

Sesión 2

Jorge Villalobos

Mario Sánchez

Ingeniería de Sistemas y Computación

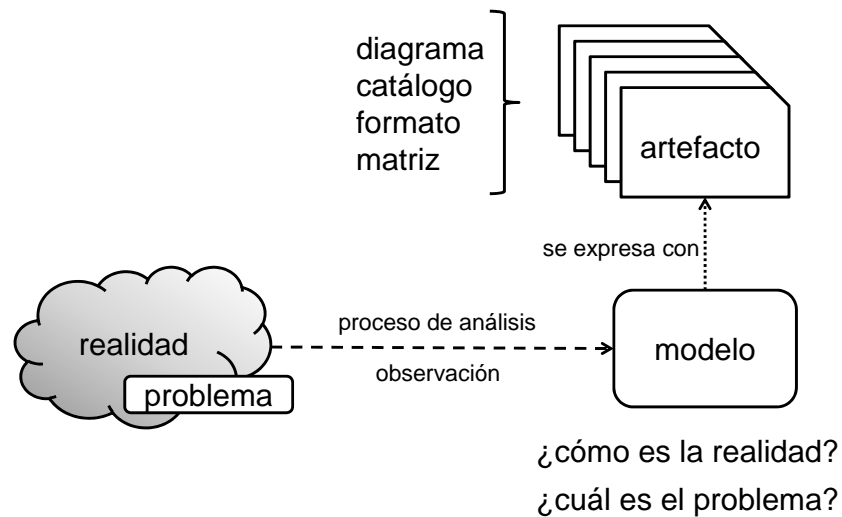


Agenda

1. Contexto
2. Análisis de una aplicación simple
3. Diagramas de contexto
4. Diagramas de clases
5. Diagramas de casos de uso
6. Diagramas de secuencia
7. Árboles de utilidad

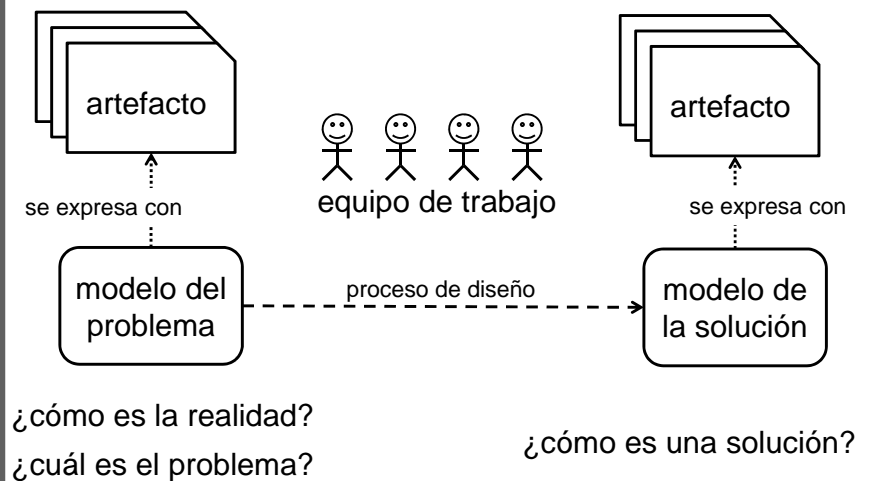
2

Análisis de un problema: objetivo



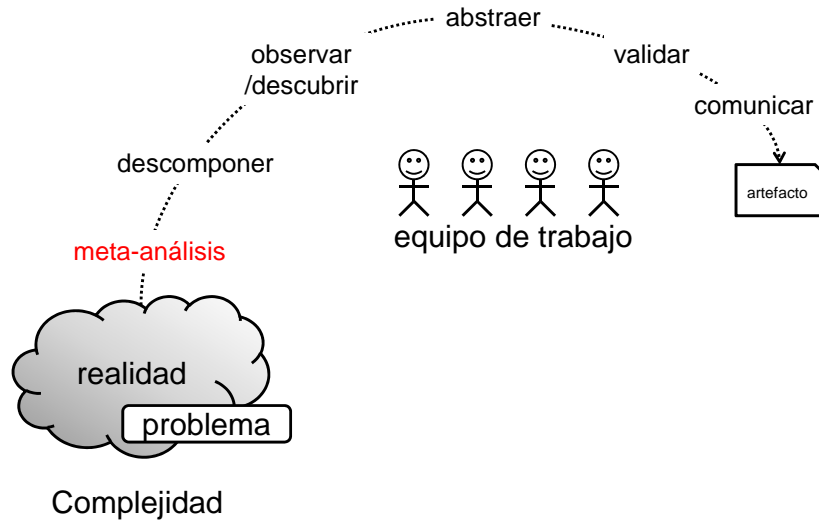
3

Diseño de una solución: objetivo



4

Análisis de un problema: etapas



Algunos conceptos importantes

- Meta-análisis
 - Estrategias para abordar el análisis
 - Alcance horizontal y alcance vertical
 - Información y meta-información
 - Observación vs. diagnóstico
 - Marco de evaluación para un diseño
 - Proceso de diseño
 - Niveles de abstracción en el diseño
 - Propiedades de una representación
- 6

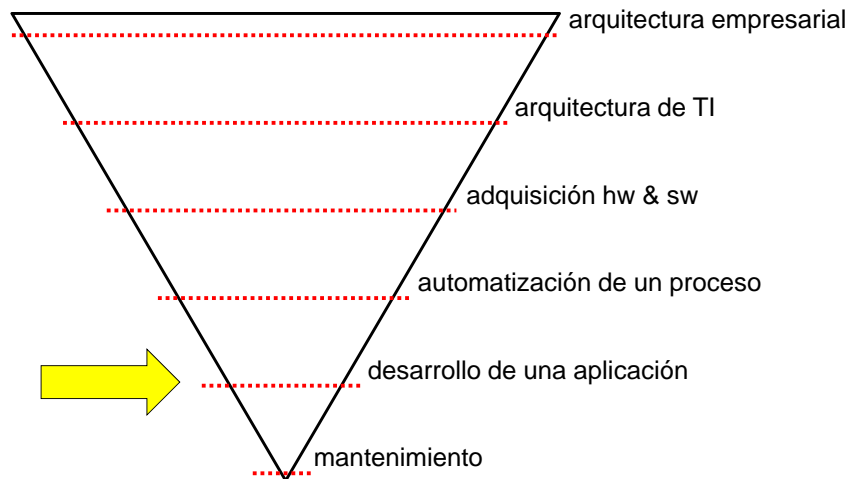
¿Preguntas?



Agenda

1. Contexto
 - ➔ 2. Análisis de una aplicación simple
 3. Diagramas de contexto
 4. Diagramas de clases
 5. Diagramas de casos de uso
 6. Diagramas de secuencia
 7. Árboles de utilidad
- 8

Análisis de una aplicación simple



9

Contexto del problema

- Ya se hizo el análisis empresarial y se conocen los motivadores de negocio y las políticas globales de TI.
- Ya se tiene una arquitectura de solución y se sabe cómo se enmarca la aplicación en ella
- Ya se ha tomado la decisión de desarrollar la aplicación, como parte de un proyecto del mapa de ruta
- Existe un primer planteamiento informal que describe la aplicación
- Ya se han identificado los stakeholders de la aplicación

10

Planteamiento del proyecto

1- Establecer el objetivo del análisis

Los artefactos del análisis serán utilizados por una empresa que va a hacer el desarrollo de la aplicación.

2- Identificar los stakeholders y las fuentes de información

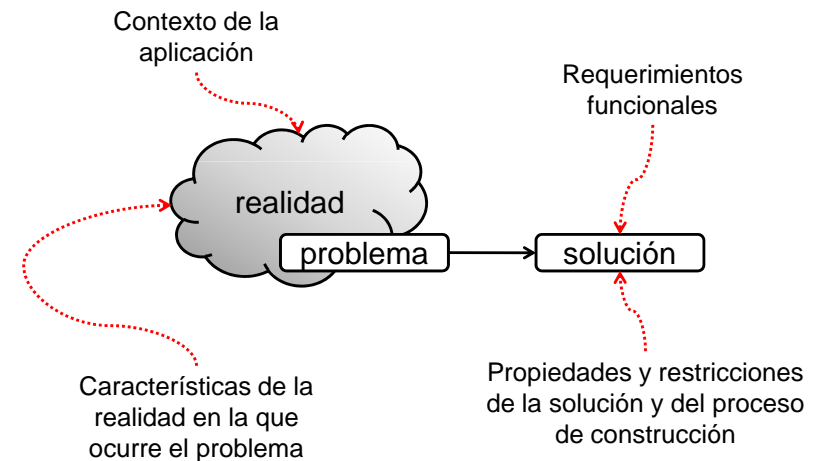
Dados en el contexto.

3- Definir la estrategia

Un solo ciclo.
Análisis por puntos de vista.

11

Puntos de vista típicos



12

Punto de vista: contexto de la aplicación

- Objetivo: entender el contexto en el que se va a desplegar/ejecutar la aplicación:
 - Motivadores de negocio (p.ej. time-to-market)
 - Estrategia de TI (p.ej. multicanalidad)
 - Lineamientos y políticas de TI (p.ej. seguridad)
 - Arquitectura de solución (p.ej. estilo arquitectural)
 - Relaciones con otras aplicaciones o componentes
- Artefactos:
 - Diagramas ad-hoc de contexto
 - Formas, diagramas de arquitectura, etc.

13

Punto de vista: modelo de la realidad

- Objetivo: entender la estructura conceptual del mundo en el que ocurre el problema:
 - Entidades de negocio
 - Relaciones, restricciones, reglas de negocio
 - Comportamiento
- Artefactos:
 - Diagramas de clases y diagramas de objetos
 - Diagramas de estados
 - Diagramas de secuencia
 - Formas y catálogos

14

Punto de vista: requerimientos funcionales

- Objetivo: entender las funciones de negocio que debe proveer la aplicación que se quiere construir y los actores que las van a utilizar:
 - Funciones: entrada → salida (qué, no cómo!)
- Artefactos:
 - Diagramas de casos de uso
 - Formas y catálogos

15

Punto de vista: propiedades & restricciones de la solución

- Objetivo: entender las propiedades que los stakeholders esperan que tenga la aplicación y las restricciones que debe respetar, tanto en su operación como en su construcción:
 - Atributos y escenarios de calidad
 - Modelo de ejecución de la aplicación
- Artefactos:
 - Árboles de utilidad
 - Diagramas de estados, autómatas o redes de petri
 - Diagramas ad-hoc

16

Planteamiento del proyecto

4- Diseñar los entregables

Encabezado de identificación y versionamiento

Definir los entregables:

- Identificar el público objetivo (stakeholders)
- Definir estrategia de comunicación (descomposición?)
- Definir las secciones (niveles de abstracción?)
- Definir los artefactos que incluye cada sección
- Definir los lenguajes
- Definir las herramientas

5- Definir los aspectos metodológicos

17

Agenda

1. Contexto
2. Análisis de una aplicación simple
- ➔ 3. Diagramas de contexto
4. Diagramas de clases
5. Diagramas de casos de uso
6. Diagramas de secuencia
7. Árboles de utilidad

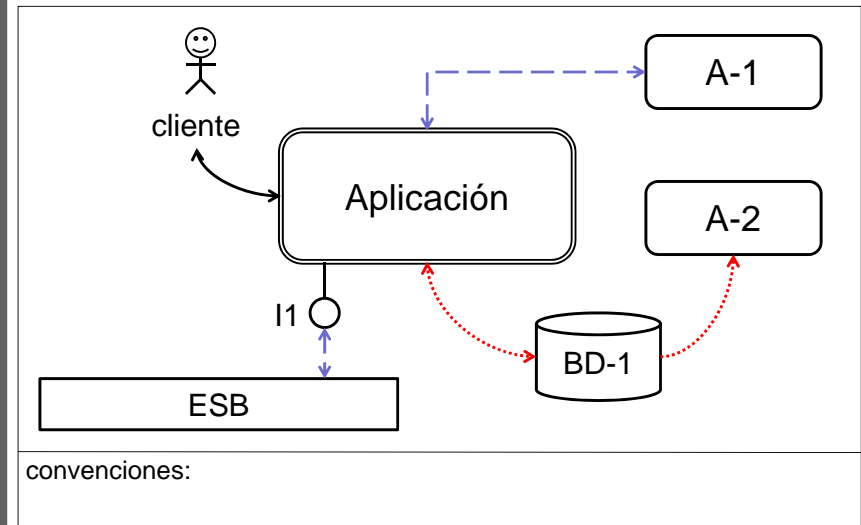
18

Diagramas de contexto

- Pretende describir todo lo que se conoce (y es pertinente) que se encuentra por fuera de la frontera de la aplicación y tiene algún impacto sobre ella
- La idea es ver la aplicación como una caja negra que se relaciona con los demás elementos del contexto
- Son diagramas ad-hoc, en el que se deben hacer explícitas las convenciones

19

Diagramas de contexto: ejemplo



20

Propiedades de una representación

- Fidelidad
- Consistencia
- Fácil de entender
- Estética
- Compatibilidad
- Completitud
- Capacidad de expresión
- Documentación

21

Agenda

1. Contexto
2. Análisis de una aplicación simple
3. Diagramas de contexto
- 4. Diagramas de clases
5. Diagramas de casos de uso
6. Diagramas de secuencia
7. Árboles de utilidad

22

Visión global del proceso

- Objetivo: identificar y estructurar la información del negocio relevante para el sistema
 - ¿Qué existe en el mundo interesante para el problema?
 - ¿Cómo se relacionan esos elementos?
 - ¿Qué restricciones cumplen?
 - ¿Cuáles son las reglas del negocio?

→ Identificar la estructura del mundo del negocio relevante para el problema

23

Visión global del proceso

- Objetivo: expresar el conocimiento del mundo en un lenguaje de descripción de modelos



24

Modelo del mundo

- En esta etapa no se toman decisiones de diseño
- La salida de esta etapa no es implementable, sólo corresponde a una abstracción del mundo del problema
- Antes de implementar hay que incorporar en el diseño los requerimientos no funcionales
- Corresponde a un modelo conceptual y no a componentes de software

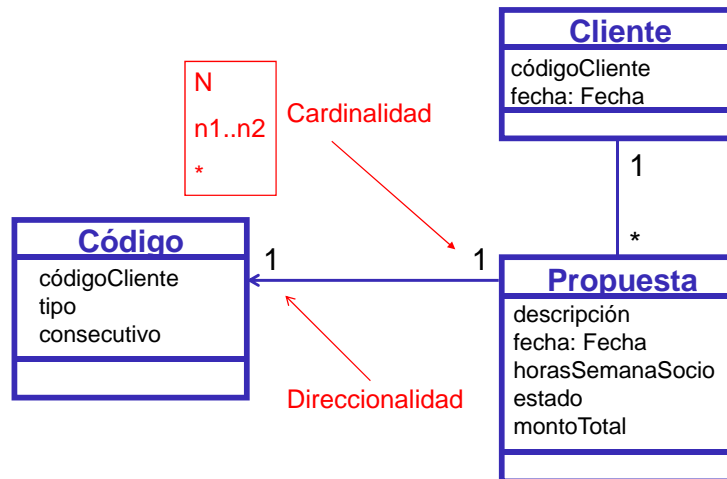
25

Modelo del mundo

- Los grandes retos:
 - Identificar la estructura del mundo
 - Identificar las relaciones entre los elementos
 - Verificar la “navegabilidad” del modelo

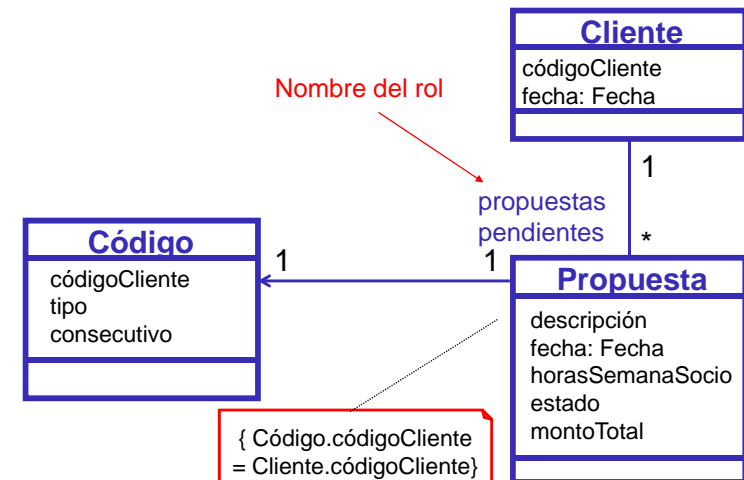
26

Diagrama de clases



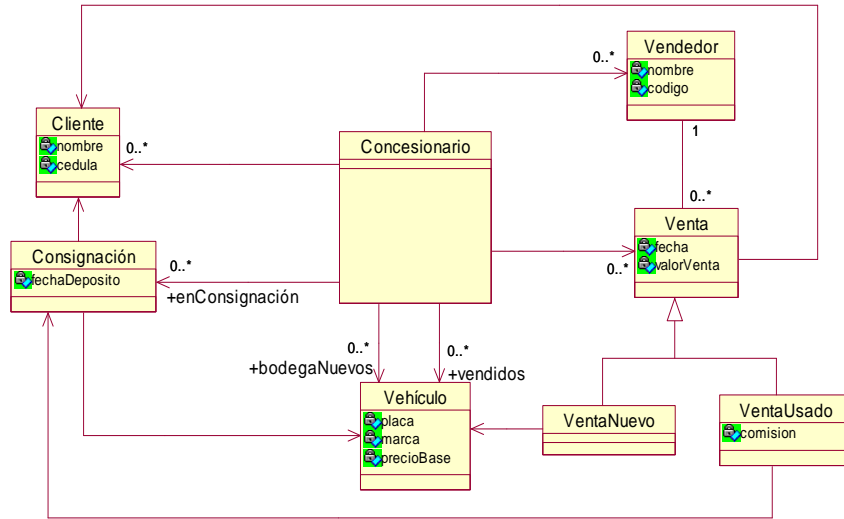
27

Diagrama de clases



28

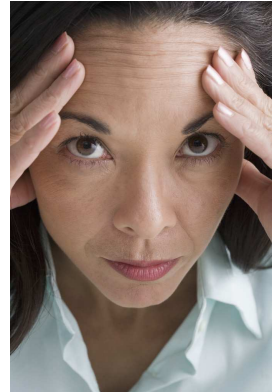
Ejemplo



29

Síntomas de errores...

- Encontrar que una clase **HACE**. Lo correcto es una descripción en términos de una clase **ES**.
- No se identifican instancias de una clase
- Sólo se identifica una instancia
- Hay dificultad para encontrar atributos y asociaciones
- Hay dificultad para encontrar operaciones: **sólo se encuentra un servicio**



30

Síntomas de errores...

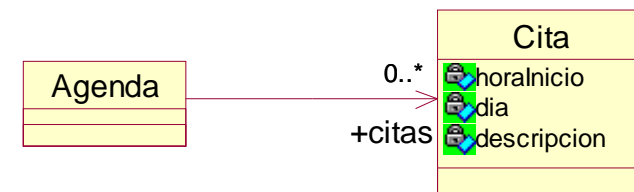
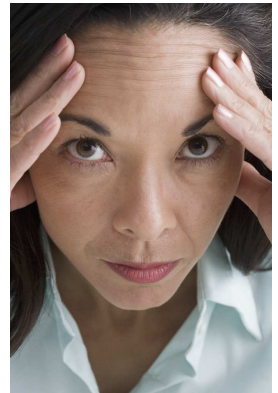
- Objetos con iniciativa propia: **confusión de actores y clases**
- Sólo se tiene información no relevante para el problema
- Las clases corresponden a una vista funcional del problema: **división del problema de la manera clásica**



31

¿Qué no está bien?

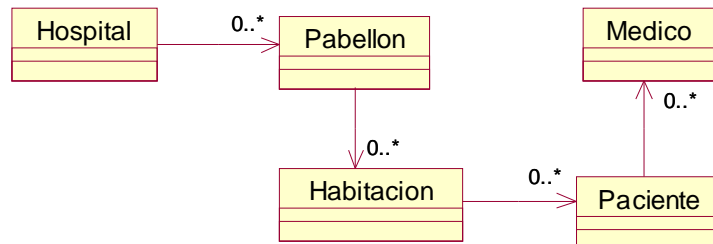
- Se quiere construir un programa para manejar las citas de una persona. Cada cita tiene una duración de 30 minutos y una descripción.



32

¿Qué no está bien?

- Se quiere construir un programa para manejar la información de los pacientes de un hospital.



33

Aspectos metodológicos

- Enfoque funcional
- Enfoque bottom-up
- Enfoque top-down
- Enfoque híbrido

34

Aproximación funcional

- Los requerimientos funcionales son la guía para la construcción del modelo:
 - Qué elementos aparecen referenciados en los RF y cómo deben interactuar entre ellos?
- Se corre el riesgo de definir una arquitectura tradicional (guiada por el problema)
- Se construye el diagrama de clases y el diagrama de secuencia a la vez

35

Aproximación bottom-up

- Qué elementos hay en el mundo del problema?
- Partir de los sustantivos del enunciado (aproximación un poco ingenua)
- Partir de una clasificación de conceptos
- Definir quiénes participan en el problema, y después mirar cómo se relacionan entre ellos
- Buscar las asociaciones usando una clasificación

36

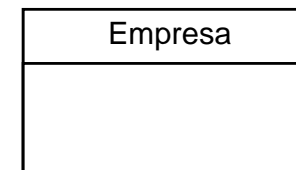
Aproximación bottom-up

- Es fácil comenzar
- Difícil saber si el modelo está completo, o saber en qué puntos hay elementos por refinar

37

Aproximación top-down

- Se basa en la idea de descomponer el mundo, partiendo del elemento que lo incluye todo
- En cada instante, el modelo es completo, sólo que menos refinado



38

Aproximación top-down

- Las asociaciones surgen dentro del proceso de descomposición, y no en una etapa posterior de conexión
- Un elemento puede modelar durante el proceso un submundo completo que sólo será refinado más adelante
- Dividir y conquistar, aplicado a los conceptos del mundo

39

Metodología propuesta

- **ETAPA 1: Identificar los elementos del mundo que participan en el problema**
 - Comenzar por el objeto que representa el mundo completo
 - Encontrar sus componentes o partes hasta llegar a elementos básicos
 - Localizar elementos con estado interno

40

Clases

- **ETAPA 1: Identificar los elementos del mundo que participan en el problema**
 - No necesariamente son elementos físicos
 - De lo conocido a lo desconocido
 - Colocar una clase en el diagrama por cada elemento en el mundo que deba ser incluido en el modelo
 - Nombre significativo y pronunciable
 - Usar minúsculas - mayúsculas (FicheroTemas)

41

Clases...

- **ETAPA 1: Identificar los elementos del mundo que participan en el problema**
 - En casos complejos, ilustrar con ejemplos utilizando un diagrama de objetos

42

Atributos

- **ETAPA 2: Enriquecer la descripción de las clases con sus atributos**
 - Modelar las características de los objetos que posiblemente correspondan a tipos básicos (int, String, Date, etc.)
 - No se modelan agrupamientos
 - Asignar nombre significativos a los atributos
 - Utilizar el lenguaje del mundo del problema
 - Tomar tan pocas decisiones como sea posible
 - No deben tener ningún orden específico

43

Relaciones Estructurales

- **ETAPA 3: Enriquecer el diagrama de clases con otras asociaciones:**
 - Buscar relaciones entre clases del modelo, que no se obtuvieron por descomposición
 - Asignar un nombre a cada rol de la asociación
 - Definir la cardinalidad de la asociación
 - Definir la direccionalidad de la asociación
 - Tomar tan pocas decisiones como sea posible
 - Verificar si algún atributo se modela mejor como una asociación a otro concepto

44

Restricciones

- **ETAPA 4: Asociar con las clases y las asociaciones la noción de validez en el mundo**
 - Determinar todas las restricciones de los atributos, con respecto al dominio de valores válidos
 - Determinar todas las relaciones que deben satisfacer los atributos y asociaciones

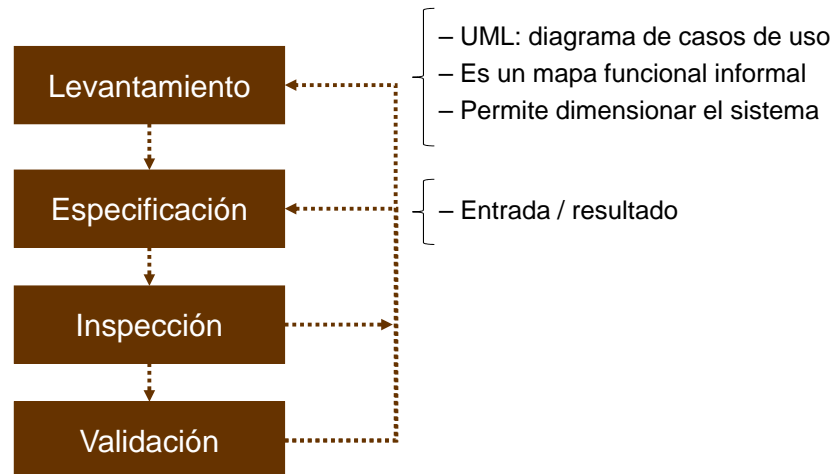
45

Agenda

1. Contexto
2. Análisis de una aplicación simple
3. Diagramas de contexto
4. Diagramas de clases
- ➔ 5. Diagramas de casos de uso
6. Diagramas de secuencia
7. Árboles de utilidad

46

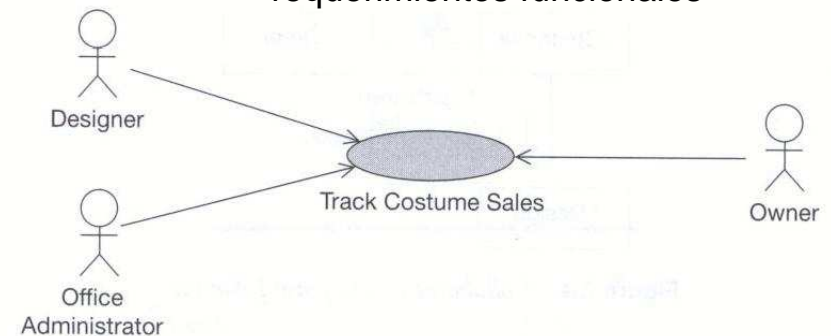
Análisis de requerimientos funcionales



47

UML: Diagrama de casos de uso

Es una herramienta gráfica e informal, para apoyar el levantamiento de requerimientos funcionales



48

Diagrama de casos de uso

- Actores:



para quién



- Casos de uso:



para hacer qué

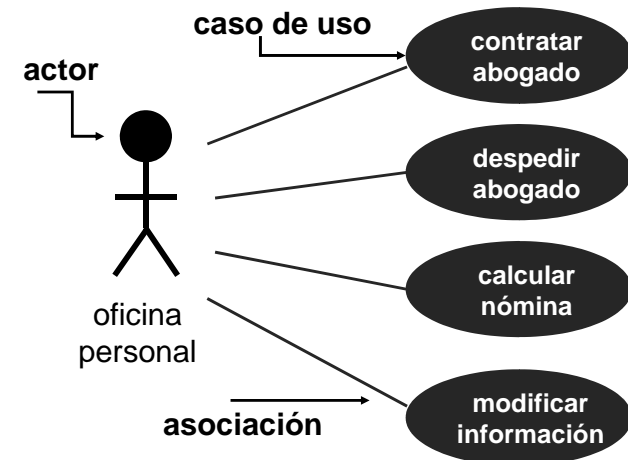


- Asociación entre actores y casos de uso:



quién puede hacer qué

Casos de uso - ejemplo



Actores

- Un actor es un usuario potencial del sistema
- Se modelan los ROLES que juegan estos usuarios frente al sistema
- Se identifican respondiendo a lo siguiente:
 - Para qué se diseña el sistema?
 - A quién está dirigido?
 - Quiénes interactúan directamente con el sistema?
 - Quiénes supervisan, reciben información, mantienen el sistema?

Actores

- Se les asigna un nombre significativo (sustantivo)
- Se diferencian de los elementos del modelo del negocio en que:
 - Tienen iniciativa
 - No tienen estado

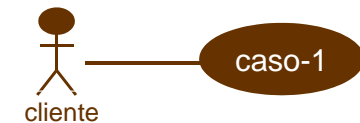
Casos de uso

- Un caso de uso es una operación que puede realizar un actor sobre el sistema
- **Especifica QUÉ y no CÓMO**
- Puede ser grande o pequeño
- Alcanza un objetivo discreto de un usuario
- **Un caso de uso se debe refinar hasta que se pueda describir en términos de entradas y salidas simples (atómico)**
- Es conveniente expresar las operaciones como verbos concretos (p. ej. "calcular nómina")

53

Asociaciones

- ¿Qué casos de uso maneja un actor?
- La asociación se representa con una línea
- **La línea representa todo el flujo de información e interacción entre el sistema y el actor**



54

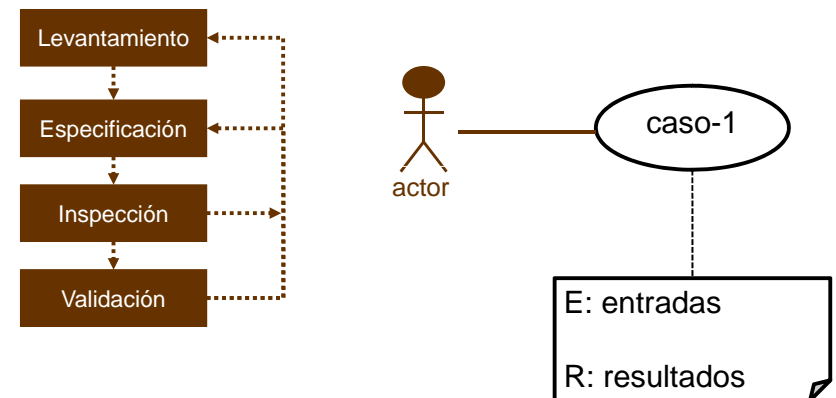
Un caso de uso NO debe:

- Expresar detalles de interacción
- Interactuar con otro caso de uso
- Corresponder a una descomposición funcional
- Corresponder a un diagrama de estados
- Corresponder a una acción del modelo del mundo
- Representar un conjunto de funciones
- Incluir cualquier RNF



55

Especificación de requerimientos funcionales



56

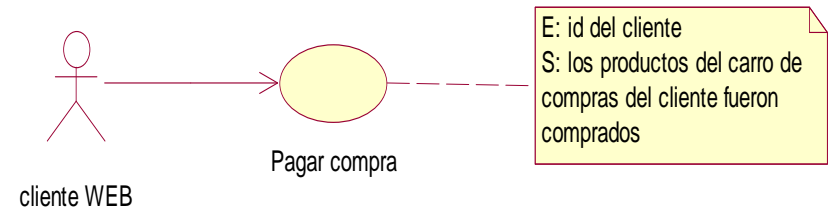
Especificación de un requerimiento funcional

- Identificador
- Nombre
- Resumen
- Actores involucrados
- Flujo normal de eventos
- Caminos alternativos
- Caminos de excepción
- Entradas
- Precondiciones
- Salidas
- Postcondiciones
- Prioridad
- Autor
- Fecha

57

Flujo de eventos

En algunos casos, en lugar de solo dar la entrada y la salida, es importante describir el flujo de eventos (actor-sistema).



58

Flujo de eventos

El actor siempre comienza:

“(1) el cliente indica que va a pagar lo que tiene en su carro de compras, (2) el sistema pide al cliente su identificación, (3) el cliente suministra su ID, (4) el sistema verifica que sea correcto y totaliza el valor de su carro de compras, (5) el sistema confirma si desea adquirir todo, (6) el cliente acepta la compra, (7) la compra se hace efectiva”

Puede haber iteraciones, y representa el camino normal más simple

59

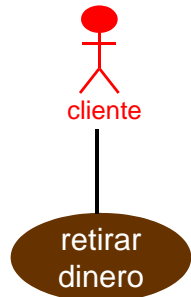
Flujo de eventos

- Se debe tener cuidado de no entrar en el diseño de la interacción o incluir RNF
- Se deben tratar aparte los caminos excepcionales de eventos (manejo de error, situaciones anormales, etc.):

“si el cliente da un ID incorrecto, el flujo de eventos se reinicia. Si el cliente se equivoca 3 veces consecutivas, el sistema cierra la sesión”

60

Flujo de eventos: ejemplo

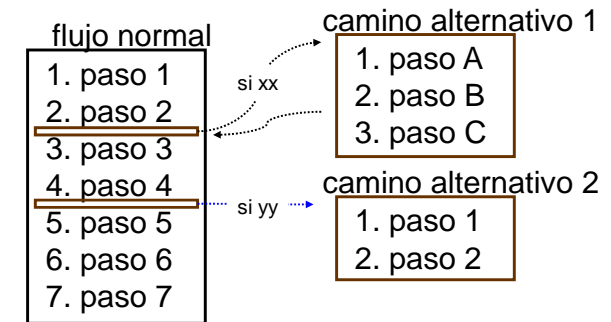


1. Insertar tarjeta
2. Validar tarjeta
3. Solicitar el código personal
4. Ingresar el código personal
5. Validar el código personal
6. Solicitar la cantidad de \$
7. Ingresar la cantidad
8.

61

Caminos alternativos

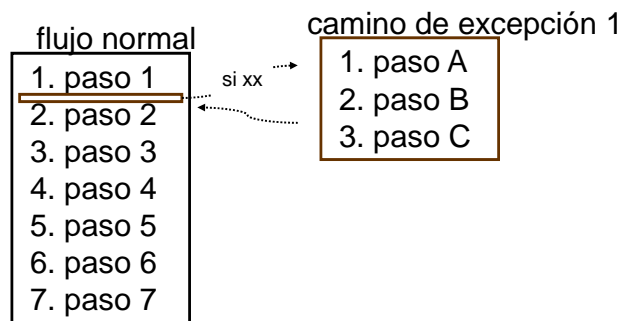
- Caminos alternativos: posibles variantes del flujo normal de eventos. No es el comportamiento típico, pero es menos usual, sin ser un error. **Se debe indicar en qué punto del flujo normal se inicia.**



62

Caminos de excepción

- Caminos de excepción: situaciones anormales del flujo de eventos. Representan la manera de manejar y recuperarse de errores. **Se debe indicar en qué punto del flujo normal se inicia.**



63

Precondiciones y postcondiciones

- Contrato del RF = **pre** + **post**
- Precondición:** condiciones sobre las entradas y el estado del modelo del negocio antes de iniciar la acción
- Postcondición:** estado del modelo del negocio y/o características de las salidas en casos normales

64

Escenarios

- Un escenario es una secuencia específica de acciones que ilustra el comportamiento de un requerimiento (ejemplos)
- Se utiliza para casos de uso complejos
- Es conveniente separar escenarios principales (asociados con flujos normales de eventos) de escenarios secundarios (asociados con flujos excepcionales de eventos)

65

Escenarios: ejemplo

contexto:

Juan tiene en su carrito de compras el libro A (\$100) y el CD B (\$59).

escenario:

1. Juan pide totalizar el valor de la compra
2. El sistema informa que el valor total es de \$159
3. Juan pide pagar el valor de la compra
4. El sistema propone pago por cheque o por tarjeta de crédito VISA
5. Juan teclea su número de tarjeta de crédito, pero da un número incorrecto
6. El sistema le informa que no es posible hacer la compra y le propone de nuevo pago por cheque o pago por tarjeta de crédito VISA
7. ...

66

Agenda

1. Contexto
2. Análisis de una aplicación simple
3. Diagramas de contexto
4. Diagramas de clases
5. Diagramas de casos de uso
- ➔ 6. Diagramas de secuencia
7. Árboles de utilidad

67

Objetivo

Análisis

Diseño

Especificación de requerimientos

Diagrama de clases

Evaluar si el modelo del mundo soporta los requerimientos planteados y ajustar el modelo del mundo

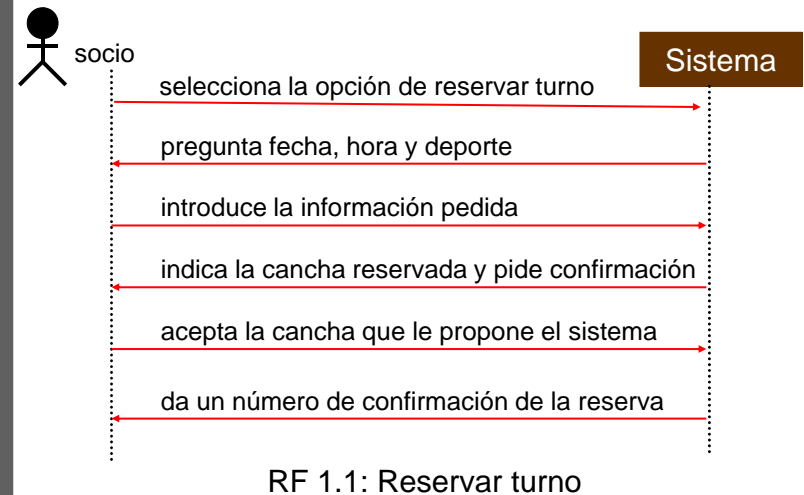
68

Diagramas de secuencia

- Diagrama de UML para visualizar y estudiar los aspectos dinámicos de cualquier sistema
- Principales usos:
 - Describir y detallar casos de uso complejos
 - Describir y estudiar un método complejo
 - Validar un modelo conceptual, contra unos requerimientos funcionales
 - Describir protocolos de comunicación
 - ...

69

Un ejemplo de uso: flujo de eventos



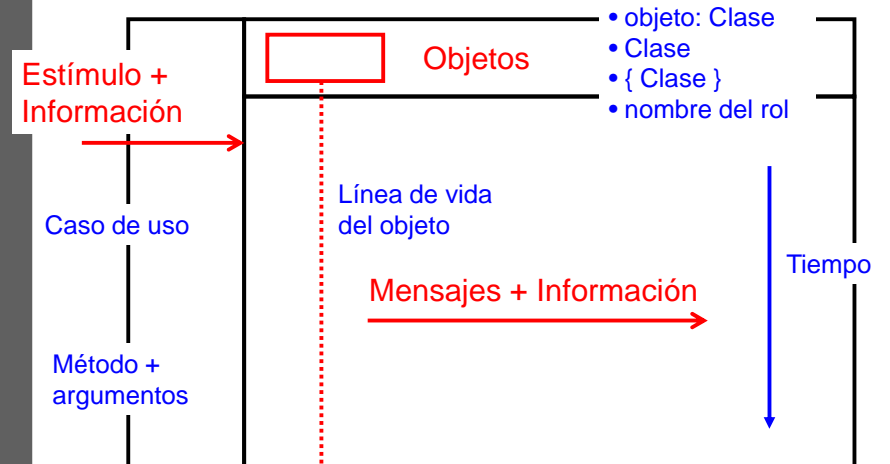
70

Diagramas de secuencia

- El tiempo va expresado de arriba hacia abajo
- No se entra en detalles de diseño de la interacción
- Se explica el proceso normal que sigue el caso de uso (flujo normal de eventos)
- No se muestran las situaciones anormales o de error (camino alternativo o de excepción)
- Puede ayudar a definir con el cliente el “proceso” de un caso de uso
- La dirección de la flecha indica quién tiene la iniciativa

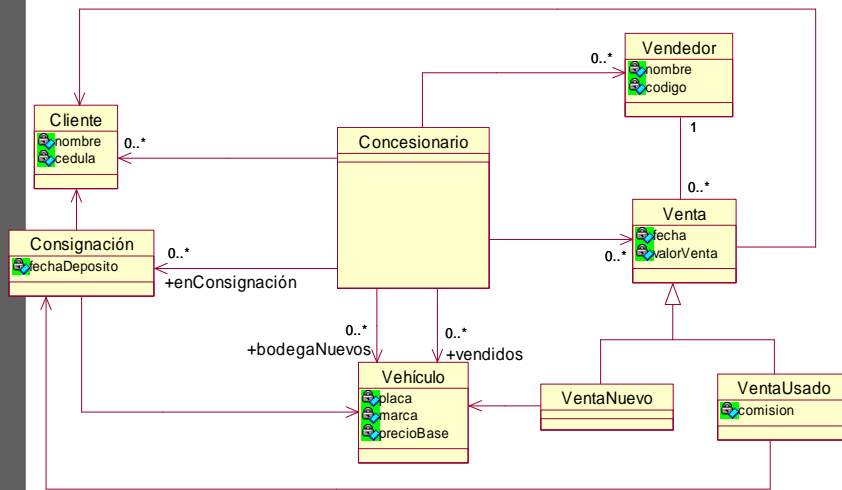
71

DS: estructura general



72

Ejemplo:



73

Ejemplo

- Haga el diagrama de secuencia del requerimiento funcional de vender un carro nuevo, para el cual entra: nombre y cédula del comprador (suponiendo que es un nuevo cliente del concesionario), código del vendedor, placa del carro y valor de venta

74

Ejemplo

- Haga el diagrama de secuencia del requerimiento funcional de vender un carro usado, para el cual entra: cédula del comprador (suponiendo que es un cliente del concesionario), código del vendedor, placa del carro y valor de venta.

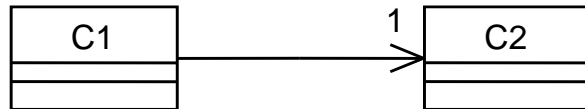
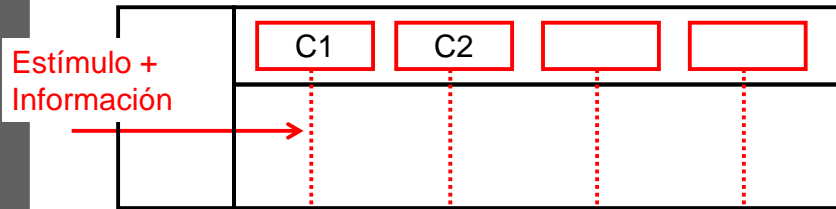
75

DS estructura general

- Permite ver la reacción de un requerimiento, o la reacción a un método
- El primer objeto que aparece es el receptor del estímulo
- En la validación del modelo, el primer objeto que aparece atendiendo un RF es la clase que representa el modelo del mundo
- En la zona de mensajes se ponen los intercambios de control e información de los objetos
- No se dibuja el retorno de los métodos

76

DS en validación del modelo



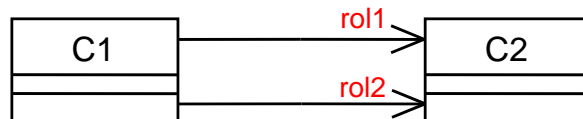
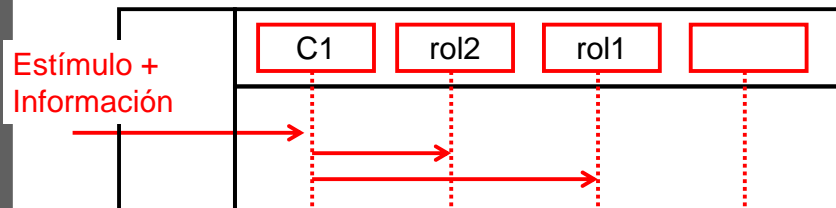
77

DS en validación del modelo

- Un objeto de la clase C2 puede aparecer en la reacción si:
 - Hay un objeto de la clase C1 en el diagrama de secuencia, y
 - En el diagrama de clases hay una asociación de C1 a C2 con cardinalidad 1, o
 - C1 tiene un atributo de la clase C2
- Se dibuja el objeto en el diagrama después del objeto de C1

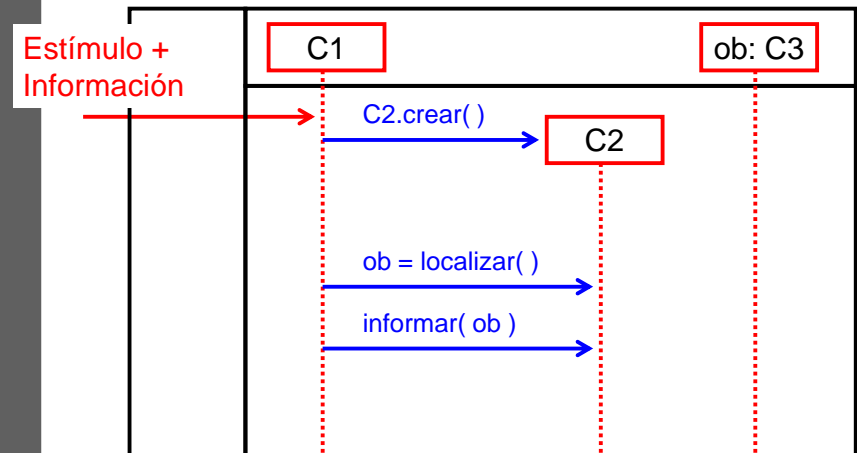
78

DS en validación del modelo



79

DS en validación del modelo



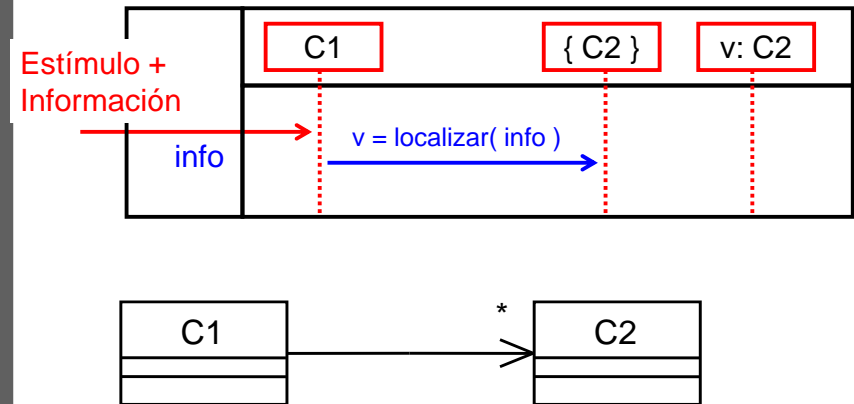
80

DS en validación del modelo

- Un objeto de la clase C2 puede aparecer en la reacción si:
 - Un mensaje a un objeto que ya está en el diagrama retorna un objeto de la clase C2, o
 - Cualquier objeto del diagrama de secuencia invoca la constructora de la clase C2

81

DS en validación del modelo



82

DS en validación del modelo

- Una contenedora de objetos de C2 puede aparecer en la reacción si:
 - Hay un objeto de la clase C1 en el diagrama de secuencia, y
 - En el diagrama de clases hay una asociación de C1 a C2 con cardinalidad distinta de 1

83

DS en validación del modelo

- Entre dos objetos del diagrama se ponen los mensajes que se envían y la información que viaja con ellos
- Si hay más de un caso posible es recomendable hacer un diagrama por cada uno
- No se hace manejo de error: se estudia la reacción si todo funciona bien
- Se entra sólo en los detalles interesantes de la reacción

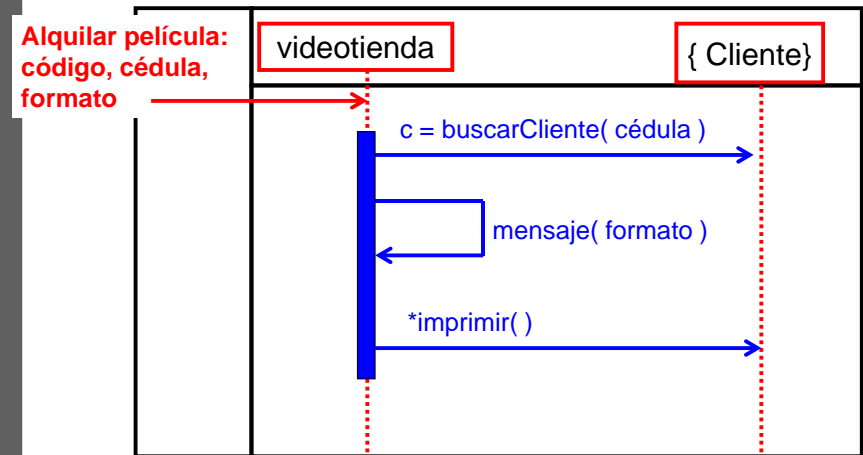
84

DS en validación del modelo

- En el mensaje se coloca el nombre y la información que se necesita
- La información puede provenir del inicio del diagrama o de objetos que han aparecido durante la reacción
- Un objeto se puede enviar mensajes a sí mismo
- Con * se indica que el mensaje se envía múltiples veces (contenedoras)

85

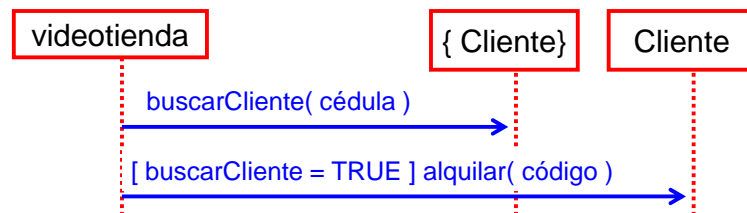
DS en validación del modelo



86

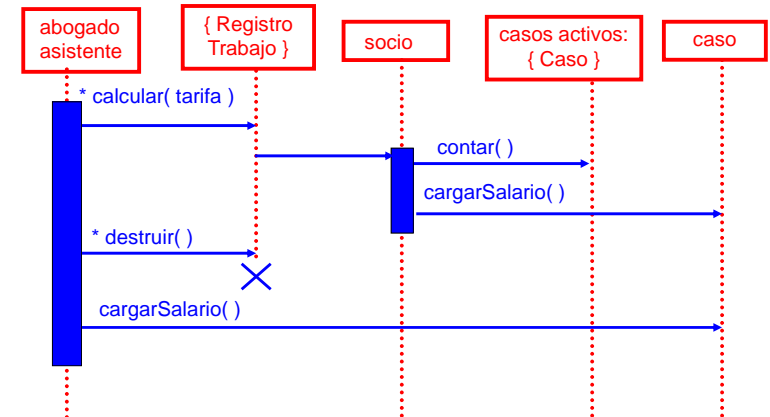
DS en validación del modelo

- Se puede indicar explícitamente la destrucción de un objeto
- Como parte del mensaje se puede indicar el caso que se está trabajando



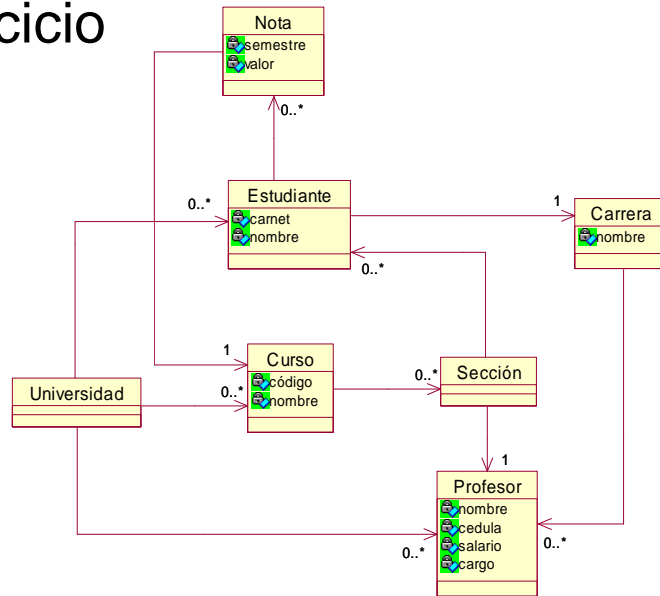
87

Ejemplo...



88

Ejercicio



89

Ejercicio

- **R1:** Cambiar una nota de un estudiante, dado su carnet, el código del curso, el semestre en que lo vio y la nueva nota.

90

Ejercicio

- **R2:** Agregar un estudiante a una sección de un curso, dado el carné del estudiante, el código del curso y el número de la sección

91

Ejercicio

- **R3:** Generar un reporte con la asignación de profesores, indicando qué secciones dicta de qué cursos.

92

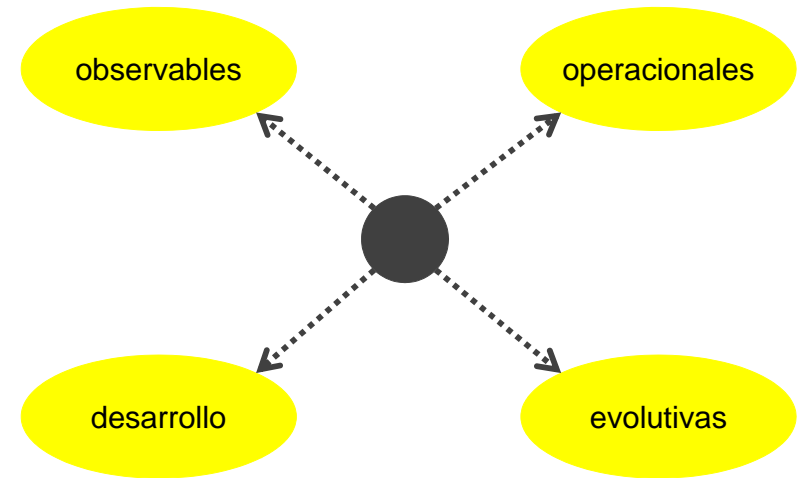
Agenda

1. Contexto
2. Análisis de una aplicación simple
3. Diagramas de contexto
4. Diagramas de clases
5. Diagramas de casos de uso
6. Diagramas de secuencia
- ➔ 7. Árboles de utilidad

93

Atributos de calidad

Ayudan a expresar requerimientos



94

Atributos de calidad

Ayudan a expresar requerimientos

observables

Performance (eficiencia)	Tiempo de respuesta
Reliability (confiabilidad)	Probabilidad de realizar las operaciones correctamente
Availability (disponibilidad)	Porcentaje de tiempo que el sistema se encuentra operando

95

Atributos de calidad

Ayudan a expresar requerimientos

operacionales

Throughput (capacidad)	Volumen de trabajo o de información (solicitudes) que atiende por unidad de tiempo
Security	Capacidad del sistema para prevenir su uso indebido (abusos)
Testability	Esfuerzo necesario para detectar y aislar adecuadamente los errores
Serviceability	Esfuerzo necesario para reparar los errores del sistema
Manageability (administrable)	Esfuerzo necesario para administrar la solución, medido en tiempo, recursos, dinero, etc.

96

Atributos de calidad

Ayudan a expresar requerimientos

Realizability	Probabilidad de que un sistema o funcionalidad se pueda construir
Planability	Probabilidad de que el plan de ejecución y los costos estimados sean reales (confianza sobre el plan que genera la solución)

desarrollo

97

Atributos de calidad

Ayudan a expresar requerimientos

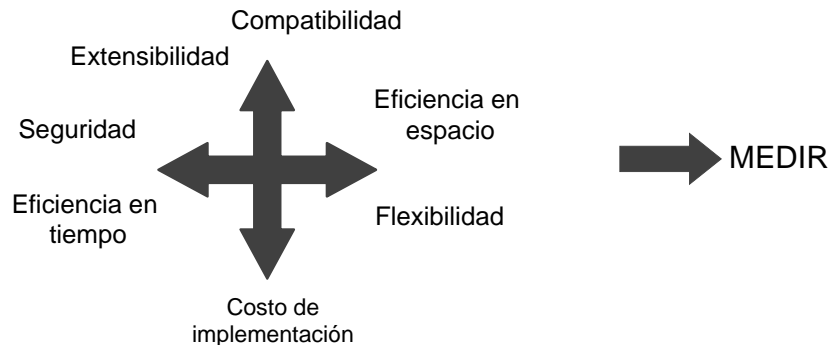
Scalability	Capacidad del sistema de mantener la calidad del servicio a medida que la carga aumenta
Flexibility	Capacidad del sistema de permitir cambios de configuración como respuesta al contexto de ejecución
Extensibility	Esfuerzo ahorrado para adicionar nuevas funcionalidades al sistema
Reusability	Esfuerzo ganado al poder utilizar componentes existentes
Portability	Esfuerzo ahorrado al migrar a una infraestructura diferente
Maintainability	Esfuerzo ahorrado para corregir errores del sistema

evolutivas

98

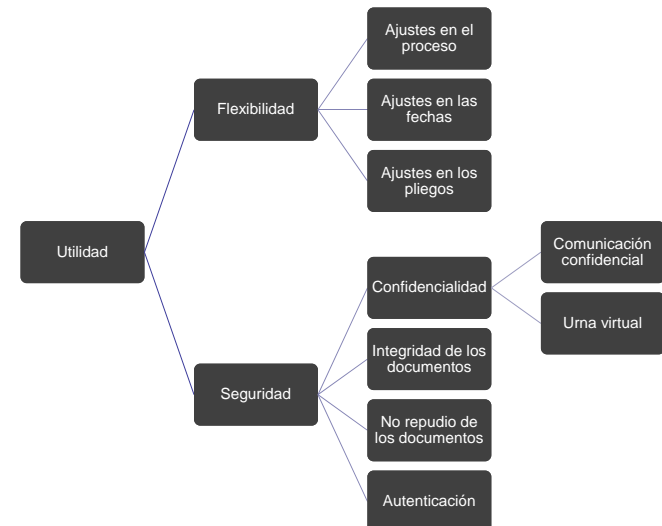
Trade-off

“A **trade-off** is a situation that involves losing one quality or aspect of something in return for gaining another quality or aspect”



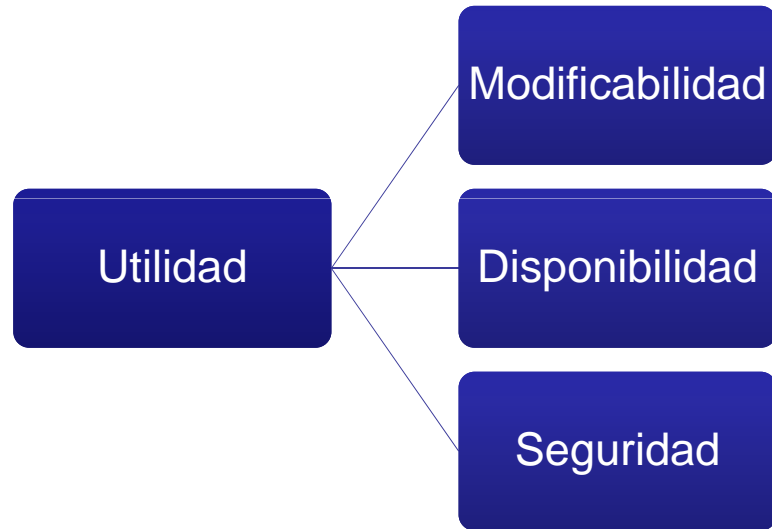
99

Árbol de utilidad: ejemplo



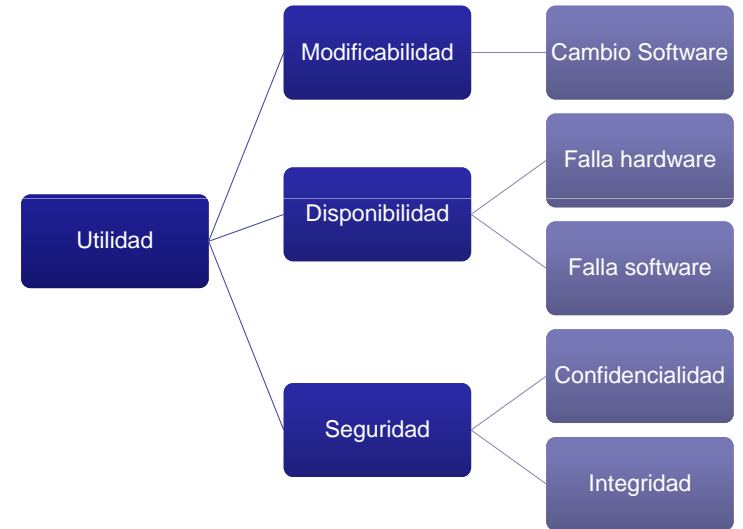
100

Arboles de Utilidad



101

Arboles de Utilidad



102

Arboles de Utilidad



103

Análisis y Diseño de Servicios Escenarios de Calidad - Modificabilidad

Elemento	Descripción
Fuente	Cliente
Estimulo	Crear una promoción para clientes prepago
Artefacto	Sistema de mercadeo
Ambiente	Condiciones normales
Respuesta	La promoción es creada y el nuevo producto queda disponible para los usuarios finales
Medida Respuesta	No se requieren más de 4 horas/hombre del equipo de tecnología para la creación del nuevo servicio

104

Análisis y Diseño de Servicios Escenarios de Calidad - Desempeño

Elemento	Descripción
Fuente	Cliente
Estimulo	El usuario solicita una cotización de productos
Artefacto	Sistema de pedidos
Ambiente	Condiciones normales
Respuesta	La solicitud es creada y el usuario es notificado sobre el inicio del proceso de su solicitud
Medida Respuesta	El sistema genera la notificación al usuario en menos de tres segundos

105

Análisis y Diseño de Servicios Escenarios de Calidad - Seguridad

Elemento	Descripción
Fuente	Cliente
Estimulo	El usuario solicita hacer un pago electrónico para los productos adquiridos
Artefacto	Tienda electrónica / Servicio de pagos del Banco
Ambiente	Condiciones normales
Respuesta	El pago electrónico es completado de forma segura y no puede ser repudiado por ninguna de las partes
Medida Respuesta	100% de las transacciones

106

¿Preguntas?



107