
Capitulo 2

Análisis y Diseño del Sistema

Conceptos Fundamentales de Diseño

- Diseñar es el esfuerzo para definir la arquitectura, componentes, interfaces y otras características de un sistema o componente (IEEE 610-1990).
- El diseño de software es la actividad del ciclo de vida del software en la cual se analizan los requisitos para producir una descripción de la estructura interna del software que sirva de base para su construcción.
- La salida es un conjunto de modelos y artefactos que registran las principales decisiones adoptadas.

Conceptos Fundamentales de Diseño

- El diseño es una etapa fundamental y, en muchas ocasiones, la más importante en el desarrollo de Software.
- Es el momento en que **los profesionales tienen que aportar sus conocimientos, experiencia y creatividad para llegar a una solución** que cumpla con los requerimientos funcionales y no funcionales establecidos en la fase de la toma de requisitos.
- Un error de diseño puede acarrear problemas en todo el proyecto y provocar que este caiga en una espiral de continuos cambios y rehacer constantemente el trabajo

Definición de Diseño

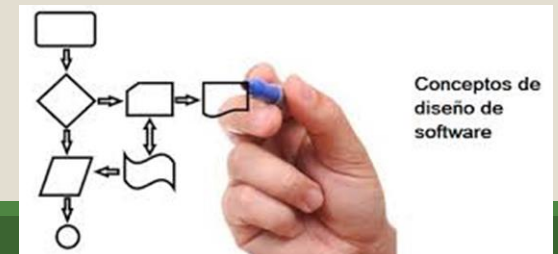
El estándar ISO 12207 establece que el diseño de software consiste de dos actividades, entre el análisis de requisitos y la construcción de software:

- **Diseño Arquitectural** (Alto nivel)

Describe la estructura y organización de alto nivel es decir, los subsistemas o componentes y sus relaciones

- **Diseño Detallado:**

Describe cada componente y su comportamiento específico, de forma que pueda procederse a su construcción.



Importancia del Diseño

- La importancia del Diseño del Software se puede definir en una sola palabra **Calidad**.
- Dentro del diseño es donde se fomenta la calidad del Proyecto.
- El Diseño es la única manera de materializar con precisión los requerimientos del cliente.

Requerimientos del Software

Los requerimientos No Funcionales están muy relacionados a la calidad del Software.

Los atributos de calidad son propiedades por las que el cliente juzga la calidad del software.

Los atributos de calidad tienen una gran influencia en la arquitectura del software

- Deben caracterizarse de una forma específica en el sistema
- Una forma poderosa de caracterizar atributos de calidad y representar la perspectiva de los involucrados **son los escenarios** .

De acuerdo al SEI, algunos ejemplos de atributos por la cual los stakeholders juzgan la calidad del software son:

➤ **Desempeño**

➤ **Seguridad**

➤ **modificabilidad**

➤ **Confiabilidad**

➤ **usabilidad**

➤ **Disponibilidad**

➤ **Calibrabilidad**

➤ **webifiabilidad**

➤ **Rendimiento**

➤ **configurabilidad**

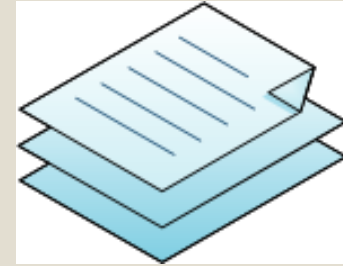
➤ **subsetabilidad**

➤ **Reusabilidad**

Quality Attributes



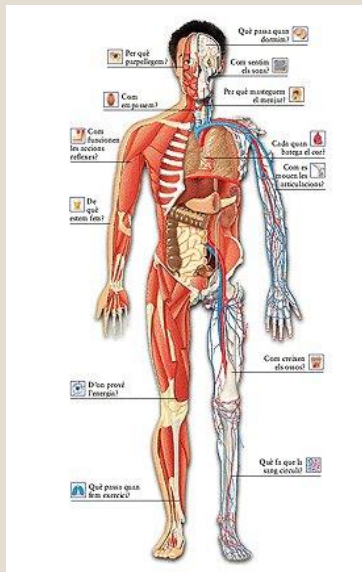
Stakeholder Concerns



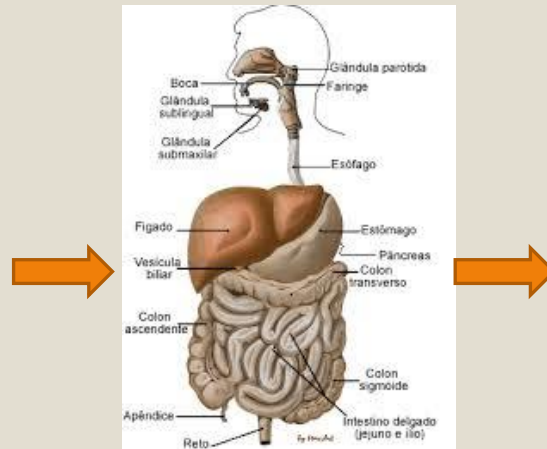
Quality Attribute Requirements

"Increase market share"	----->	Modifiability, Usability
"Maintain a quality reputation"	----->	Performance, Usability, Availability
"Introduce new capabilities seamlessly"	----->	Performance, Availability, Modifiability
"Provide a programmer-friendly framework"	----->	Modifiability
"Integrate with other systems easily"	----->	Interoperability, Portability, Modifiability

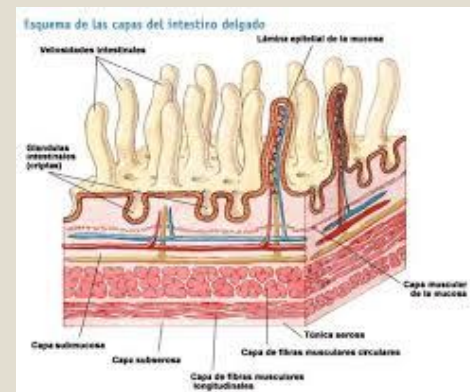
Estructuras y Vistas



El cuerpo humano es un conjunto de múltiples estructuras

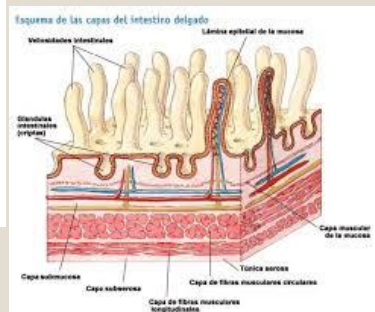
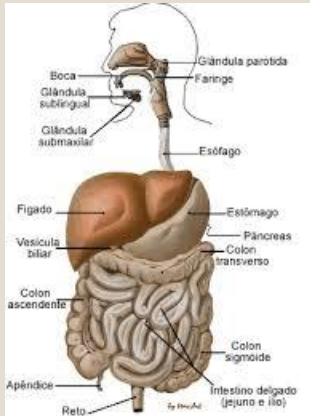


Una vista estática de una de las estructuras humanas



Una vista dinámica
de esa estructura

Estructuras, Vistas y Stakeholders



Estas vistas son necesarias para un gastroenterólogo

....pero no le sirven a un oftalmólogo

- Diferentes stakeholders están interesados en diferentes estructuras.
- Las vistas deben representar las estructuras en las cuales los stakeholders están interesados.

Con el software pasa algo similar.....

Estructuras y Vistas

Estructura

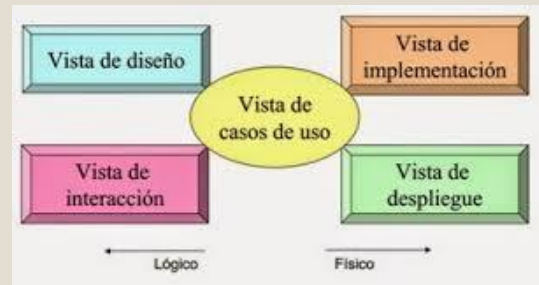
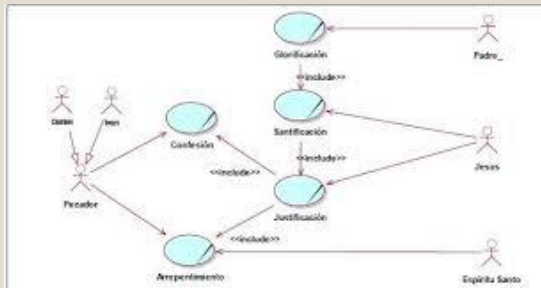
- Es el conjunto actual de elementos de arquitectura tal como existen en hardware o software

Vista

- Es una representación de un conjunto coherente de elementos de la arquitectura, los cuales son escritos y leídos por los stakeholders del sistema.

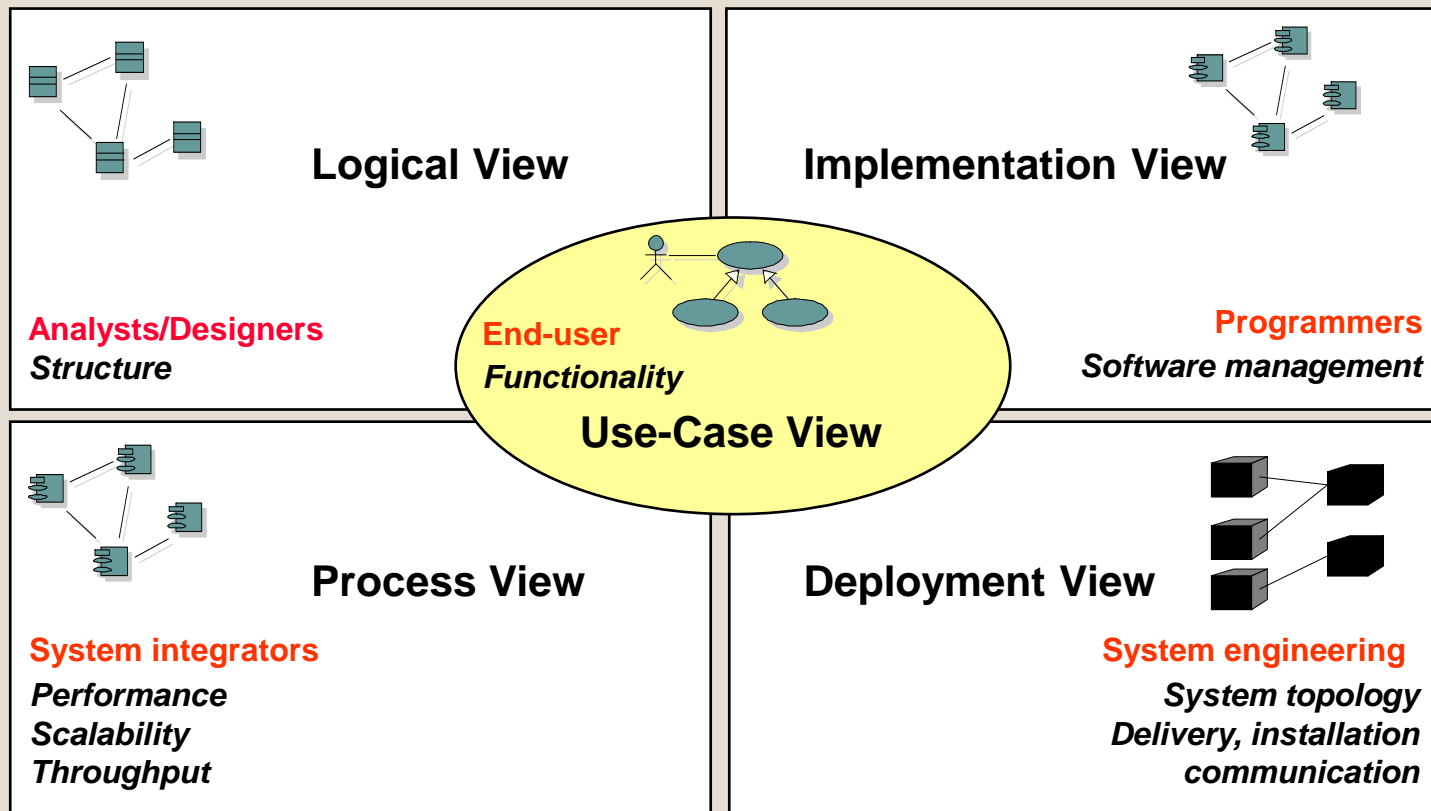
Una Vista (Viewpoint)

- Representa un conjunto de elementos, patrones, plantillas y convenciones para construir un tipo de vista.

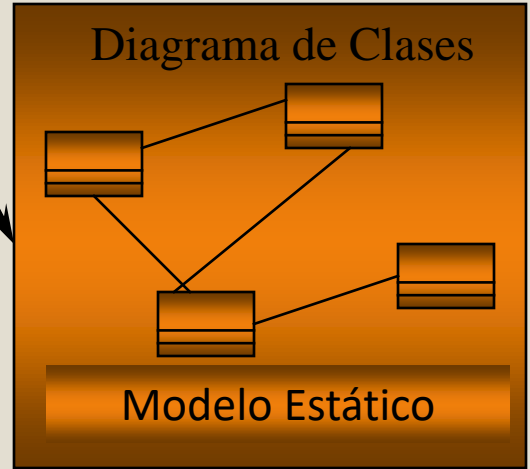
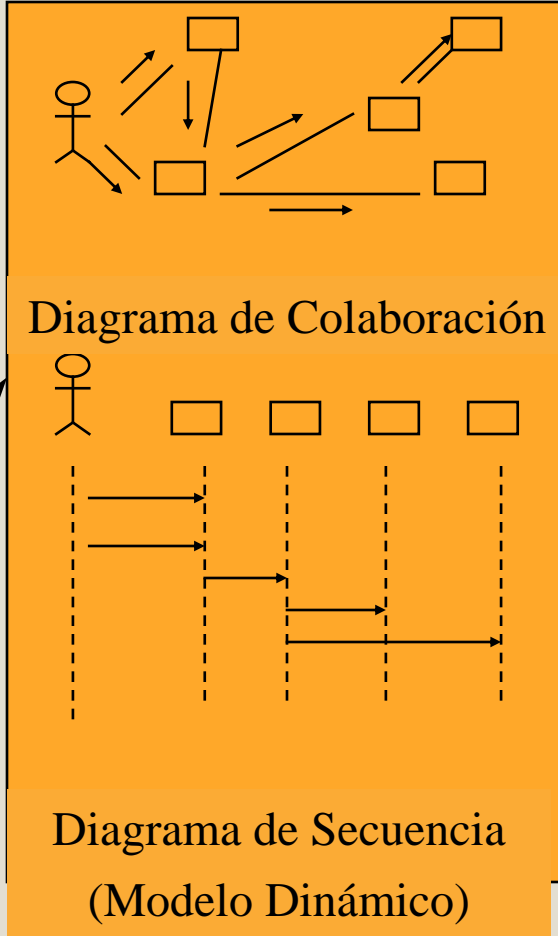
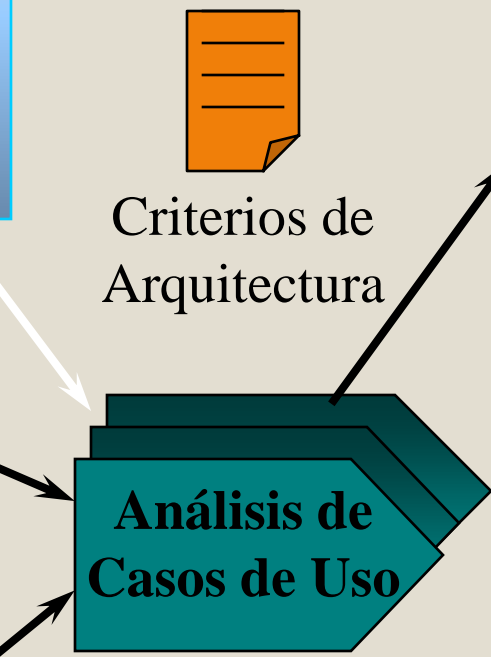
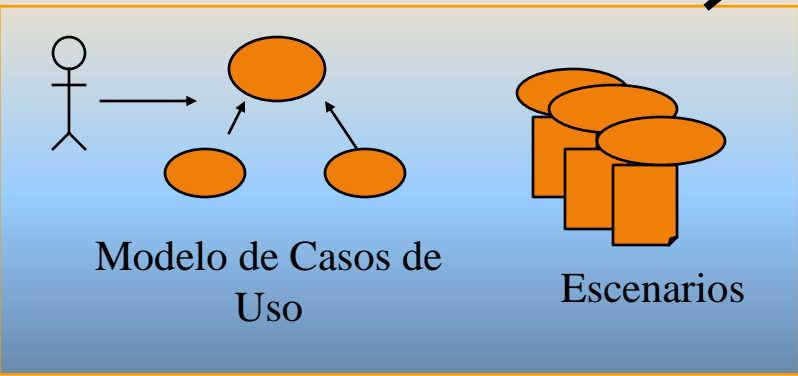
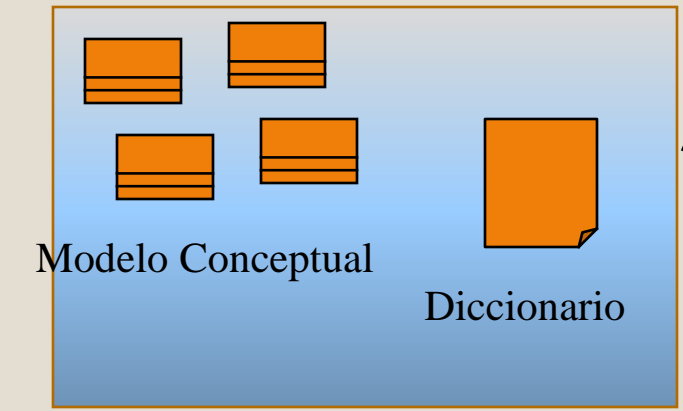
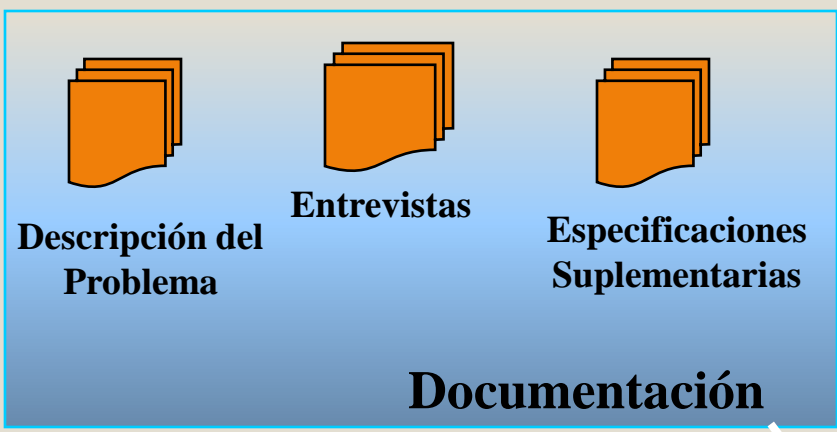


Modelo de Vista 4+1

Framework para Descripción de Arquitectura, basado en vistas lógicas y físicas UML y una vista funcional de casos de uso.

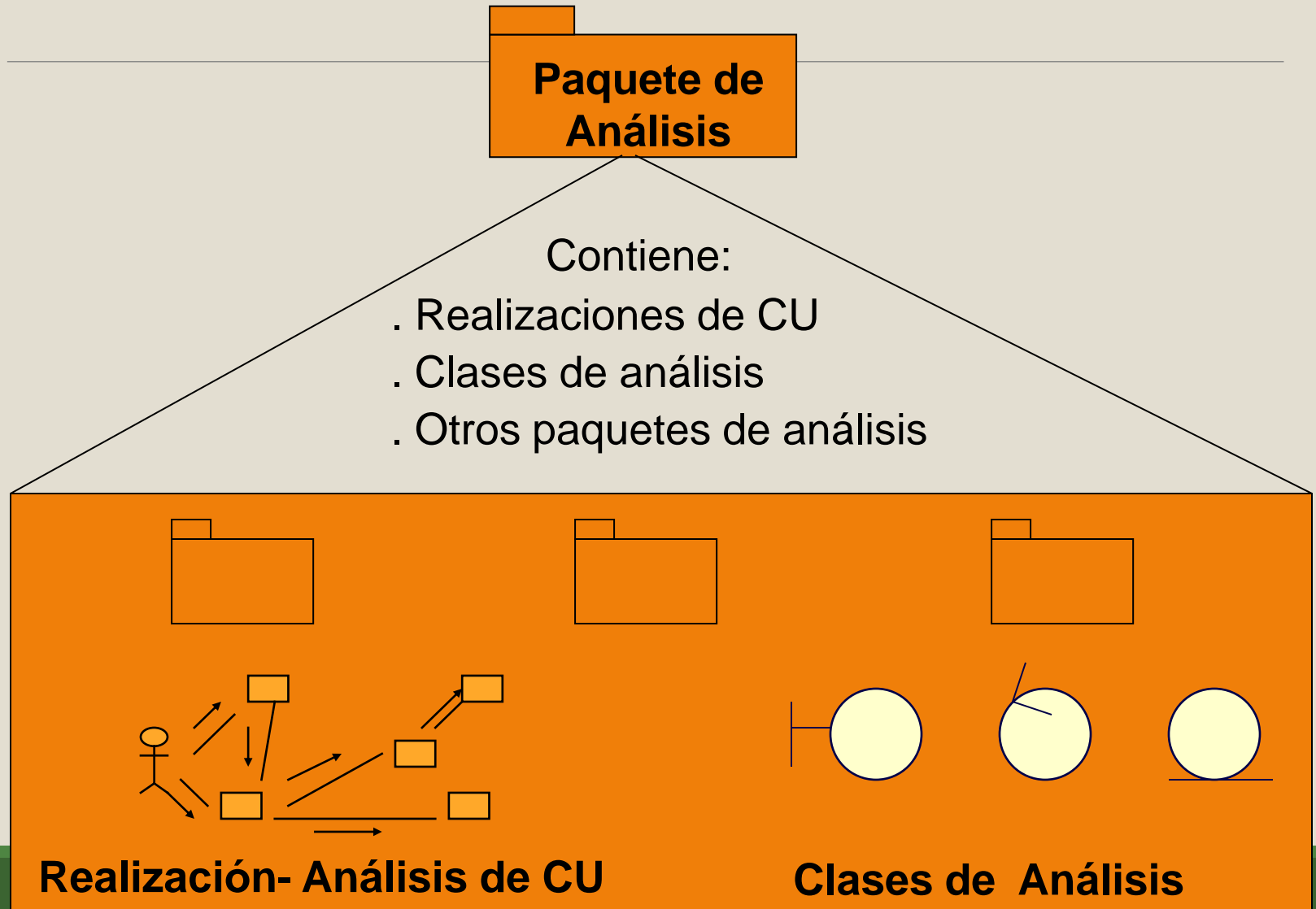


Construcción del Modelo de Análisis



Construyendo el Modelo de Análisis

El Modelo de Análisis



El Modelo de Análisis

Puede agruparse en 2 partes o vistas

- El “Modelo Estático” que representa la estructura del modelo de análisis e incluye:
 - Diagrama de Paquetes
 - Diagrama de Clases
- El “Modelo Dinámico” que muestra las interacciones y responsabilidades que se manejan
 - Diagramas de Interacción
 - Diagrama de Secuencia
 - Diagrama de Colaboración

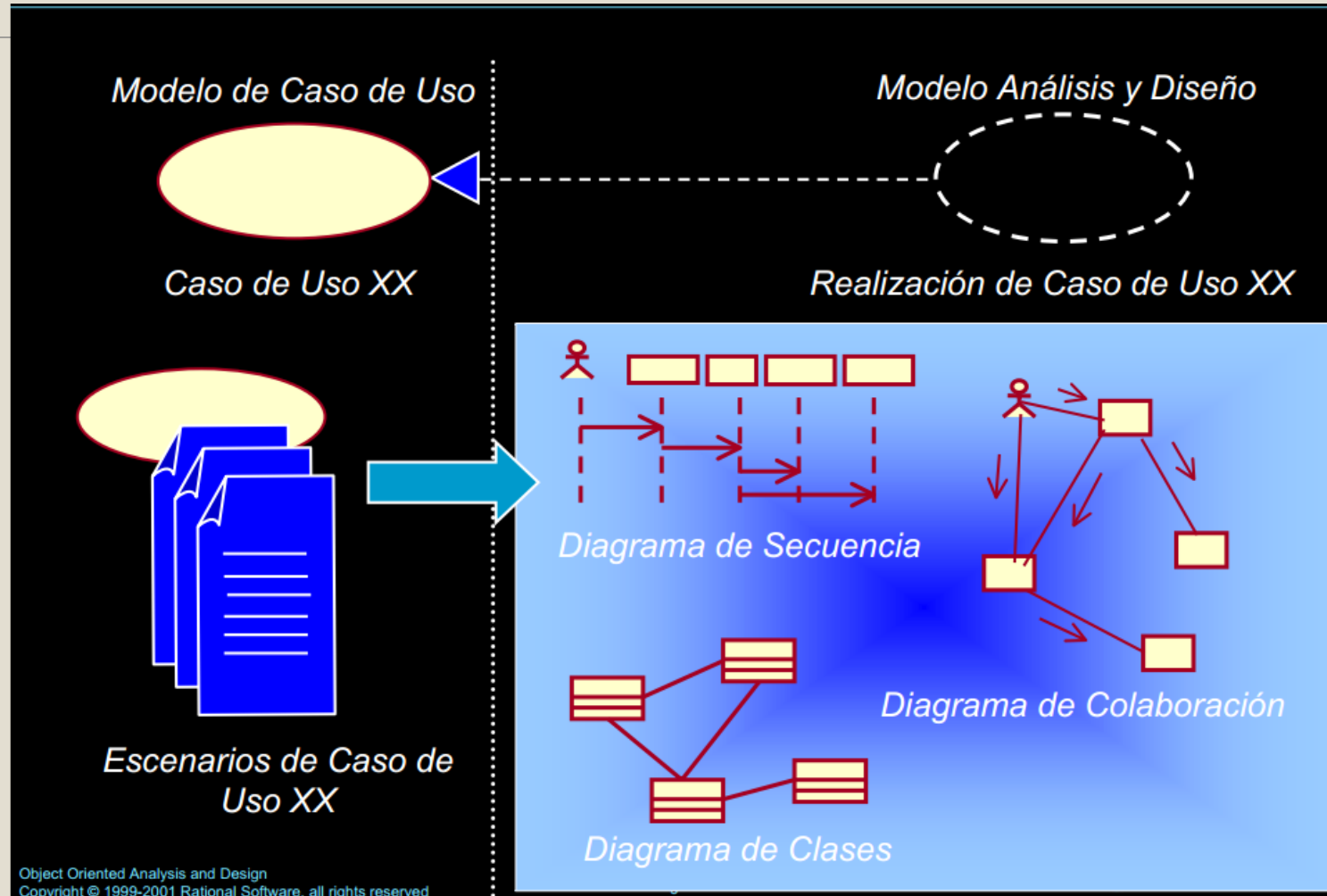
Clases de Análisis un primer paso a ejecutables



Realización de Casos de Uso

- Una Realización de Casos de Uso (RCU) describe cómo un escenario de un CU es realizado por varios objetos colaborando entre sí.
- Esto se representa con diagramas de secuencia, colaboración y de clases.
- La definición de una RCU se inicia con el Análisis de Casos de Uso (para el Modelo de Análisis) y se completa con el Diseño de Casos de Uso (para el Modelo de Diseño).
- El objetivo final de una RCU es especificar qué clases deben ser construidas para implementar ese CU.
- En UML, una RCU se muestra como un óvalo con límite punteado, que está asociado al caso de uso que realiza, con una flecha de línea punteada y cabeza cerrada.

Qué es una realización de Casos de Uso



Revisión del Modelo de Análisis

Una vez el modelo de análisis llega a ser estable es revisado primero por los desarrolladores con el cliente.

El objetivo de la revisión es asegurarse de que la especificación del sistema es correcta, completa, consistente y realista.

Diagrama de Clases de Análisis (DCA)

¿Qué son Clases?

- ☀ Una Clase representa un conjunto de Objetos que comparten:
 - ✱ Las mismas Propiedades (atributos)
 - ✱ El mismo comportamiento (operaciones)
 - ✱ Las mismas relaciones con otros objetos (asociaciones)
 - ✱ La misma semántica (descripción breve)
- ☀ Por lo tanto, una clase es una pieza de software que actúa como un molde para fabricar tipos particulares de objetos que disponen de los mismos atributos y métodos.

¿Qué son Clases de Análisis?

Una clase de análisis representa la abstracción de una o varias clases y/o subsistemas del diseño del sistema.

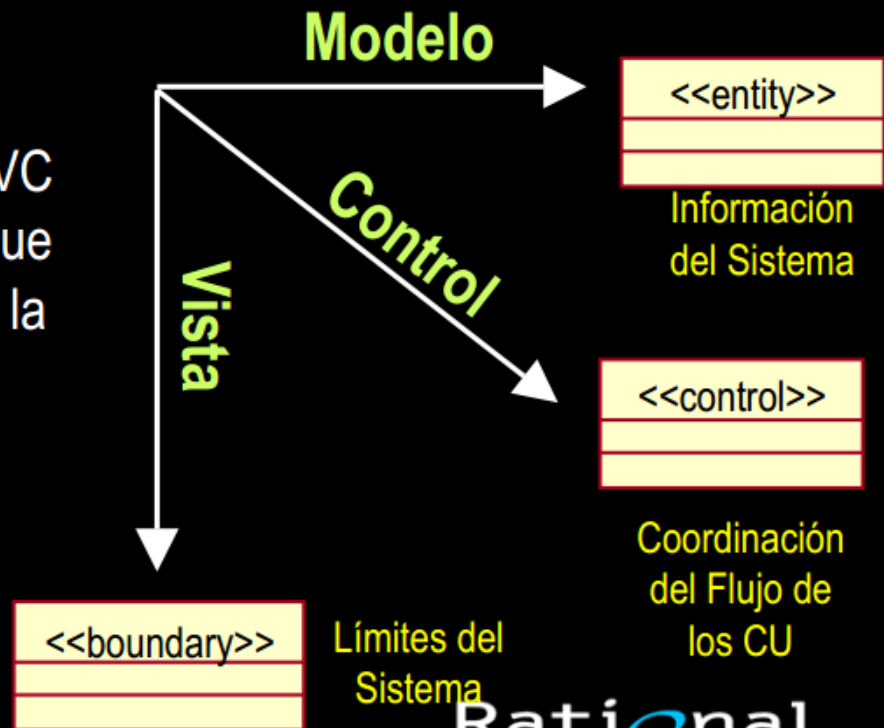
Las clases de análisis se centran en el tratamiento de requerimientos funcionales y pospone los no funcionales denominándolos requerimientos especiales, hasta llegar a las actividades de diseño e implementación.

Las clases de análisis siempre encajan en uno de estos tres estereotipos: **interfaz, control y entidad**.

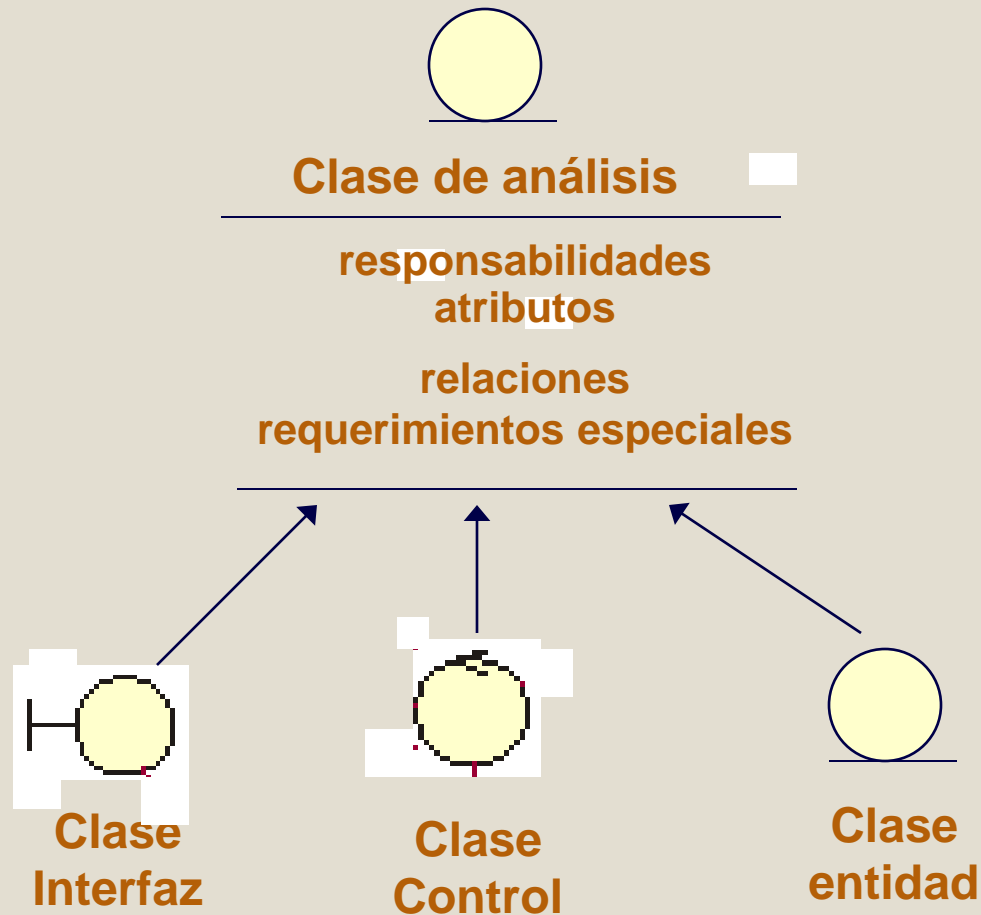
Qué son las clases de análisis

- Son clases estereotipadas segun la metodología OOSE de Ivar Jacobson para crear modelos ideales de objetos

- Esta metodología se basa en el patrón de análisis MVC (Model-View-Controller), que define clases enfocadas a la **separación de responsabilidades** para conseguir componentes extensibles y reutilizables.



Estereotipos de una Clase de Análisis

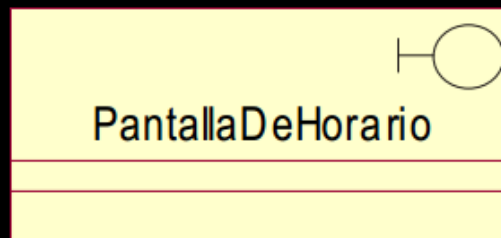
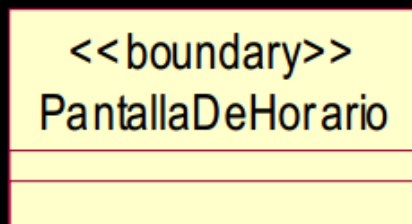


Clase Límite

- Una clase de límite, modela la comunicación entre lo que rodea al sistema y su funcionamiento interno
- Clases de límite típicas
 - Pantallas o interfaces de usuario
 - Reportes
 - Interfaces programáticos a otros sistemas

Ejemplo:

En el caso de uso “Inscribirse en Cursos”, se utiliza una pantalla de horario para que el estudiante ingrese las opciones de cursos

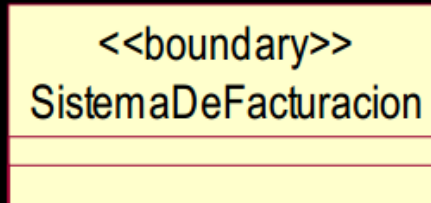


Interfases con otros Sistemas

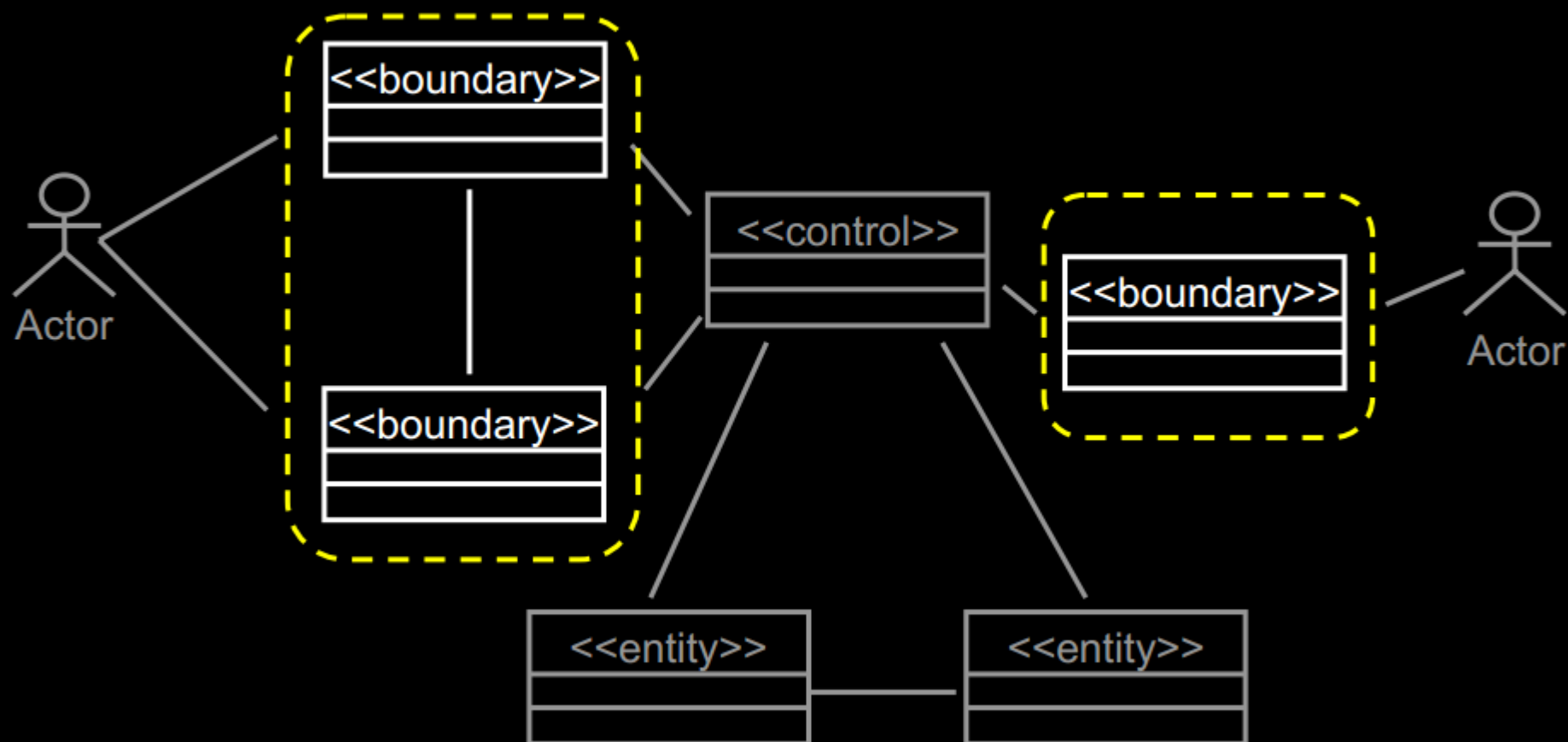
- Una clase de límite también se puede usar para modelar una interfase (API) con otro sistema
- Las características importantes de este tipo de clase de límite son:
 - Las **funciones** que provee el otro sistema
 - La **información** a ser pasada al otro sistema
 - El “**protocolo**” de comunicación usado para “hablar” con el otro sistema

Ejemplo:

En el caso CU “Correr Proceso de Cierre” hay información que debe ser enviada a un Sistema de Facturación externo. Se puede crear una clase de límite llamada SistemaDeFacturación para representar la interfase con el sistema externo.



El Rol de una Clase Límite



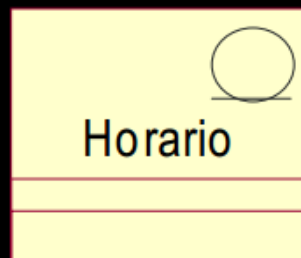
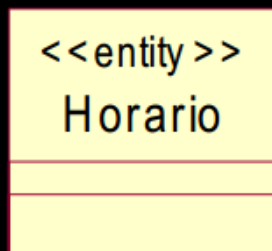
Modelar la interacción entre el sistema y sus alrededores

Clase Entidad

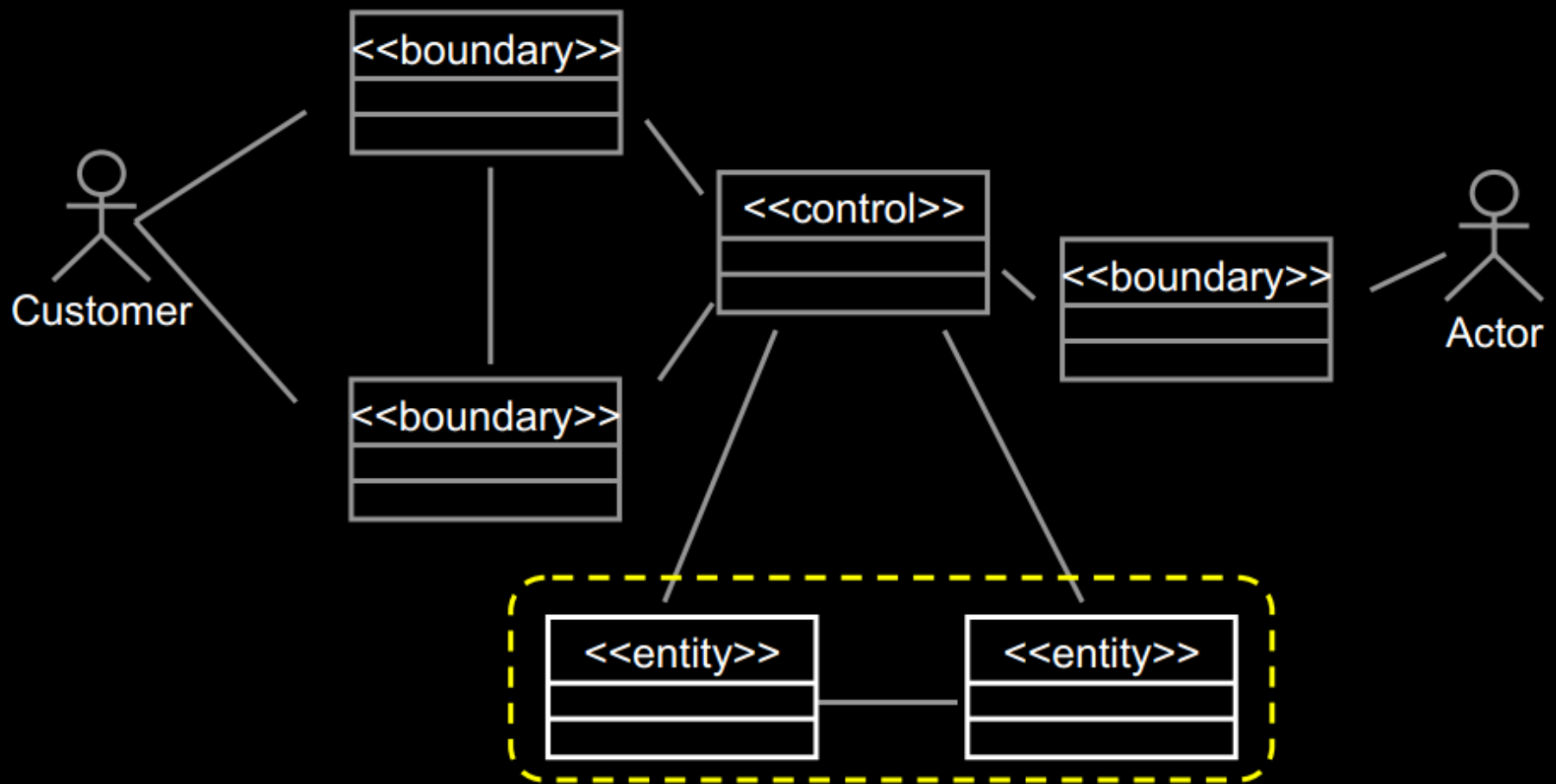
- Una clase de entidad corresponde a las **abstracciones principales** del Modelo Conceptual y modela la estructura y comportamiento asociado a una clase que generalmente es de **larga duración** (persistente).
 - Puede reflejar un fenómeno de la vida real
 - Su comportamiento es independiente de sus alrededores

Ejemplo:

Una de las clases de entidad en el CU “Inscribirse en Cursos” es Horario



El Rol de una Clase de Entidad



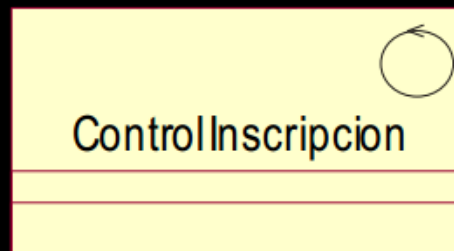
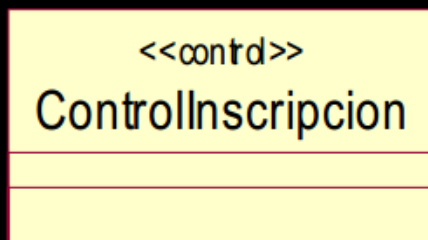
Almacenar y administrar la información en el sistema

Clase Control

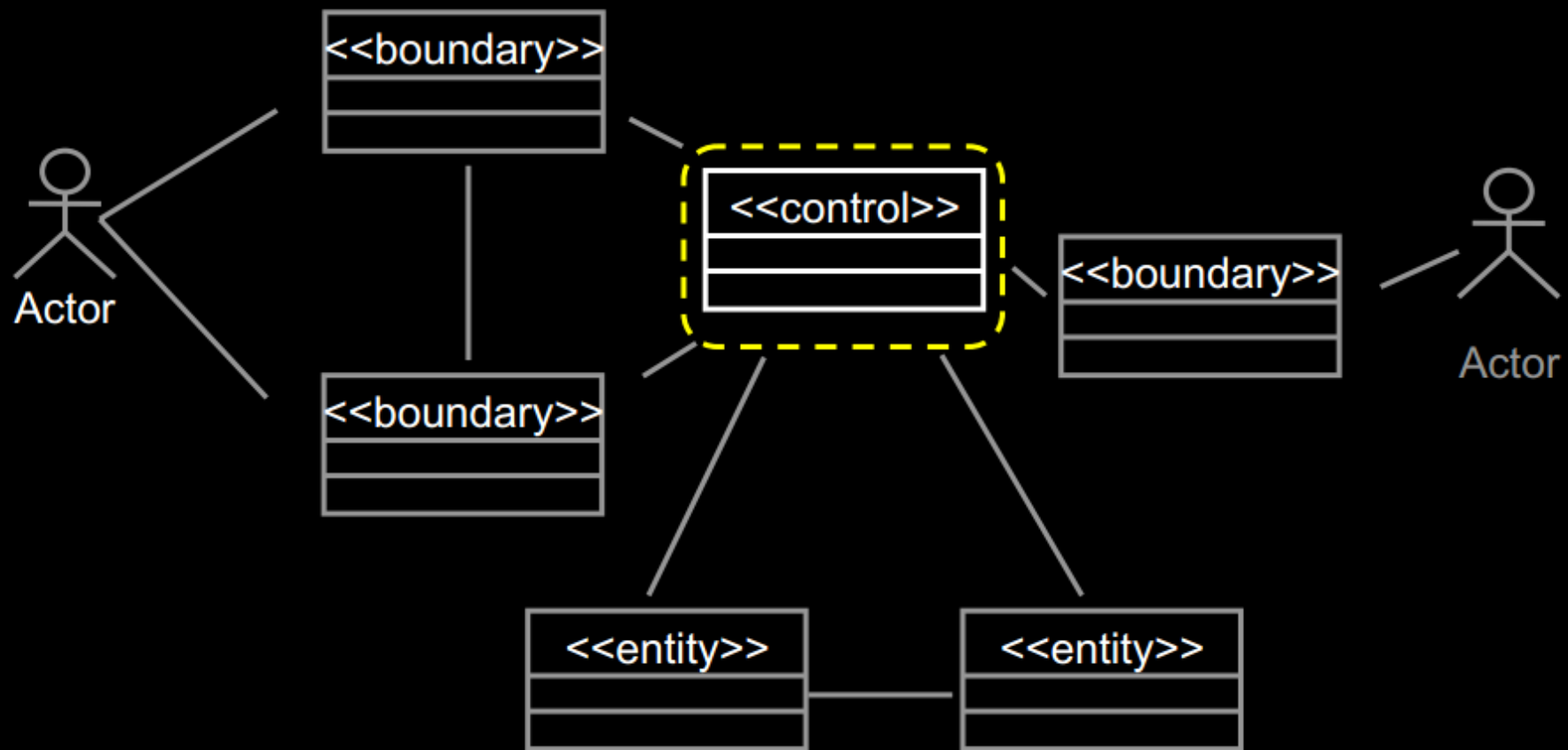
- Una clase de control modela comportamiento de **control** o **coordinación del flujo de eventos** asociado a uno o más CU
 - Sirve como **intermediario** entre las clases de limite y las de entidad
 - Controla la **secuencia o la coordinación de la ejecución** del flujo de eventos enviando mensajes a los objetos controlados
 - Controla aspectos de **conurrencia** para las clases controladas
 - **Crea, modifica y elimina** a los objetos controlados
 - La mayoría de las veces es la implementación de un objeto intangible

Ejemplo:

En el caso de uso “Inscribirse en Cursos”, hay una clase ControllInscripción que coordina el CU



El Rol de una Clase de Control

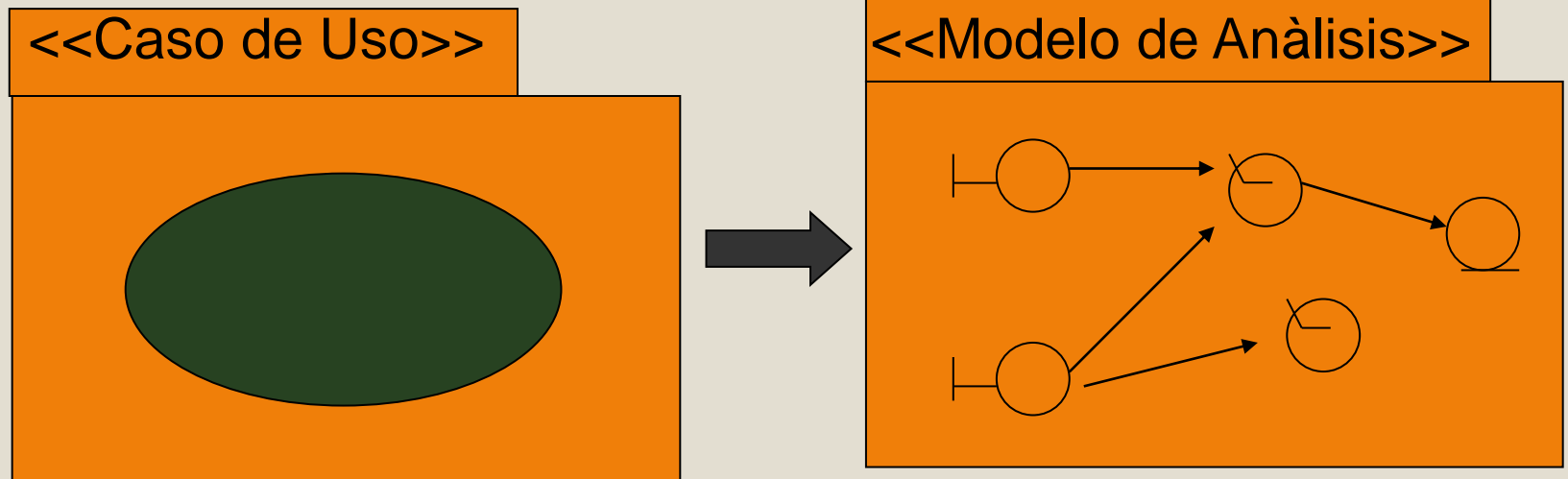


Coordinar el comportamiento del caso de uso

¿Qué es un Diagrama de Clase de Análisis?

- El Diagrama de Clase (DCA) es el diagrama principal para el análisis y diseño de un sistema.
- El objetivo es convertir los casos de uso de sistema en términos de clases y objetos.
- El DCA se desarrolla utilizando la información obtenida en los Casos de Uso.
- Se utiliza para desarrollar diagramas de Secuencia y de Colaboración.

De Diagrama de CU a Modelo de Análisis



Análisis de Escenarios

IDENTIFICACIÓN DE
CLASES/OBJETOS PARTICIPANTES EN
UN CASO DE USO

Escenario: “Crear un nuevo horario”

Flujo básico del CU “Inscribirse en curso”

Estela Gómez escoge del Menú principal la opción “Inscribirse en Cursos” y de la pantalla de inscripción selecciona “Crear un Horario Nuevo”.

Estela consulta la lista de cursos disponibles selecciona los cursos Inglés 101 (Cód. 2189), Geología (Cód. 2177), Historia Mundial (Cód. 7659), Cálculo I (Cód. 2545) como opciones primarias. Luego selecciona como opciones alternas Programación algorítmica (Cód. 4367) y Español (Cód. 8875).

Escenario: “Crear un nuevo horario”

Flujo básico del CU “Inscribirse en curso”

En la pantalla de horario, Estela ingresa los códigos de los cursos escogidos y somete el horario a validación.

El sistema determina que las opciones de Estela cumplen con todas las validaciones y añade a Estela a la lista de estudiantes inscritos de cada grupo indicado en el horario.

El sistema presenta una copia “lista para imprimirse” del horario y Estela la imprime.

Termina el CU.

Análisis de escenarios para identificar clases u objetos que participan en un CU.

1. Refine el escenario a analizar.

2. Haga una lista con los sustantivos del escenario.

No repita los sustantivos.

Sustantivo: Parte de la oración que designa a un ser u objeto.

3. Identifique los sustantivos como sustantivo propio o sustantivo común.

4. Clasifique la participación del sustantivo como:

- **Objeto entidad:** los que almacenan datos persistentes
- **Objeto límite:** los que representan o manipulan interfaces
- **actor, instancia de un objeto, atributo.**
- **Fuera de contexto, etc.**

5. Por cada CU defina una clase control: por ejemplo “ControlLogin”.

Identificación de sustantivos

Estela Gómez escoge del **Menú principal** la **opción** “Inscribirse en **Cursos**” y de la **pantalla de inscripción** selecciona “Crear un **Horario Nuevo**”.

Estela consulta la **lista de cursos disponibles** y selecciona los cursos **Inglés 101** (Código 2189), **Geología** (Código 2177), **Historia Mundial** (Código 7659), **Cálculo I** (Código 2545) como **opciones primarias**. Luego selecciona como **opciones alternas Programación algorítmica** (Código 4367) y **Español** (Código 8875)

En la **pantalla de horario**, Estela ingresa los **códigos** de las opciones escogidas y somete el horario a validación.

El **sistema** determina que las opciones de Estela cumplen con todas las **validaciones** y añade a Estela a la **lista de estudiantes inscritos** de cada **grupo** indicado en el horario.

El sistema presenta una **copia “lista para imprimirse”** del horario y Estela la imprime.

Termina el **CU**.

Lista de sustantivos del escenario

❖ Estela Gómez	→ Sust. Propio	→ actor
❖ Menú principal	→ Sust. propio	→ objeto: cls. límite
❖ opción	→ Sust. común	→ fuera de contexto
❖ cursos	→ Sust. común	→ objeto: cls. entidad
❖ pantalla de inscripción	→ Sust. propio	→ objeto: cls. límite
❖ horario	→ Sust. común	→ objeto: cls. entidad
❖ lista de cursos disp.	→ Sust. común	→ objeto: cls. entidad
❖ Inglés 101	→ Sust. propio	→ inst. clase lis...
❖ Geología	→ Sust. propio	→ inst. clase lis...
❖ Historia mundial	→ Sust. propio	→ inst. clase lis...
❖ Cálculo I	→ Sust. propio	→ inst. clase lis...

Lista de sustantivos del escenario

opciones primarias	→ Sust. común	→ atributo cls horario
opciones alternas	→ Sust. común	→ atributo cls horario
Programación algorit.	→ Sust. propio	→ inst. de clase lis...
Español	→ Sust. propio	→ inst. de clase lis...
pantalla de horario	→ Sust. propio	→ objeto: cls límite
Códigos	→ Sust. común	→ atributo cls horario
Sistema	→ Sust. común	→ fuera de contexto
Validaciones	→ Sust. común	→ acciones
lista de est. inscritos	→ Sust. común	→ objeto: cls entidad
Grupo	→ Sust. común	→ objeto: cls entidad
Browser	→ Sust. común	→ fuera de contexto
copia (para imprimirse)	→ Sust. común	→ objeto: cls límite
CU	→ Sust. común	→ fuera de contexto

Clases tipo Entidad identificadas en Escenario de “Inscribirse en Cursos –Crear un Horario”-

Curso:

una materia ofrecida por la universidad que es parte de un Plan de Estudios del estudiante

ListaCursosDisponibles:

conjunto de todos los cursos a ofrecer en un semestre

Horario:

lista de cursos y grupo (s) escogidos por un estudiante para un semestre

ListaEstudiantesInscritos:

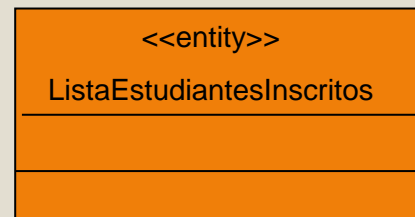
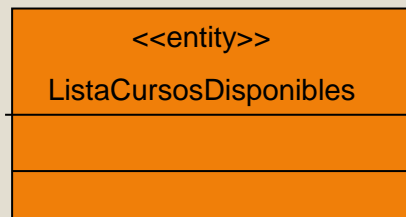
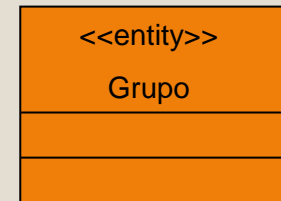
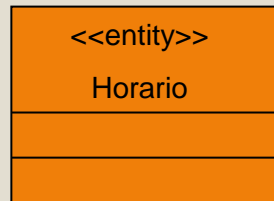
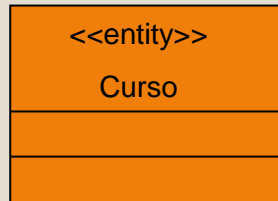
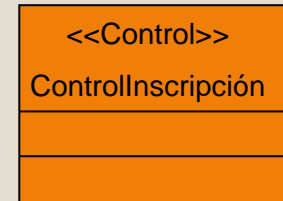
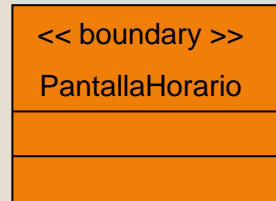
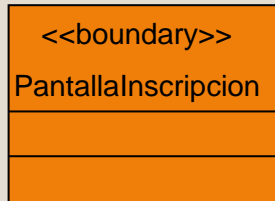
lista de estudiantes matriculados en un curso particular de un grupo

Grupo:

conjunto de cursos abiertos en un salón particular, ofrecidos en un horario específico.

Diagrama de clases resultante: VOPC

Clases participantes: CU “Inscribirse en Cursos”



Diagramas de Clases Participantes

Un diagrama de clases muestra una o más clases en un mismo plano, usando la nomenclatura que se ha presentado antes.

Cada Realización de Caso de Uso (RCU) tiene uno o más diagramas de clases que muestran las clases participantes en el CU y sus relaciones. Tales diagramas son llamados Vista de Clases Participantes (“View of Participating Classes”) lo que se resume como VOPC.

Los VOPC inician muy sencillos y pueden llegar a ser muy detallados y complejos, por lo que se puede necesitar usar varios para cada RCU.

Resumen del Modelo de Análisis

- El Modelo de Análisis es el resultado del proceso de Análisis
- El Modelo de Análisis incluye un modelo Estático y un modelo Dinámico del sistema
- Para desarrollar el Modelo de Análisis debe hacerse “Análisis de Casos de Uso”
- Con el Análisis de Casos de Uso se inicia la Realización de Casos de Uso (RCU)
- El producto más importante del Análisis de Casos de Uso son las Clases de Análisis (VOPC)
- Las Clases de Análisis son el primer paso hacia la creación de componentes ejecutables

Resumen del Modelo de Análisis

- **El estereotipo representa un tipo de clase**
 - Cada Clase de Análisis debe tener un estereotipo
- **Una clase límite modela la comunicación entre lo que rodea al sistema y su funcionamiento interno.**
- **Una clase de entidad modela la información y comportamiento asociado, que es de larga duración (debe ser almacenada)**
- **Una clase de control modela comportamiento de coordinación específico a uno o más casos de uso**
- **Para cada RCU deben hacerse uno o mas VOPC, o diagramas de clases participantes**