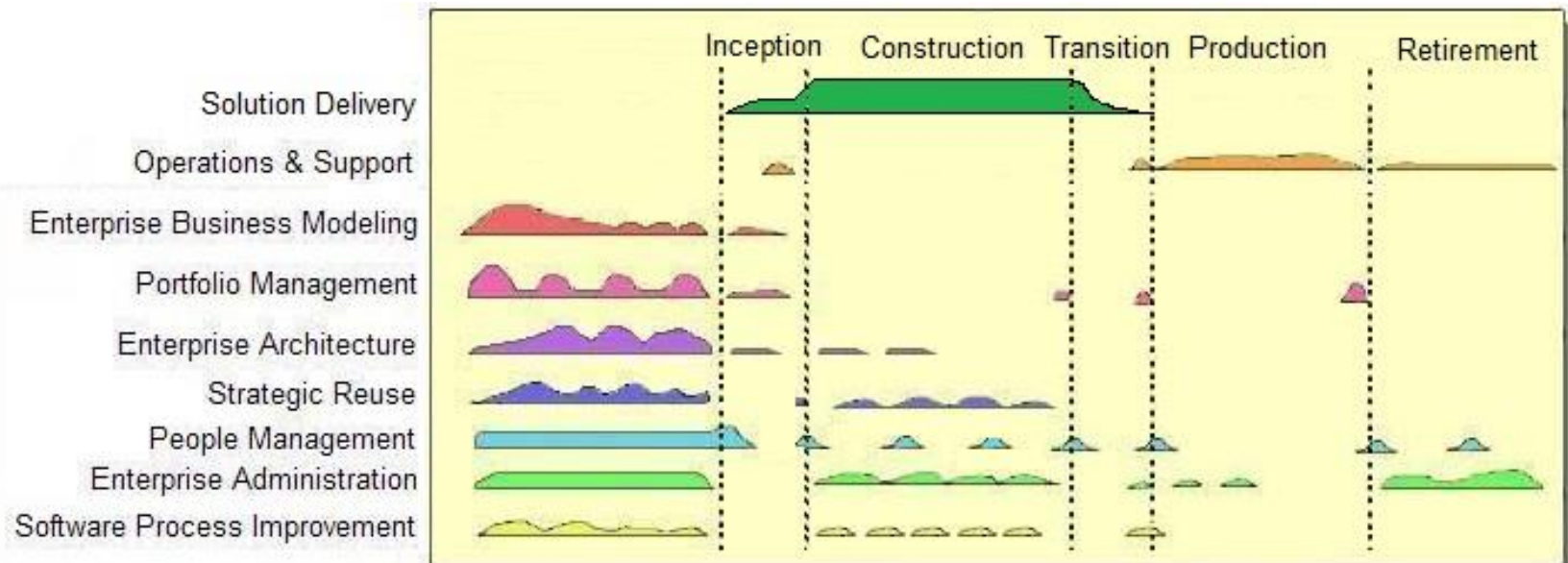
El Proceso Unificado



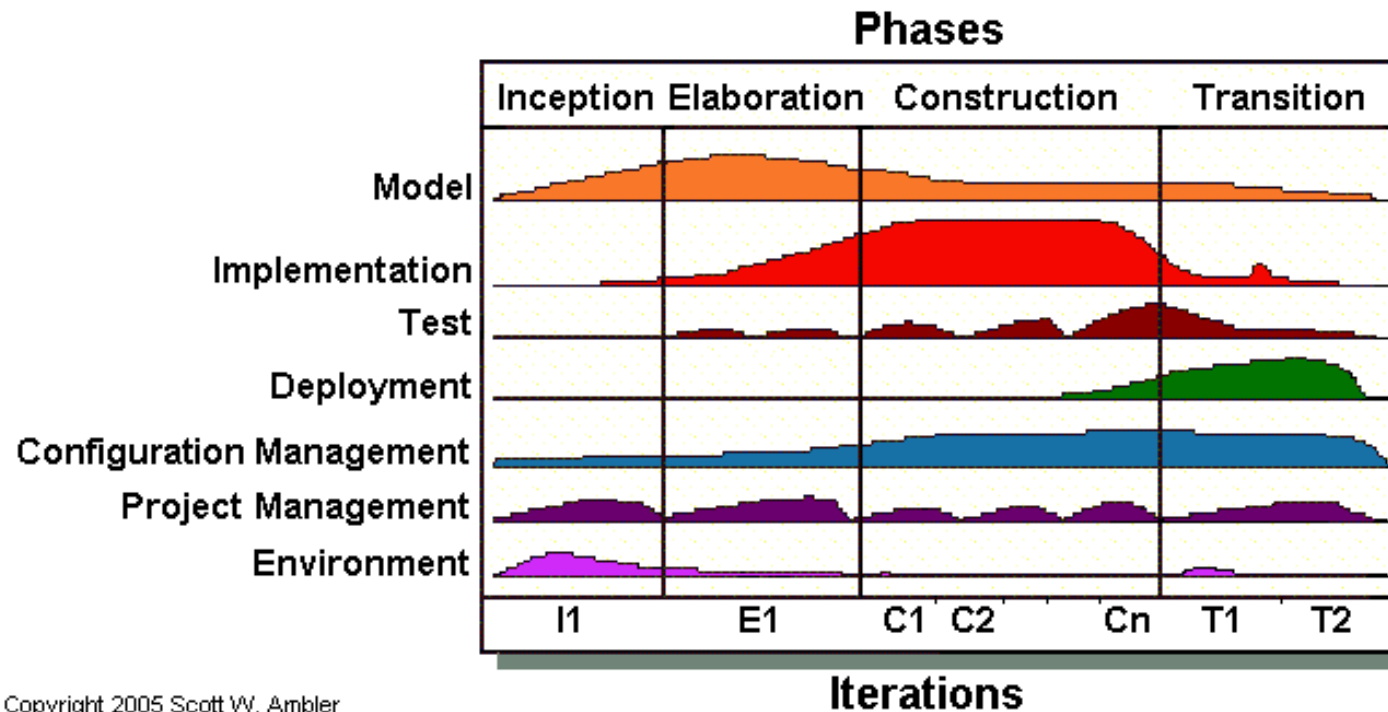
El Proceso Unificado de Rational



Enterprise Unified Process

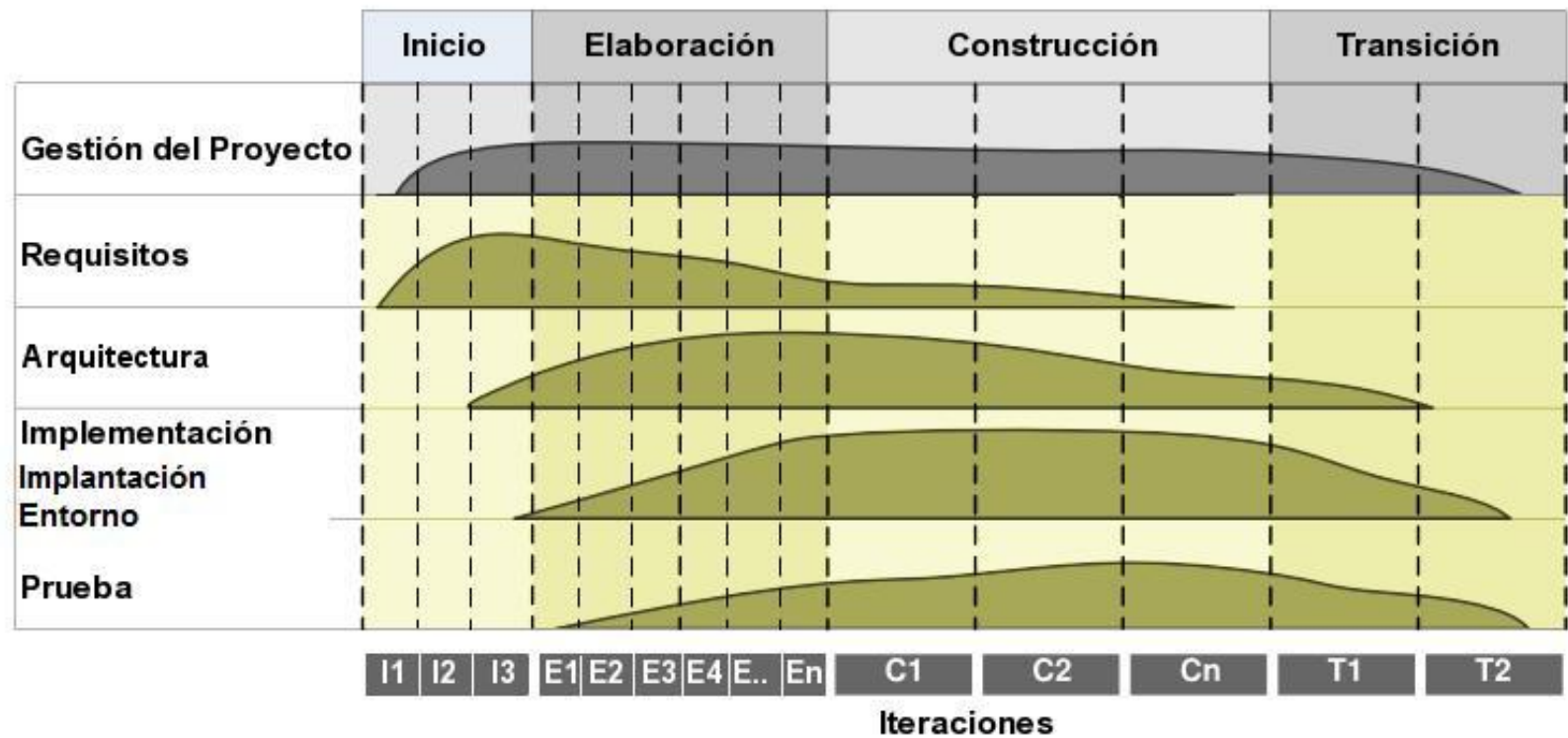


Agile Unified Process

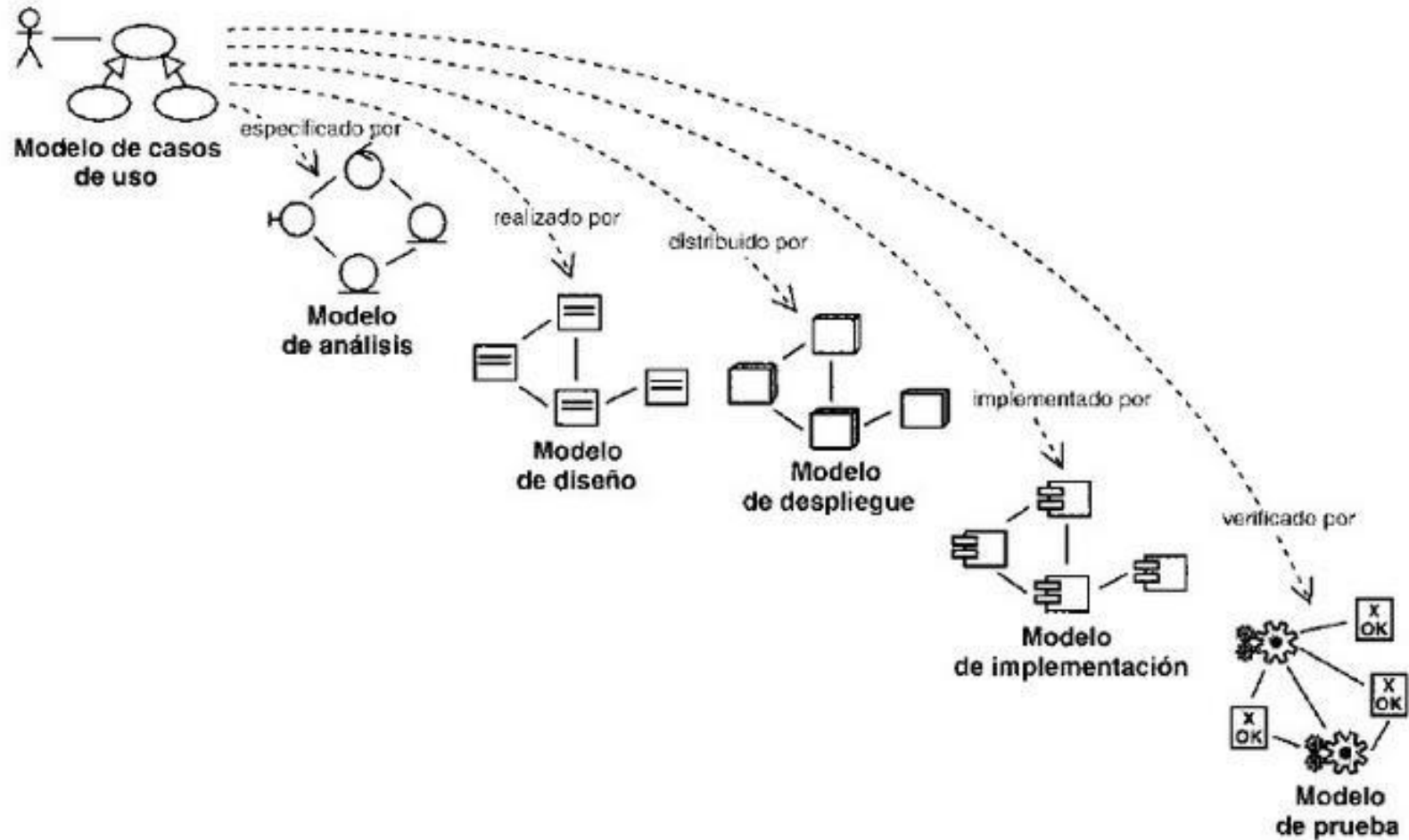


Copyright 2005 Scott W. Ambler

Open UP



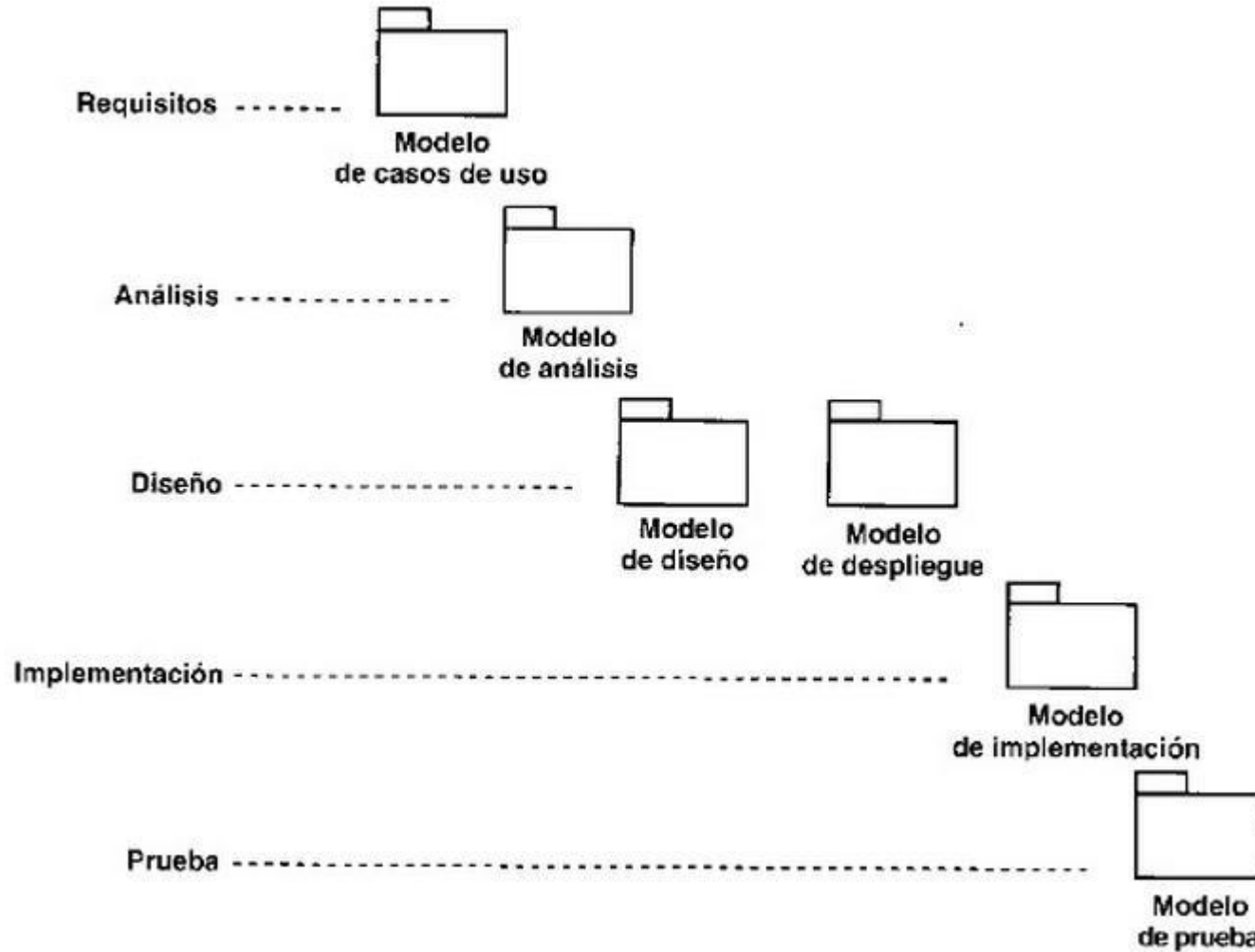
Modelo general del PU



Roles



Colección de modelos

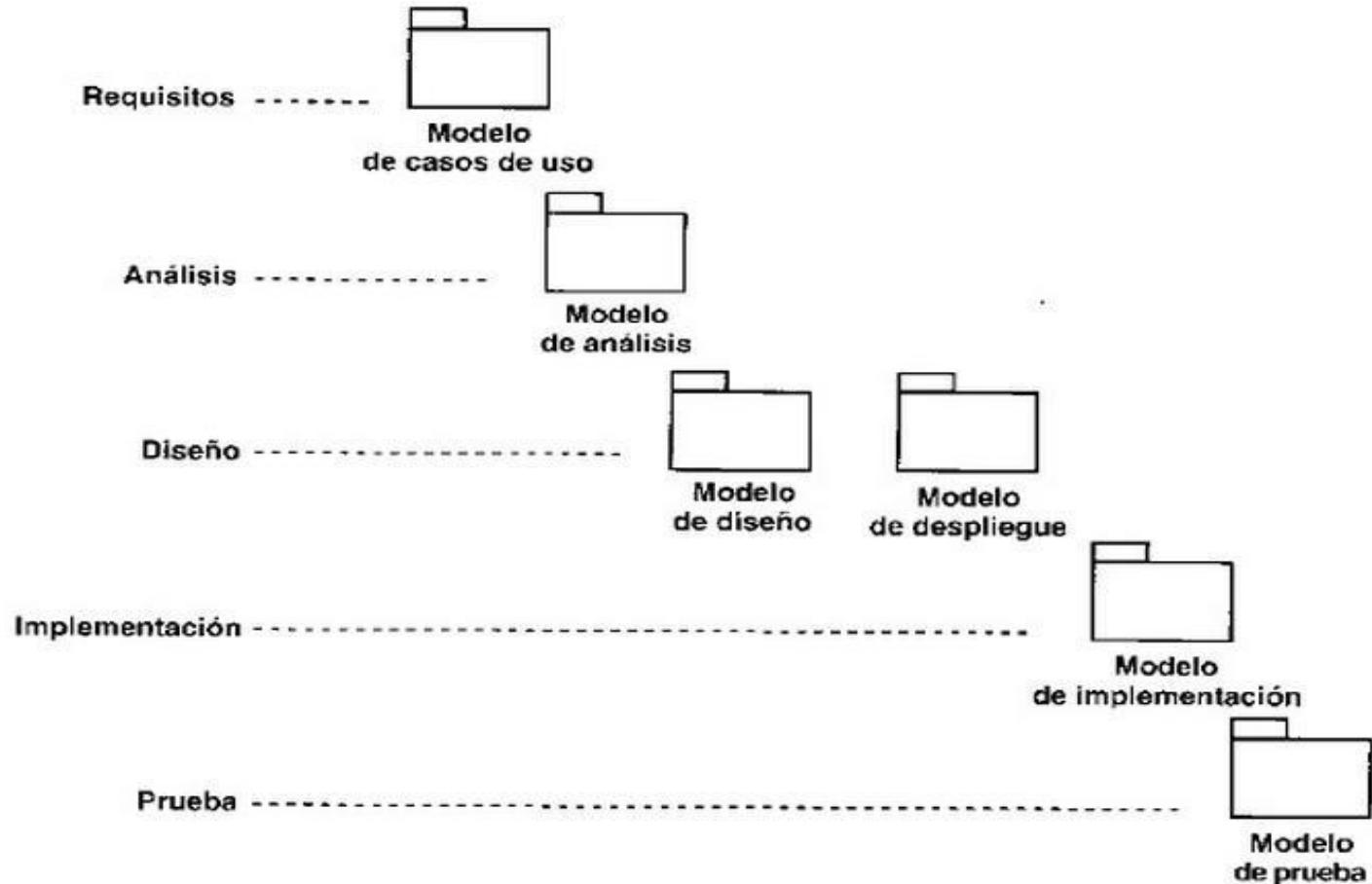


Colección de modelos

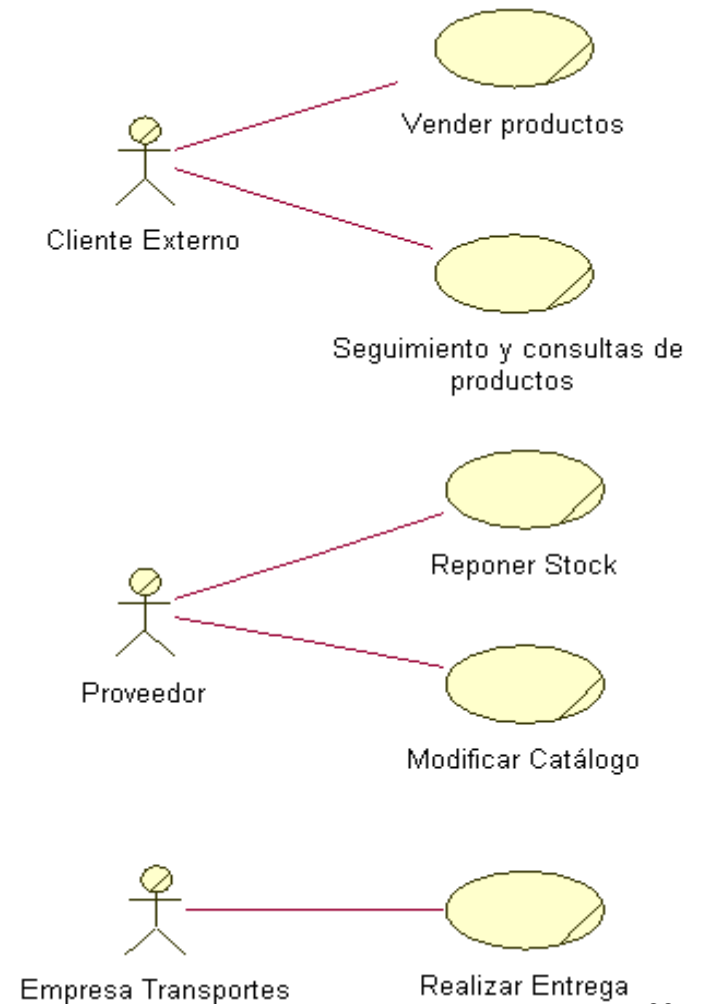
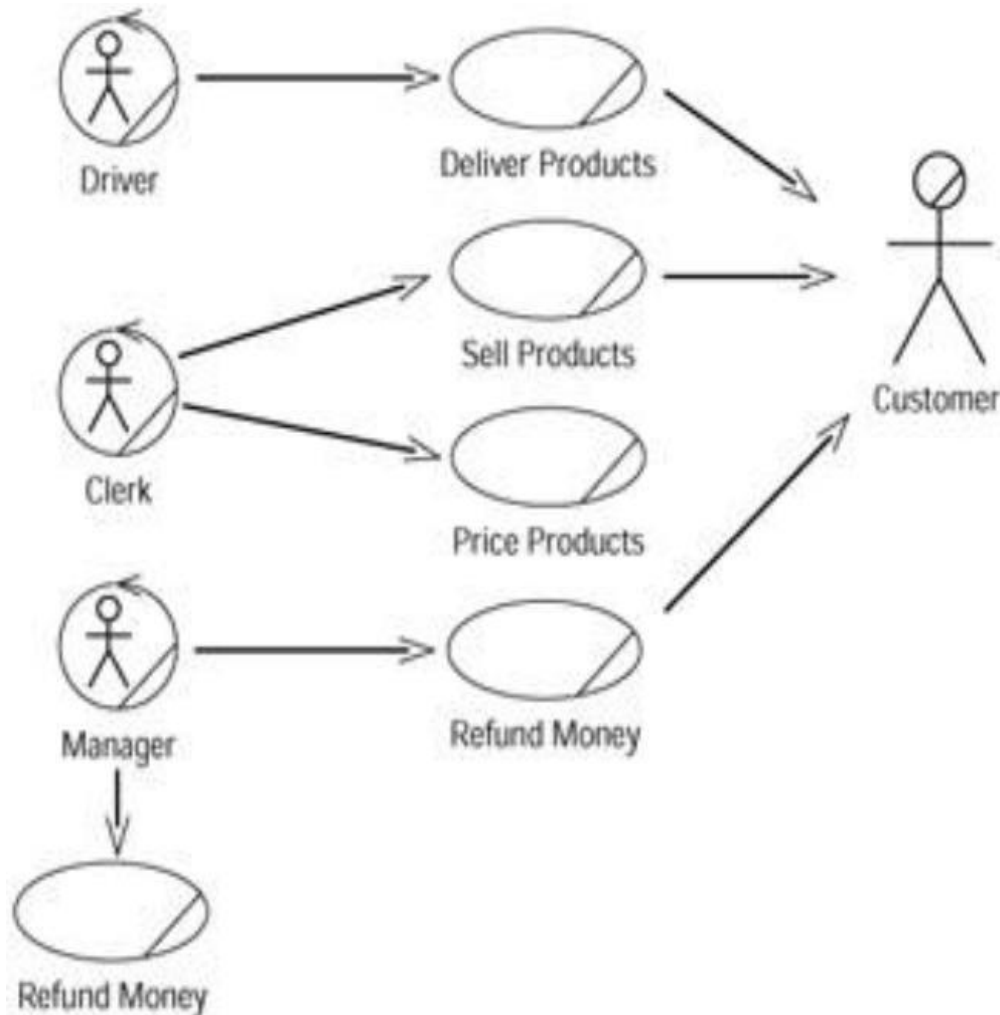
Modelamiento del negocio

Modelo de negocio

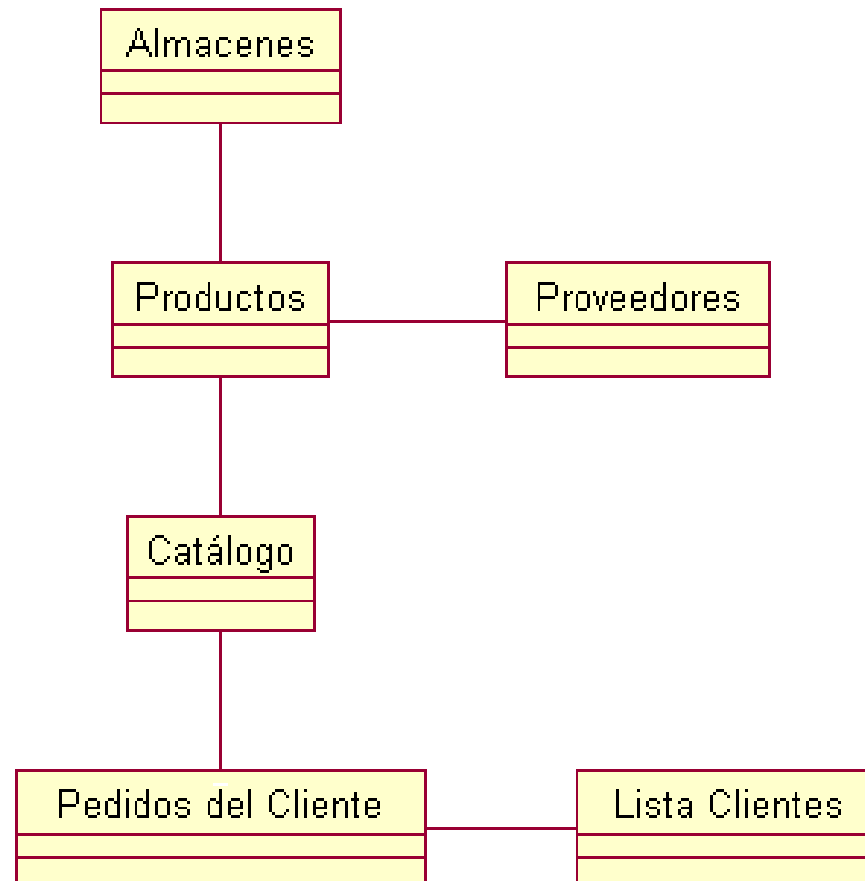
Modelo de dominio



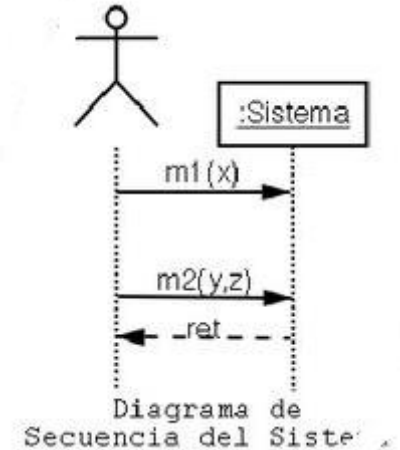
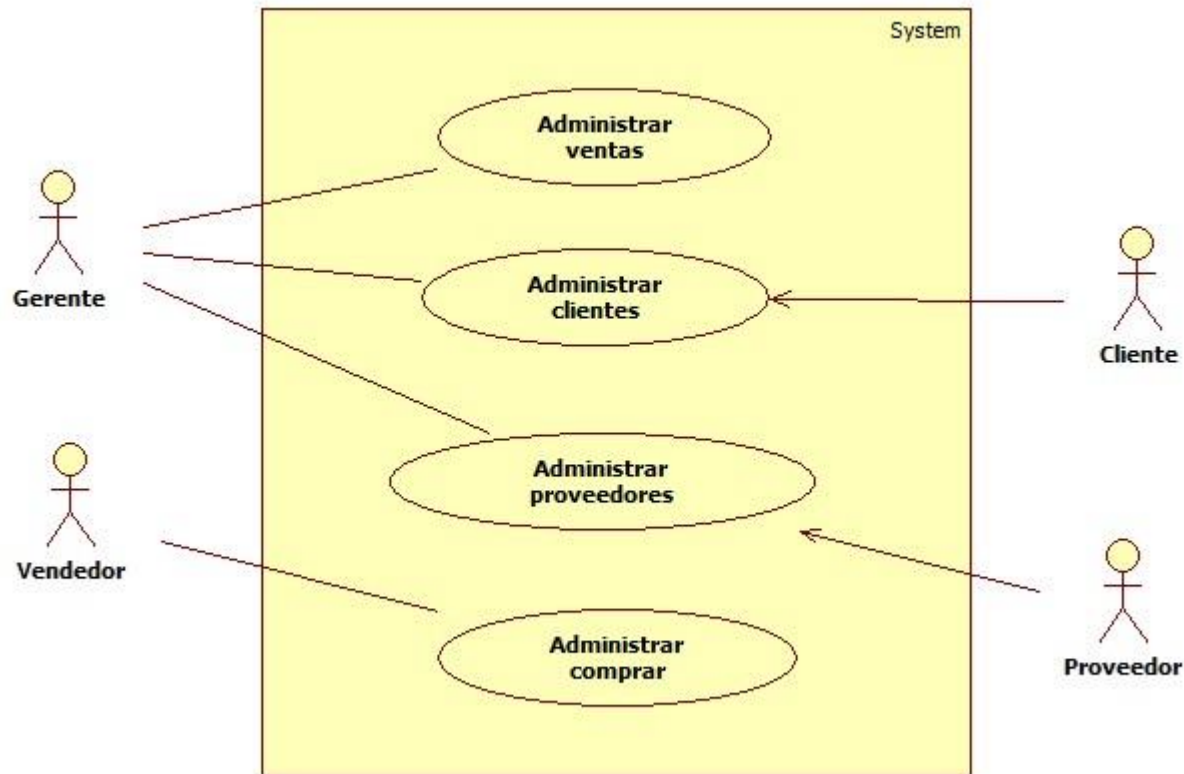
Modelo de negocio



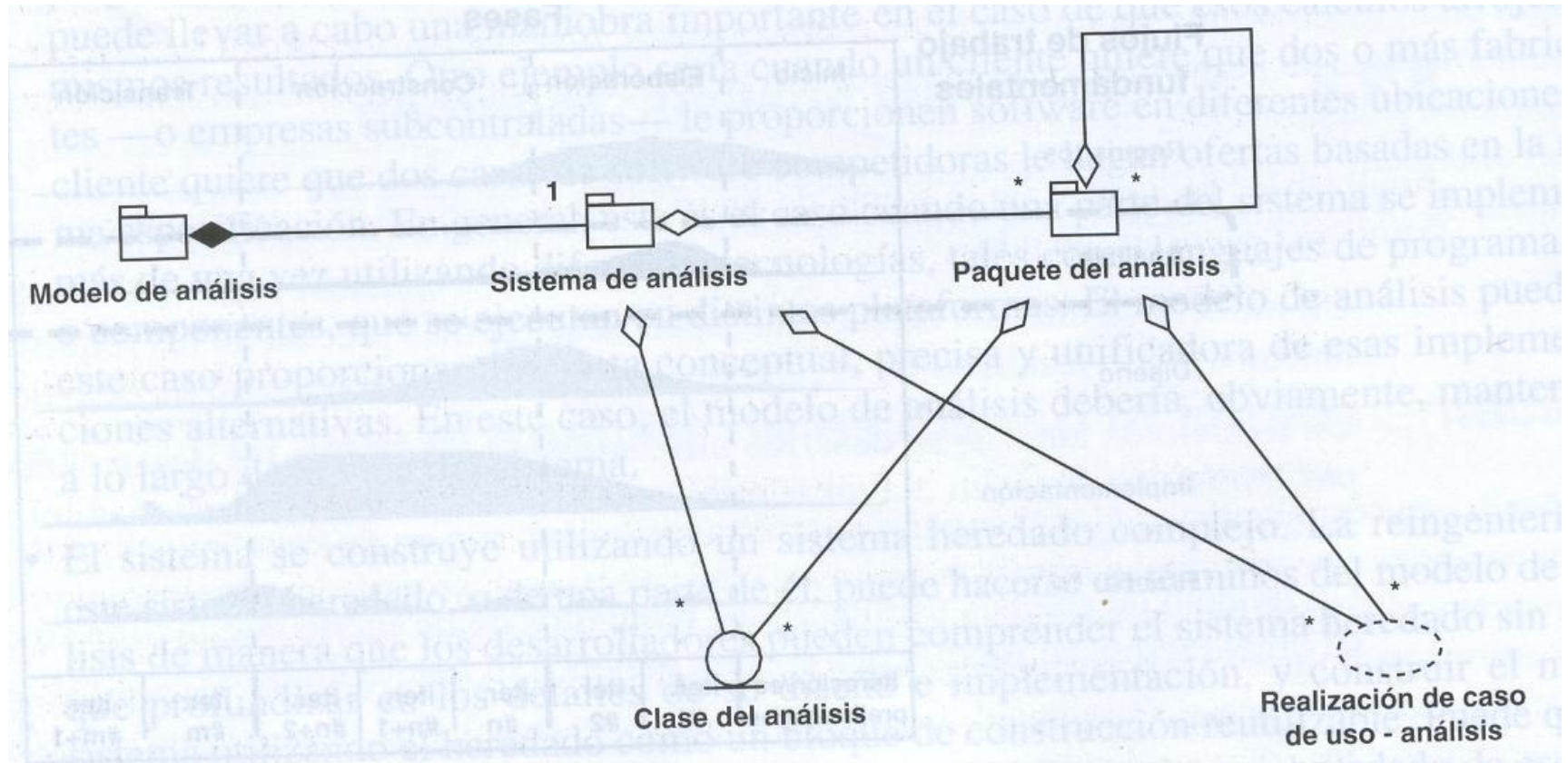
Modelo de dominio



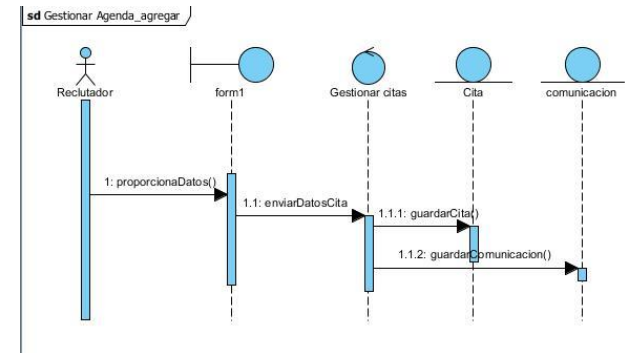
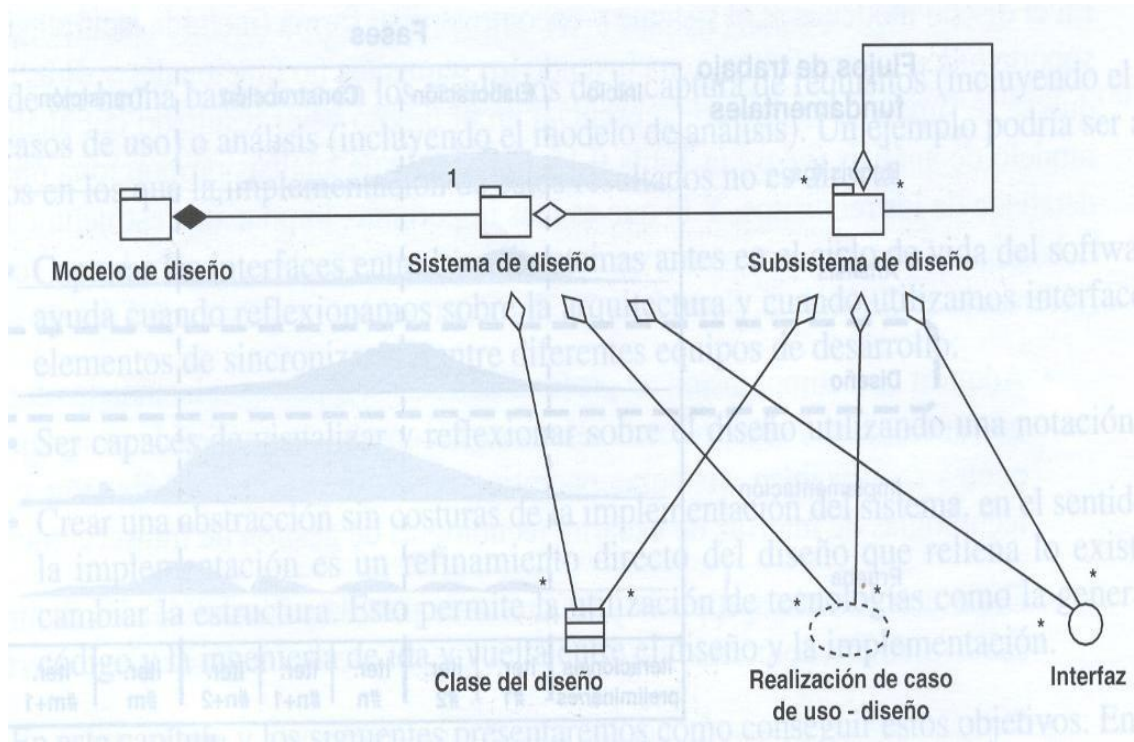
Modelo de casos de uso



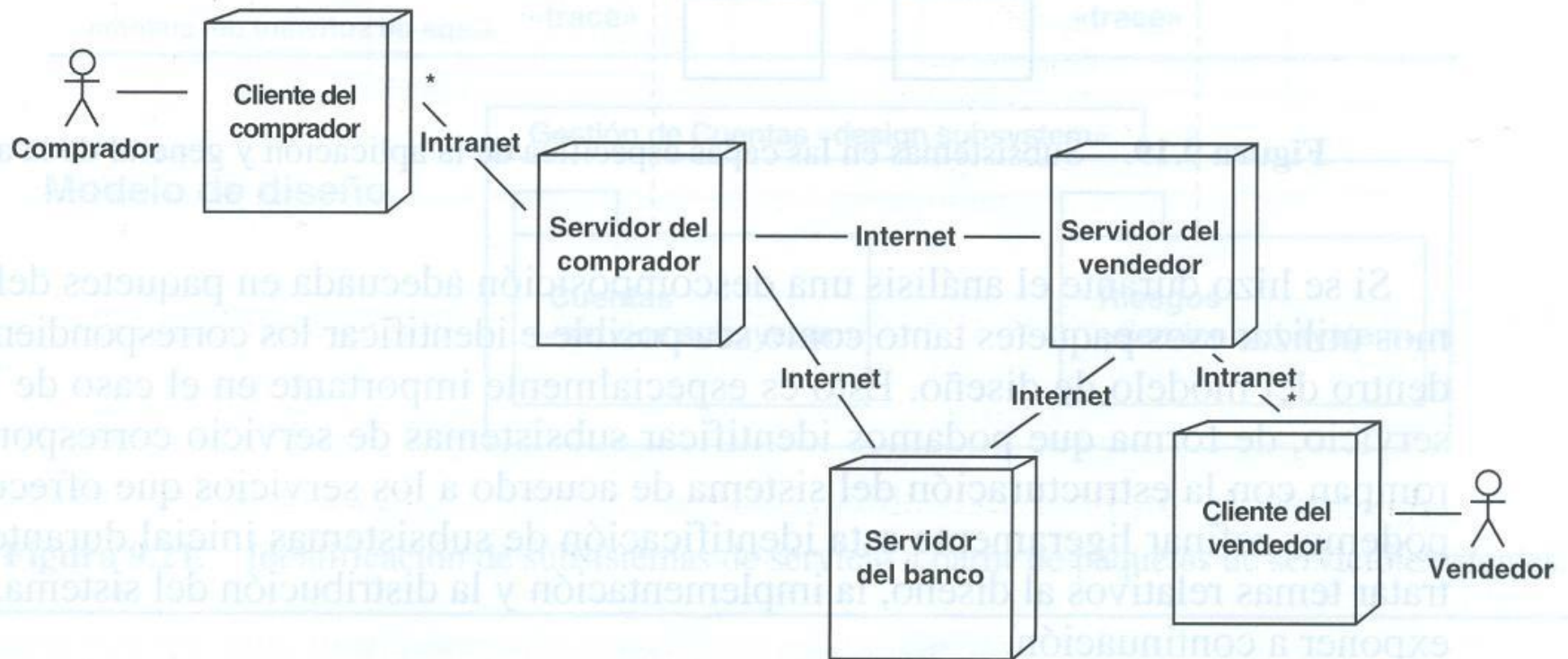
Modelo de análisis



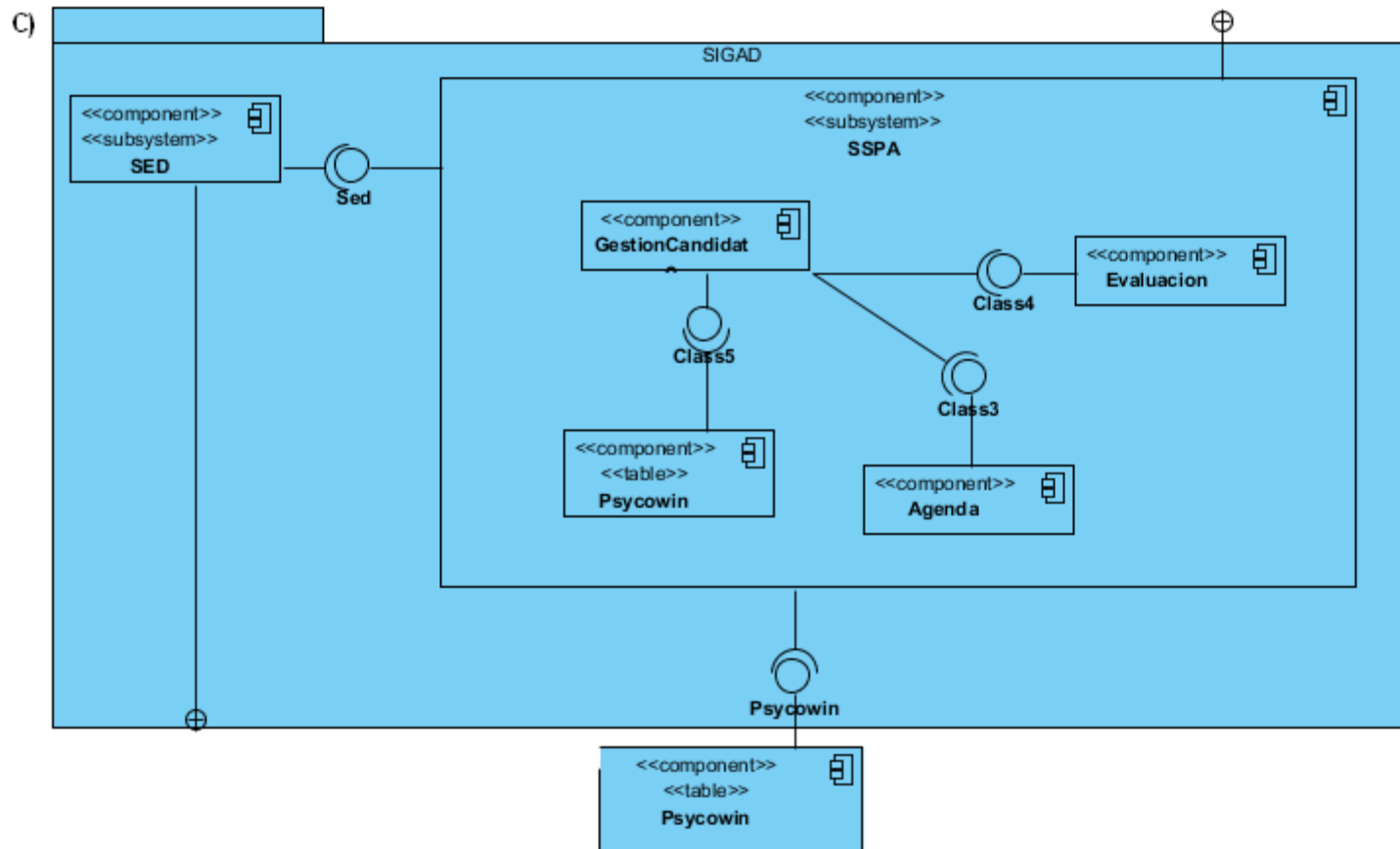
Modelo de diseño



Modelo de despliegue



Modelo de implementación



La disciplina del Análisis

- El flujo de trabajo o disciplina del análisis del PU tiene dos propósitos:
 - Obtener una comprensión más profunda de los requerimientos.
 - Describir dichos requerimientos de tal forma que el diseño resultante y la implementación sean fáciles de mantener.

Análisis Clásico vs. Análisis OO

Análisis	
Tradicional o estructurado	Orientado a objetos
El análisis está orientado a los Procesos del sistema.	El análisis está orientado a los Objetos.
Perspectiva de análisis Entrada-Proceso-Salida	Modela los objetos que son parte del Sistema.
Utiliza diagramas estructurados como representación gráfica del sistema.	Utiliza diagramas orientados a objetos como representación gráfica del sistema.
El diseño inicia una vez que ha culminado la fase de análisis de sistema.	El diseño inicia aún antes de concluir con la etapa de análisis.
Las herramientas utilizadas son: Diagrama de Flujo de Datos (DFD), Diagramas de Entidad-Relación y Diccionario de Datos.	Las herramientas utilizadas son: Diagramas de Clases, Diagramas de Objetos, Diagramas de Actividades y Diagramas de Interacción.



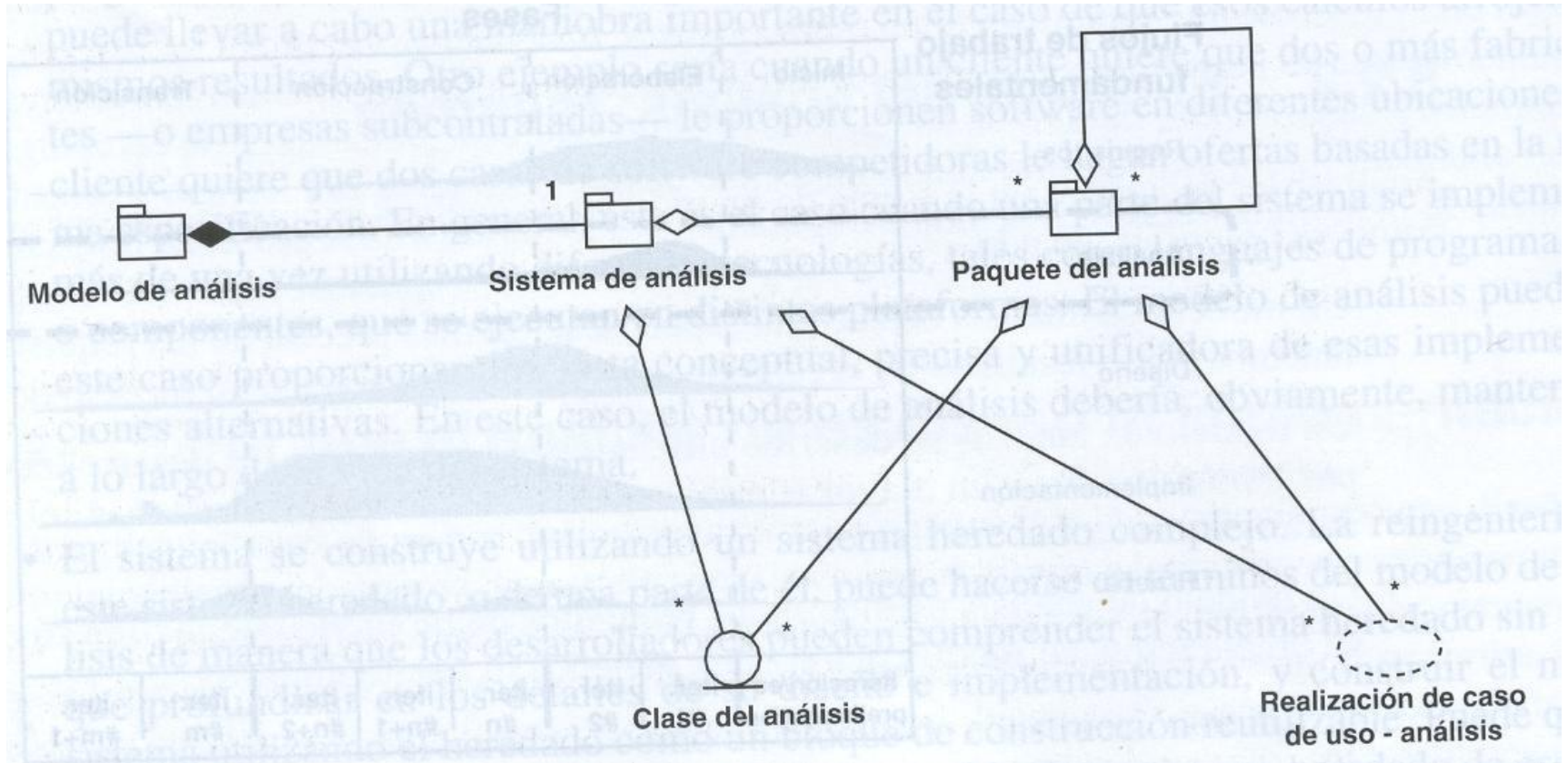
Dirigido por casos de uso

- El PU es controlado por los casos de uso.
- Durante el análisis, los casos de uso se describen en términos de **clases** del producto software.
- El producto de trabajo (artefacto) más importante del análisis es el **modelo de análisis**.

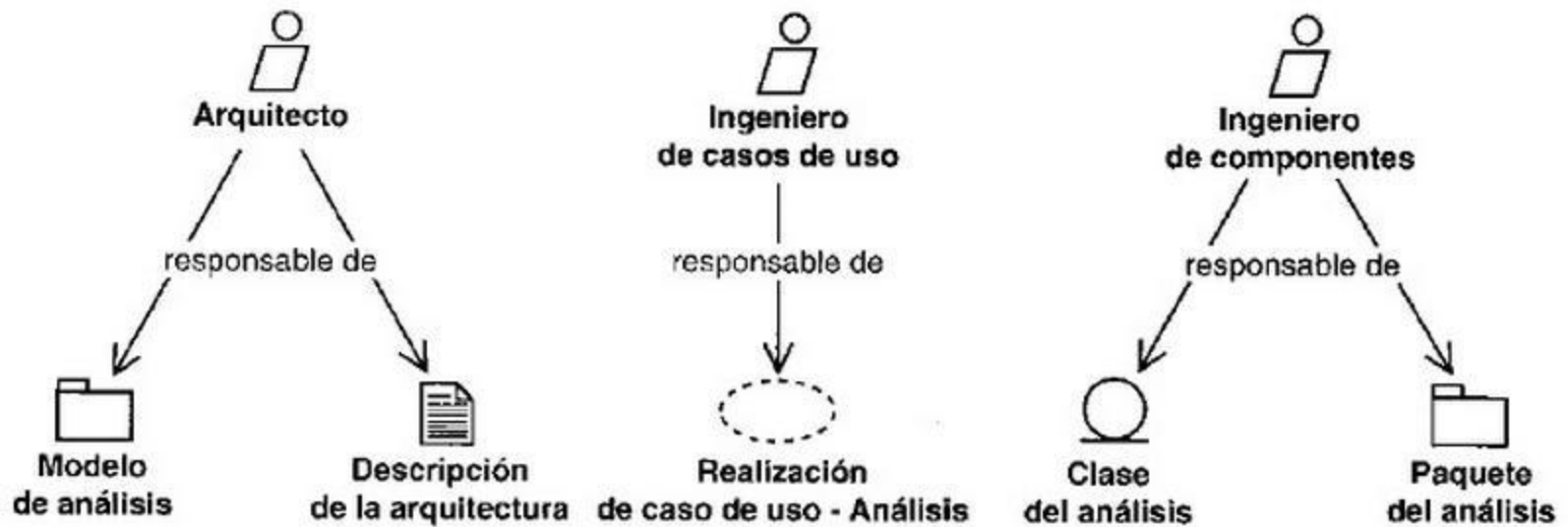
Modelo de análisis

Modelo de casos de uso	Modelo de análisis
Descrito con el lenguaje del cliente.	Descrito con el lenguaje del desarrollador.
Vista externa del sistema.	Vista interna del sistema.
Estructurado por los casos de uso; proporciona la estructura a la vista externa.	Estructurado por clases y paquetes estereotipados; proporciona la estructura a la vista interna.
Utilizado fundamentalmente como contrato entre el cliente y los desarrolladores sobre qué debería y qué no debería hacer el sistema.	Utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, cómo debería ser diseñado e implementado.
Puede contener redundancias, inconsistencias, etc., entre requisitos.	No debería contener redundancias, inconsistencias, etc., entre requisitos.
Captura la funcionalidad del sistema, incluida la funcionalidad significativa para la arquitectura.	Esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida la funcionalidad significativa para la arquitectura; sirve como una primera aproximación al diseño.
Define casos de uso que se analizarán con más profundidad en el modelo de análisis.	Define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso.

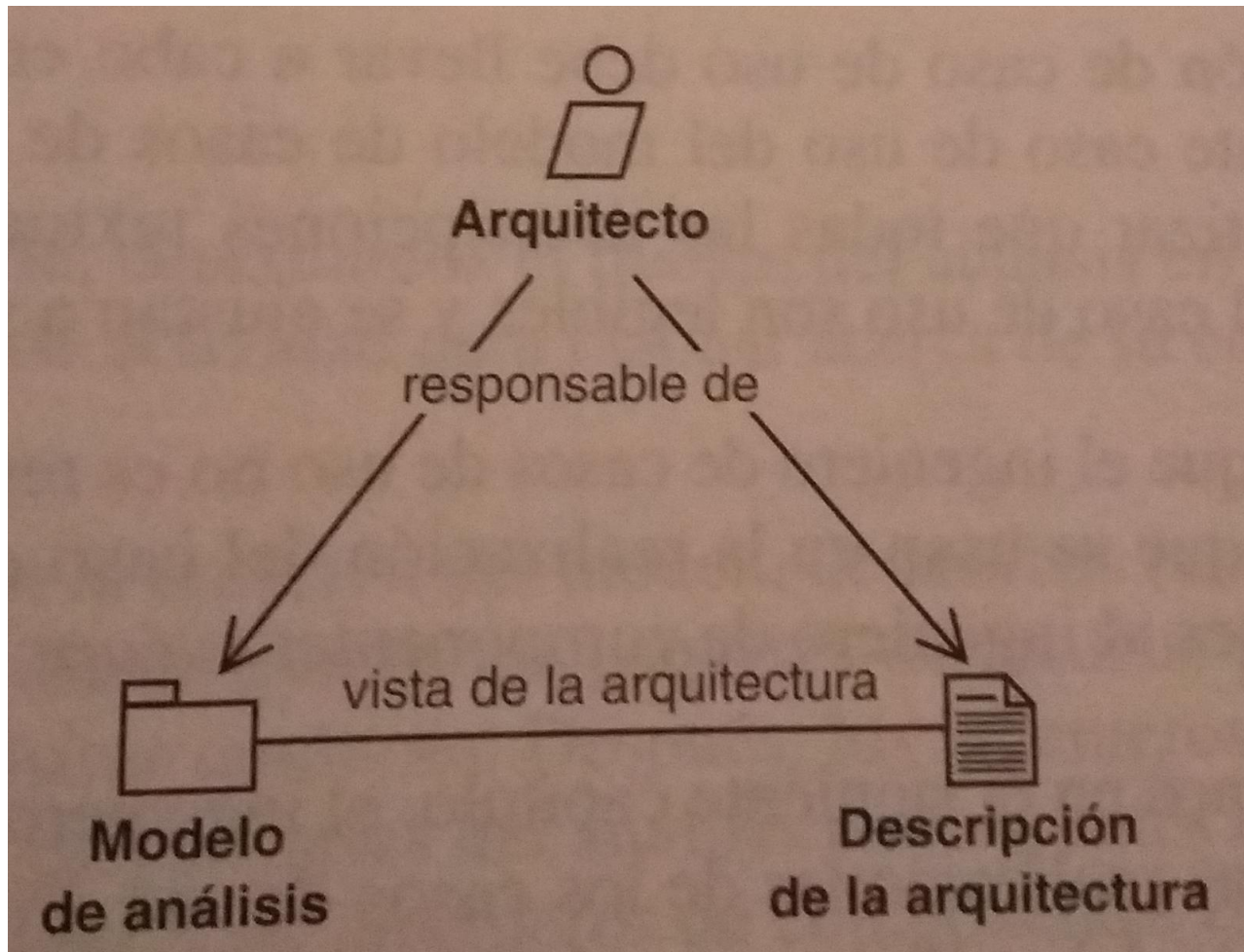
Modelo de análisis



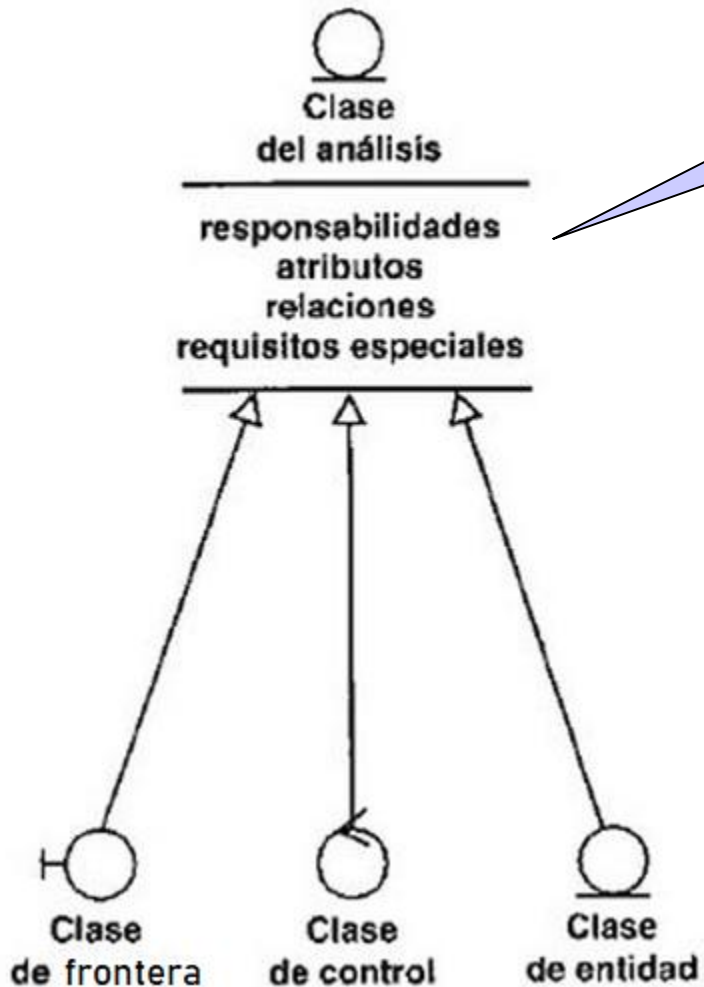
Roles en el Análisis OO



Modelos del análisis



Clases del análisis



Nótese la ausencia de operaciones

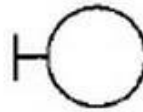
Clases del análisis

- El PU tiene tres tipos estereotipados de clases: de entidad, de frontera (borde) y de control.

Alternativa 1:



Cuenta

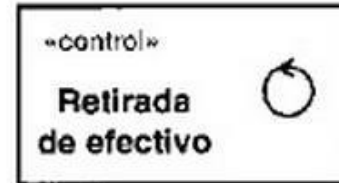
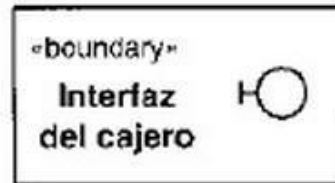
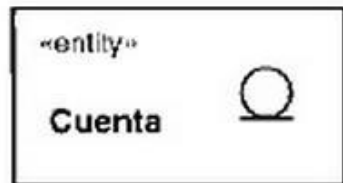


Interfaz
del cajero



Retirada
de efectivo

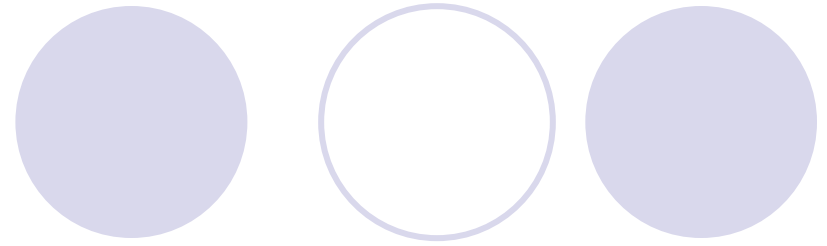
Alternativa 2:



Clase de frontera (borde)

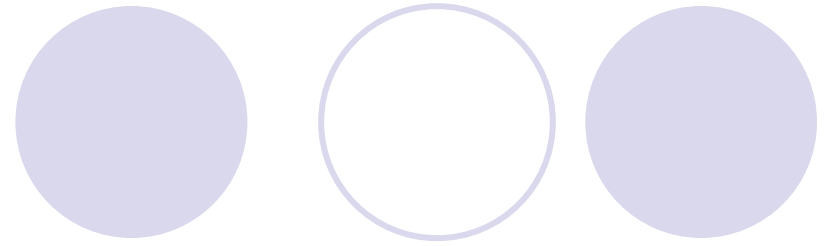
- Se utilizan para modelar la interacción entre el sistema y sus actores (usuarios y sistemas externos). Esta interacción implica recibir y presentar información y peticiones de y hacia los usuarios.
- Representan abstracciones de ventanas, formularios, paneles, interfaces de comunicaciones, interfaces de impresoras, sensores, terminales, APIs, informes impresos, y reportes.

Clase de control



- Representan coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto.
- Se utilizan para representar derivaciones, algoritmos y cálculos complejos, como la lógica del negocio, que no pueden asociarse con ninguna información concreta o de larga duración.

Clase de entidad



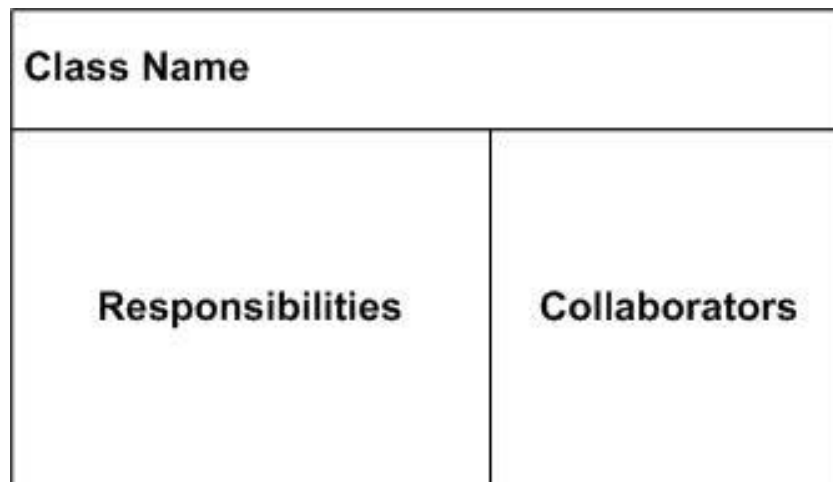
- Se utilizan para modelar información que posee una vida larga y que es a menudo persistente.
- Modelan la información y el comportamiento asociado a algún fenómeno o concepto, como una persona, un objeto del mundo real o un suceso.

Extracción de las clases de entidad

- Consiste en tres pasos que se llevan de manera iterativa e incremental:
 - **Modelado funcional:** presenta los escenarios de todos los casos de uso (un escenario es una instancia de un caso de uso)
 - **Modelado de clases de entidad:** determina las clases de entidad y sus atributos. Después, determina las interrelaciones e interacciones entre las clases de entidad. Presenta esta información en forma de un diagrama de clases.
 - **Modelado dinámico:** determina las operaciones realizadas por o a cada clase de entidad o subclase. Presenta esta información en forma de un diagrama de estados y actualiza el diagrama de clases.

Modelado de clases de entidad

- Extracción de sustantivos
 - Describir el software en uno o varios párrafos
 - Identificar los sustantivos
 - Excluir los sustantivos que se ubican afuera de la frontera del problema
 - Considerar los sustantivos abstractos como atributos de clase
- Tarjetas CRC (Class-Responsibility-Collaboration)



Tarjetas CRC

Simulación de escenario para un cajero electrónico

- GestorDeCuenta
- VerificadorDeNIP
- GestorDeReintegros
- GestorDeIngresos
- CajaElectronica
- SelectorDeActividad
- LectorDeTarjetas

Tarjetas CRC

GestorDeCuenta

- . Verificar validez de cuenta
- . Devolver NIP
- . Verificar información de reintegros/ingresos

Tarjetas CRC

VerificadorDeNIP	
. Recibir NIP de GestorDeCuenta;	GestorDeCuenta
. Devolver falso si no hay cuenta	
. Presentar ventana de solicitud de NIP	
. Recibir NIP del usuario	

Tarjetas CRC

GestorDeReintegros	
. Preguntar al cliente la cantidad a retirar	GestorDeCuenta
. Verificar cantidad con GestorDeCuenta	CajaElectronica
. Ordenar a CajaElectronica que de efectivo	

Tarjetas CRC

GestorDeIngresos	
<ul style="list-style-type: none">· Pedir a CajaElectronica que entregue y recupere sobre de depósito	CajaElectronica

Tarjetas CRC

CajaElectronica	
. entregar efectivo	GestorDeIngresos
. entregar sobre de ingreso con indicación de hora	GestorDeReintegros
. Recuperar sobre de ingreso	

Tarjetas CRC

SelectorDeActividad	
. mostrar menú de operaciones	GestorDeIngresos
. leer la selección del usuario	GestorDeReintegros
. llamar al gestor de operaciones correspondiente	

Tarjetas CRC

LectorDeTarjetas	
<ul style="list-style-type: none">. Mostrar mensaje de bienvenida, esperar tarjeta. Pedir a VerificadorDeNip que compruebe validez. Llamar al SelectorDeActividad. Devolver tarjeta al usuario	<p>VerificadorDeNip</p> <p>SelectorDeActividad</p>

La disciplina del Diseño

- Los propósitos de esta disciplina son:
 - Adquirir una comprensión en profundidad de los aspectos relacionados con los requerimientos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, etc.
 - Crear una entrada apropiada y un punto de partida para actividades de implementación.

Modelo de diseño

Modelo de análisis

Modelo conceptual, porque es una abstracción del sistema y permite aspectos de la implementación.

Genérico respecto al diseño (aplicable a varios diseños).

Tres estereotipos conceptuales sobre las clases: Control, Entidad e Interfaz.

Menos formal.

Menos caro de desarrollar (ratio al diseño 1:5).

Menos capas.

Dinámico (no muy centrado en la secuencia).

Bosquejo del diseño del sistema, incluyendo su arquitectura.

Creado principalmente como “trabajo de a pie” en talleres o similares.

Puede no estar mantenido durante todo el ciclo de vida del software.

Define una estructura que es una entrada esencial para modelar el sistema —incluyendo la creación del modelo de diseño.

Modelo de diseño

Modelo físico, porque es un plano de la implementación.

No genérico, específico para una implementación.

Cualquier número de estereotipos (físicos) sobre las clases, dependiendo del lenguaje de implementación.

Más formal.

Más caro de desarrollar (ratio al análisis 5:1).

Más capas.

Dinámico (muy centrado en las secuencias).

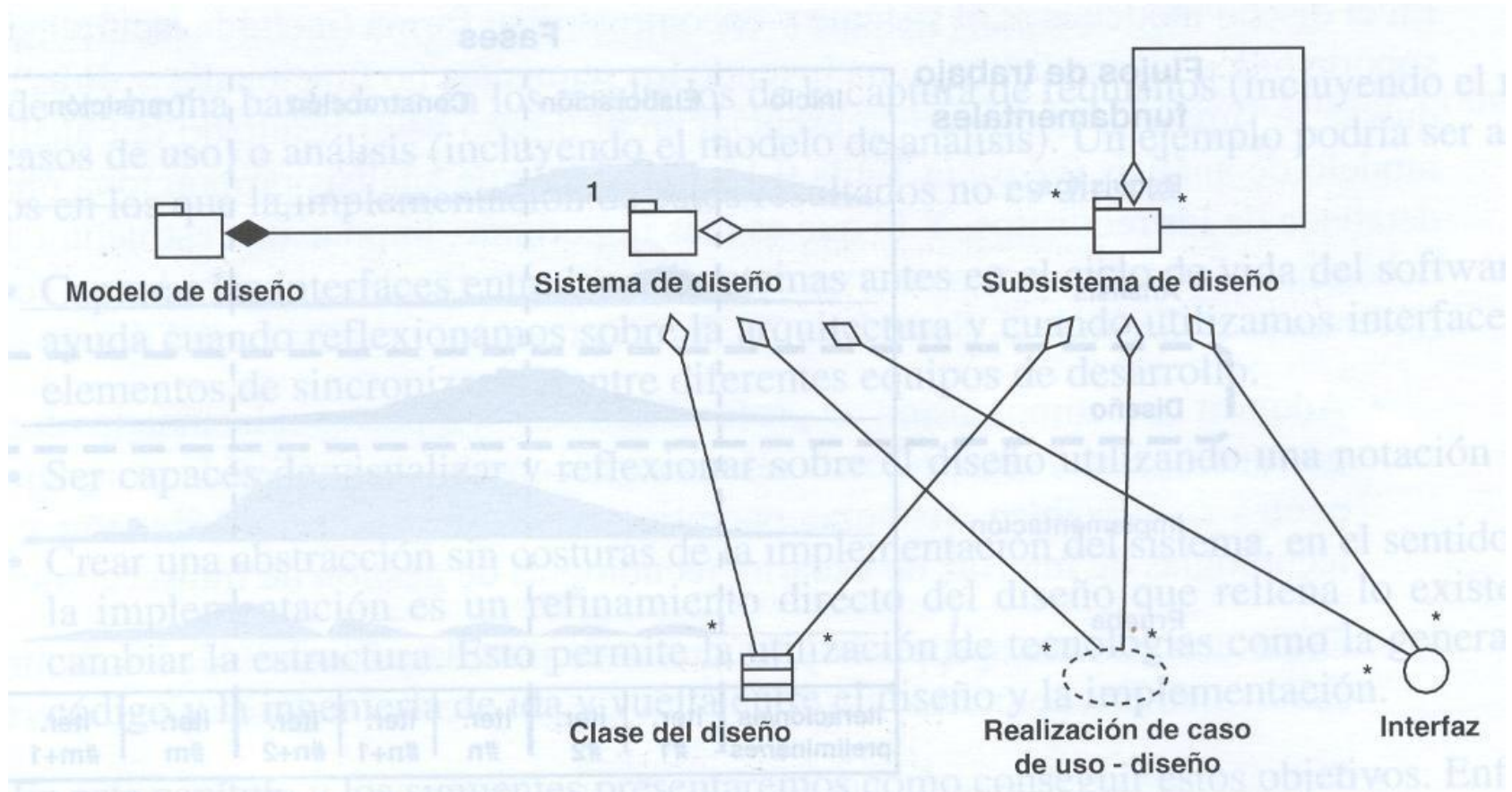
Manifiesto del diseño del sistema, incluyendo su arquitectura (una de sus vistas).

Creado principalmente como “programación visual” en ingeniería de ida y vuelta; el modelo de diseño es realizado según la ingeniería de ida y vuelta con el modelo de implementación (descrito en el Capítulo 10).

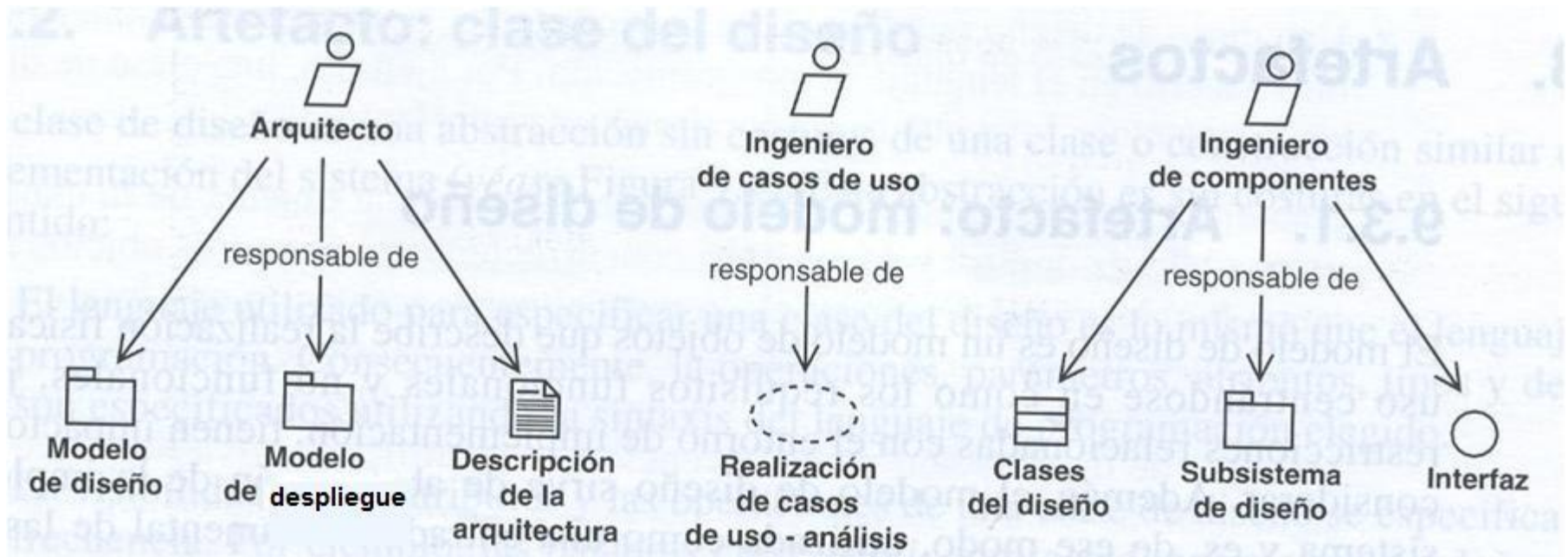
Debe ser mantenido durante todo el ciclo de vida del software.

Da forma al sistema mientras que intenta preservar la estructura definida por el modelo de análisis lo más posible.

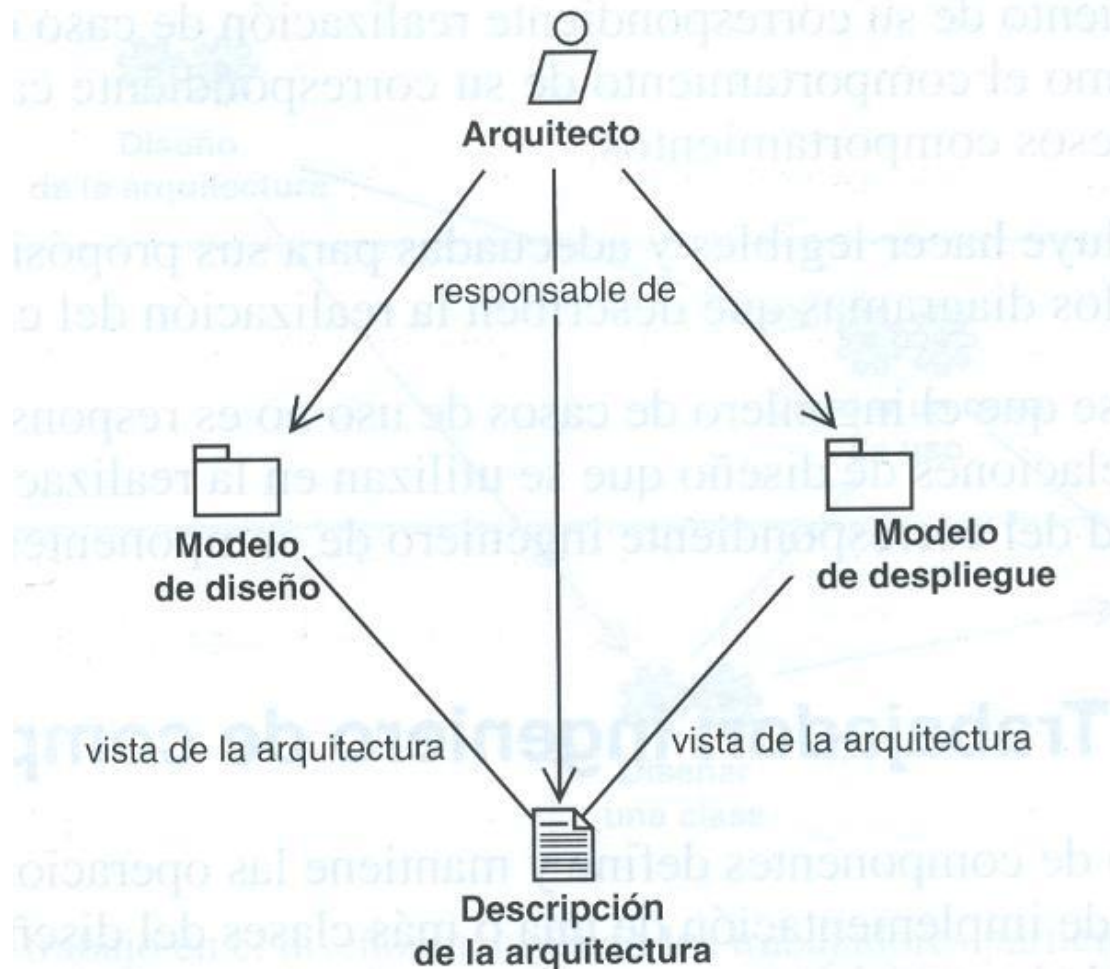
Modelo de diseño



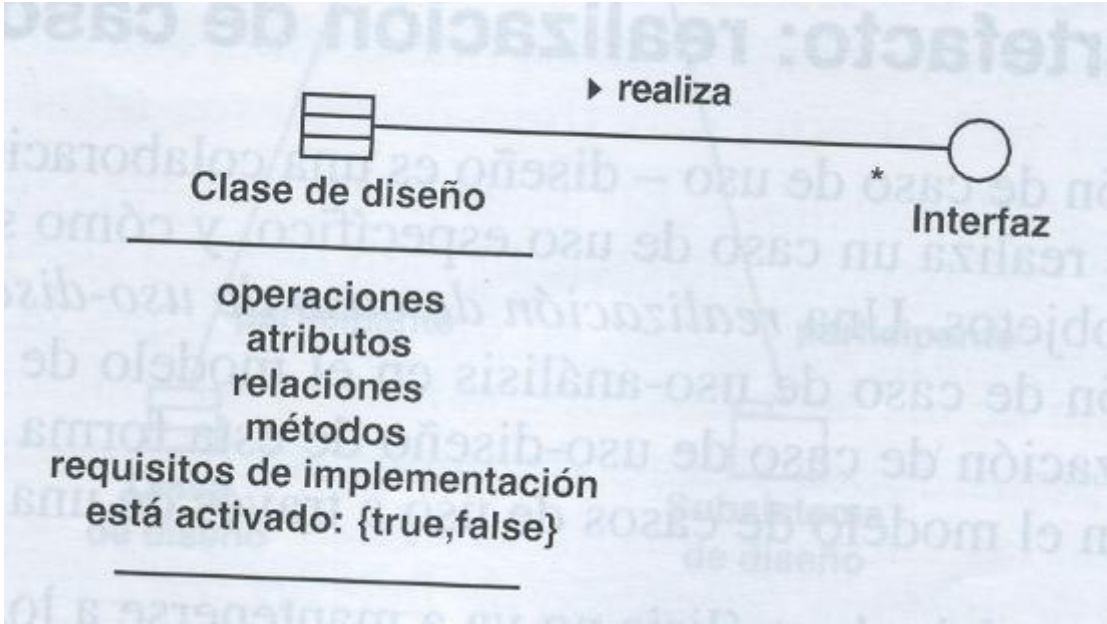
Roles en el Diseño OO



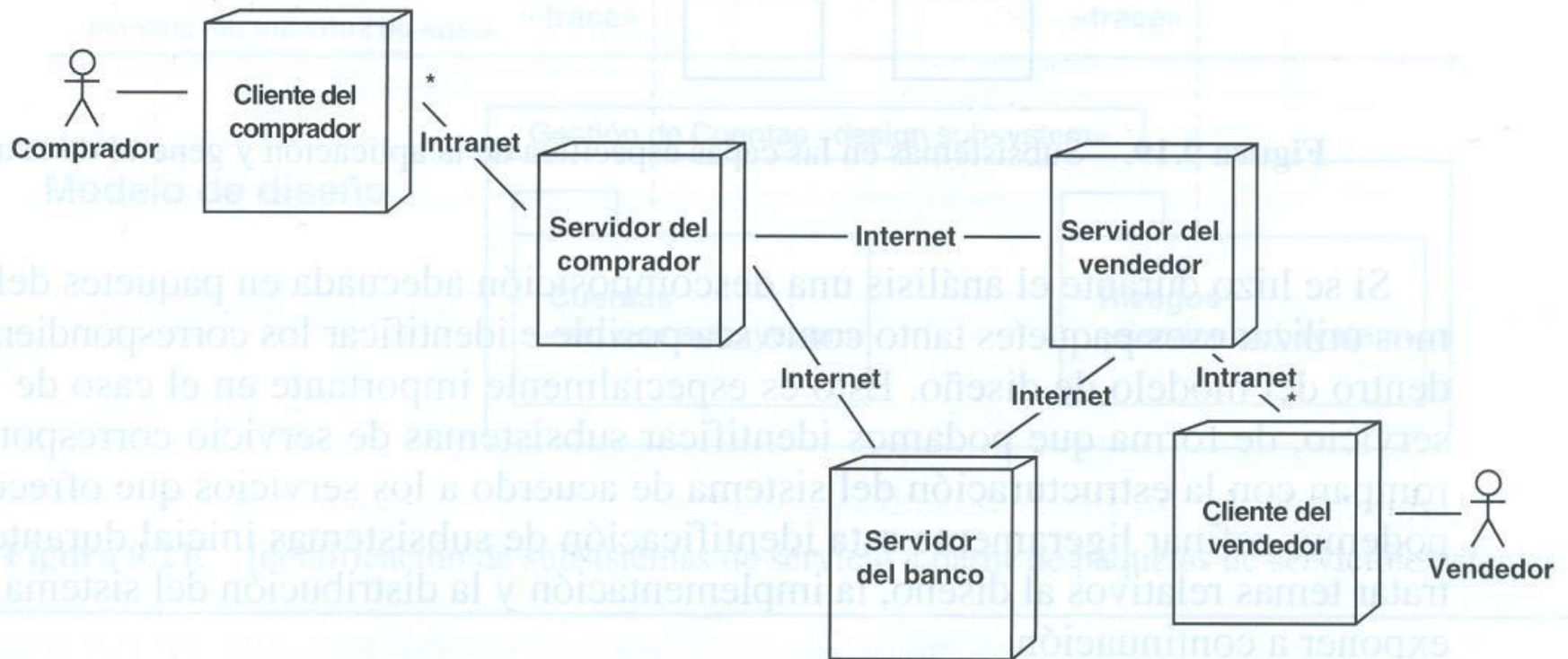
Modelos del diseño



Clases del diseño



Modelo de despliegue



RUP: Productos de trabajo del análisis y diseño

- Arquitectura de prueba de concepto
- Arquitectura de referencia
- *Data Migration Specification*
- Documento de arquitectura de software
- Mapa de navegación
- **Modelo de análisis**
- **Modelo de datos**
- **Modelo de despliegue**
- **Modelo de diseño**
- Modelo de servicio
- Modelo de servicio de objetivos
- **Prototipo de interfaz de usuario**

Fin

