

## Fase de Análisis

Sebastián Ríos Pérez

Fuente: "UML y PATRONES", Craig Larman, 1Ed y 2 Ed.

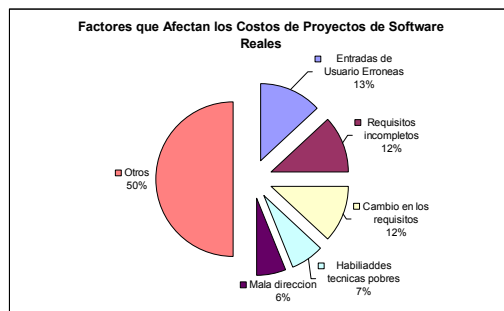
## Introducción

- La gran mayoría de aplicaciones se construyen para Terceras personas
- Por lo tanto, se requiere entender que es lo que ellos necesitan
- Es posible saber que es lo que necesitan los usuarios?
- Con que grado de Exactitud se requiere saber esta informacion?



2

## Introduccion (cont)



3

## Introduccion (cont)

- La tarea mas compleja es lograr **encontrar, comunicar y registrar** Todo lo que se necesita de modo que sea claro para el cliente y para el equipo de desarrollo

4

## Pasos a seguir en Análisis

- Algunas de las tareas a realizarse en la etapa de *análisis* son las siguientes:
1. Definir los requerimientos.
  2. Definir los casos esenciales de uso.
  3. Crear y perfeccionar Casos de uso
    1. Crear diagrama de casos de uso
  4. Definir los diagramas de secuencia de sistema (DSS)
  5. Crear los contratos del sistemas
  6. Crear y perfeccionar el modelo conceptual

5

## Definición de Requerimientos

## Los requerimientos

- son una descripción de las necesidades o deseos de los Clientes. La meta principal en esta etapa es identificar y documentar lo que en realidad se necesita, en una forma en que pueda fácilmente ser transmitido al cliente y al equipo de desarrollo. Se recomienda aquí definir al menos los siguientes puntos:

- Panorama general
- Metas
- Funciones del sistema
- Atributos del sistema

7

## Caso de estudio: el punto de venta

Supongamos como caso de estudio el sistema de una terminal de punto de venta. Esta terminal es un sistema automatizado con el que se registran las ventas y se realizan los pagos. Por lo general este tipo de sistemas comprenden hardware (un computador y un lector de código barras) y software (el sistema que se ejecuta en la terminal). Suponga que se nos ha contratado para crear este software

8

## Ejemplo:

### Panorama general

Este proyecto tiene por objeto crear un sistema de terminal para el punto de venta que se utilizará en las ventas al menudeo.

### Metas

En términos generales, la meta es una mayor automatización del pago en las cajas registradoras, y dar soporte a servicios más rápidos, más baratos y mejores. Más concretamente, la meta incluye:

- Pago rápido de los clientes.
- Análisis rápido y exacto de las ventas.
- Control automático del inventario.

### Funciones del sistema

Las funciones del sistema son lo que éste deberá de hacer. Hay que identificar estas funciones y listarlas en grupos lógicos. Para verificar que X es en verdad una función del sistema, la siguiente frase deberá tener sentido: "El sistema deberá hacer X". Por ejemplo: "el sistema deberá autorizar pagos a crédito".

Las funciones pueden clasificarse en tres categorías: evidentes, ocultas y superfluas. Las evidentes deben realizarse, y el usuario debe saber que se han realizado. Las ocultas también deben realizarse, y puede que no sean visibles para el usuario. Muchas de estas funciones se omiten (erróneamente) durante el proceso de obtención de requerimientos. Las superfluas son opcionales, y su inclusión no repercute significativamente en el costo ni en otras funciones.

9

## Clasificación de Requisitos

### ■ Modelo FURPS

- **Funcional:** características fundamentales, capacidades y seguridad
- **Facilidad de Uso (Usability):** facilidad de Uso, Ayuda y Documentación
- **Fiabilidad (Reliability):** Capacidad de recuperación ante un fallo y grado de previsión
- **Rendimiento (Performance):** tiempos de respuesta, disponibilidad, uso de recursos, etc
- **Soporte (Supportability):** facilidad de mantenimiento, Internacionalización, Configurabilidad

10

## Utilidad de FURPS

- Permite reducir los riesgos de no considerar alguna de las facetas del desarrollo de un sistema
- Permite estandarizar algunos criterios para poder obtener los requerimientos....

11

## Casos de Uso

## Definición de casos de uso

- Los casos de uso sirven para encontrar y capturar los requerimientos (especialmente los funcionales)
- Son descripciones narrativas en lenguaje natural de los procesos en un formato estructurado de prosa.
- Los casos de uso no son propiamente un elemento del análisis y diseño orientado a objetos (pueden ser utilizados en análisis no orientados a objetos).
- Tienen la gran virtud de mantener la información descrita de forma simple y comprensible por ambas partes

13

## Ejemplo: Jugar Binario

- Se tiene un juego de dados llamado "Binario" en que un jugador lanza 2 dados. Si el resultado suma 7 entonces Gana si no Pierde.
- **Jugar Binario:** *Este caso de uso comienza cuando el jugador recoge y tira 2 dados. Si los puntos suman siete, gana y pierde si suman cualquier otro número.*

14

## Definición de Actor

- Actor: Puede ser una Persona (identificada por su rol), otro sistema informático u Organización.
- Ejemplo
  - Si el software es un Video Juego, entonces el actor físico sería el niño, joven o adulto que juega el video juego. Sin embargo, desde el punto de vista del análisis del software el Actor es un **Jugador**.

15

## Actores y funciones

- Lo primero es reconocer a todos los actores que están involucrados en el sistema.
- Luego sus funciones básicas

Cajero	Registra los productos. Entrega el cambio
Cliente	Compra productos Paga los productos
Gerente	Inicia Cierra
Administrador del sistema	Incorpora nuevos usuarios

16

## Definición de Escenario

- Escenario: Es una secuencia específica de acciones e interacciones entre los actores y el sistema que se está estudiando.
- Muchas veces, se les denomina a los escenarios, instancias de Casos de Uso.
- Siempre hay al menos dos escenarios posibles.....

17

## Escenarios básicos

- **Escenario principal (de éxito):** El escenario principal, es aquella secuencia de acciones que permite concluir exitosamente con lo que se desea satisfacer en el caso de uso.
- Escenario de fallo: Es lo que ocurriría en caso de información mal ingresada, tipos de datos incorrectos, fallas en Hardware o Software, entre otras

18

## Ejemplo: Jugar Binario

- **Caso de uso: Jugar Binario**
- **Escenario principal de Éxito:**
  - *Este caso de uso comienza cuando el jugador recoge y tira 2 dados. Si los puntos suman siete, gana y pierde si suman cualquier otro número.*
- **Escenario de fallo:**
  - El jugador lanza los dados y al menos un dado no queda en posición totalmente horizontal sobre la superficie de juego, entonces debe tomar ambos dados y lanzarlos
  - En este caso podría definirse que solo lanzara 1 dado!!!!

19

## Ejemplo 2: PDV

- **Caso de Uso: Gestionar devoluciones**
- Escenario principal de Éxito:
  - Un Cliente llega a una Caja con artículos para devolver. El cajero usa el PDV para registrar cada uno de los artículos devueltos...
- Escenarios alternativos:
  - Si el cliente pagó con tarjeta de crédito y se rechaza la transacción de reembolso a su cuenta, entonces informar al cliente y pagarle en efectivo.
  - Si el Identificador del artículo no se encuentra en el sistema entonces, notificar al Cajero y sugerir la entrada manual del código de identificación....(que pasa si el código se salió del embase?)
  - Si el sistema detecta fallos con la comunicación con los sistemas de contabilidad externos.....

20

## Como detectar Casos de Uso

- Se debe focalizar la atención en buscar de que manera el sistema en cuestión logra entregar un resultado observable de valor para el usuario.
- Tratar de centrarse en la pregunta ¿cómo puedo, utilizando el sistema, proporcionar un valor observable para el usuario, o cumplir sus objetivos?

21

## Como detectar Casos de Uso

- Muchas veces un caso de uso puede utilizar varias funcionalidades del sistema, esta es una potencialidad importante que debe notar
  - Los casos de uso están pensados en los objetivos de los actores y no en las funciones
  - Por tanto las funciones pueden ser ordenadas de manera mas clara tanto para los Clientes como para el equipo de desarrollo

22

## Objetivos y Alcances de los Casos de Uso

- Lo típico es no saber si algo es un caso de uso o no
- Por lo tanto se tienen distintos niveles de granularidad
  - Desde el nivel más alto, actividades de la empresa
  - Hasta el nivel muy detallados
- Actividades atómicas es lo que NO se debe usar como CU
  - Ejemplo: "Eliminar el registro de la venta"
  - Mejor: "Procesar la Venta"

23

## Casos de Uso y Objetivos

- Los usuarios tienen objetivos (son sus necesidades)
- Por esto lo que se debe tratar de usar es los Casos de uso de Nivel de Usuario
  - Esto nos lleva a encontrar los Objetivos del Usuario
  - Definir un caso de uso para cada uno de estos Objetivos del Usuario
- Por lo tanto hay un cambio de enfoque:
  - Ya no se pregunta por ¿cuáles son los CU?
  - Se pregunta ¿cuáles son tus Objetivos?

24

## Ejemplo:

Analista: Me podría decir algunos de sus objetivos en el contexto del uso del PDV?

Cajero: Iniciar la sesión rápidamente y poder capturar las ventas

A : Cual es el objetivo más general que persigue usted al iniciar la sesión?

C: Intento identificarme en el sistema, de ese modo el sistema puede saber que estoy autorizado para capturar ventas

A: Pero, y eso para qué?

C: Para evitar los robos, la alteración de datos, y para no mostrar información privada de la compañía

25

## Objetivos de Subfunción

### ■ Por ejemplo:

- Identificarme y ser validado, se ha eliminado, sin embargo son objetivos validos para el usuario

### ■ Solo ocasionalmente deben escribirse estos como casos de uso

- Tipicamente si este objetivo se repite en otros casos de uso
- O si varios Actores requieren del mismo

- Por lo tanto de acuerdo a esto se puede colocar el caso de uso: Autenticar Usuario

26

## Caso de uso Importante

### ■ CRUEL

- Crear
- Recuperar
- Actualizar
- Eliminar

### ■ Defina estos como un solo caso de uso

- Gestionar XXX
- Administrar XXX

### ■ Ejemplo: Gestionar Usuarios, Administrar Pedidos

27

## Guia para trabajar con Casos de Uso

- Identificar a los Actores
- Identificar los Objetivos de los Actores
- Escribir los casos de Uso

28

## Definicion II de caso de uso

- Informalmente, un caso de uso es una colexión de escenarios el de éxito y un conjunto de fallo (o de flujo alternativo) relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo.

29

## Casos de Uso de Caja Negra

- La idea fundamental es pensar solo en el QUE y no en el COMO
- Se debe especificar las *responsabilidades* que debe tener el sistema de este modo logramos encontrar todos las funcionalidades del sistema

Jugar  
Binario

30

## Ejemplo: Caso de uso Registrar la Venta

- Estilo de Caja Negra
  - El sistema registra la venta
- Que es lo que NO se debe colocar
  - El sistema registra la venta en la Base de Datos
  - O peor.... Se genera una sentencia SQL, Insert...

31

## Tipos de Formalidad

- Dependiendo de la necesidad que se tenga, los casos de uso se escriben en diferentes formatos
  - Breve: resumen conciso, tratando de que sea un párrafo, por lo general solamente el escenario general de éxito
  - Informal: Formato de párrafo en estilo informal, múltiples párrafos y pueden contener varios escenarios
  - Completo: Se detalla por completo todos los pasos y variaciones.

32

## Formatos oficiales

- [www.usecases.org](http://www.usecases.org) Ahí se encuentran varios formatos disponibles para descargar y utilizar
- Desde los mas simples a los mas completos

33

## Plantilla Casos de Uso

- **Caso de uso: Nombre del Caso de Uso**
- Actor Principal:
- Personal Involucrado e intereses:
- Precondiciones:
- Garantías de éxito (Postcondiciones):
- Escenario principal de Éxito (Flujo básico):
- Extensiones (o Flujos Alternativos)
- Requisitos especiales:
- Lista de tecnología y variaciones de datos:
- Frecuencia:
- Temas pendientes:

34

## Ejemplo de plantilla estilo completo

- Caso de uso: Procesar venta
  - Actor Principal: Cajero
  - Personal Involucrado e intereses:
    - Cajero: quiere entradas precisas, rápidas y sin errores de pago, ya que las pérdidas se deducen de su salario
    - Vendedor: quiere que las comisiones de las ventas estén actualizadas
    - Compañía: quiere registrar la transacción con precisión y satisfacer los intereses de los clientes...
    - Servicio de Impuestos Internos: quiere recopilar los impuestos de cada venta...

.....

35

## Precondiciones

- Es lo que debe cumplirse siempre antes de comenzar cualquier escenario de un caso de uso
- Las precondiciones NO SE CHEQUEAN dentro del caso de uso, si no que se ASUMEN verdaderas
- Una Precondición, por lo general implica que se concluyo exitosamente un escenario dentro de otro caso de uso
- No es necesario colocar precondiciones obvias como: "El sistema debe tener electricidad"

36

## Postcondiciones o Garantías de éxito

- Establecen que es lo que debe cumplirse cuando se termina el escenario principal o bien algún escenario
- Debe satisfacer las necesidades de todos los involucrados

37

## Escenario principal de éxito

- “El camino Feliz” ☺
- Es la secuencia de acciones que debería cumplirse normalmente si todo sale bien
- Note que no lleva ninguna bifurcación condicional
- Todo el manejo de pasos alternativos se dejará para la definición de escenarios
- El tipo de interacciones que se recogen son
  - Interacción entre actores
  - Validaciones (Normalmente a cargo del sistema)
  - Un cambio de estado realizado por el sistema (Ej. Registrar o modificar algún dato)
- Sin embargo el primer paso es la ACCION que desencadena el Caso de Uso

38

## Extensión o Flujos alternativos

- Indican todos los posibles escenarios o bifurcaciones condicionales que pueden ocurrir en el sistema
- Una extensión tiene 2 partes:
  - Una condición
    - Es algo que puede ser detectado por el sistema o un actor
  - El manejo de la condición
    - Este se puede anotar como una secuencia de pasos o solo resumirlo en un paso

39

## Extensión o Flujos alternativos cont.

- Al final del manejo de la extensión, se vuelve al flujo principal a menos que se indique lo contrario en el manejo.
  - Ej. 1a. Se detecta un usuario no valido
    1. El sistema despliega un mensaje de error
    2. Se emite una alarma sonora
    3. El sistema se bloquea
- A veces un escenario es muy complejo, en este caso hay que decidir si se anota como un caso de uso a parte.
  - Ej. Por ejemplo en el caso del PDV el pago con tarjeta de crédito

40

## Requisitos especiales

- Cuando se tienen requisitos que no son funcionales pero que afectan fuertemente al caso de uso, entonces se recogen en el mismo caso de uso
- Por ejemplo:
  - Rendimiento, fiabilidad, facilidad de uso y otras restricciones de diseño (hardware necesario, etc)

41

## Lista de Tecnología y Variaciones de datos

- Muchas veces hay muchas maneras de hacer algo, el COMO. Sin embargo el qué esta dado.
- Aquí se colocan entonces las restricciones sobre el como se debe realizar un paso (Hardware)
- O diferentes formatos o variables que se pueden recoger dentro de algún paso del escenario

42

## Otra Plantilla de Casos de Uso

- **Caso de uso:** Nombre del caso de uso
- **Actores:** Lista de actores (agentes externos), en la cual se indica quién inicia el caso de uso
- **Propósito:** Intención del caso de uso
- **Resumen:** Repetición del caso de uso de alto nivel o alguna síntesis similar.
- **Tipo:** Primario, secundario u opcional. Esencial o real.
- **Referencias cruzadas:** Casos relacionados de uso y funciones también relacionadas del sistema.
- **Descripción:** Descripción del caso de uso.

43

## Ejemplo de caso de Uso alto nivel

- **Caso de uso:** Comprar productos
- **Actores:** Cliente, cajero
- **Tipo:** Primario
- **Descripción:** Un Cliente llega a la caja registradora con los artículos que va a comprar. El Cajero registra los artículos y cobra el importe. Al terminar la operación, el Cliente se marcha con los productos.

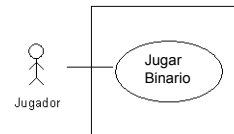
44

## Ejemplo de caso de Uso Breve

- **Caso de uso:** Comprar productos en efectivo
- **Actores:** Cliente (iniciador), Cajero
- **Propósito:** Capturar una venta y su pago en efectivo.
- **Resumen:** Un Cliente llega a la caja registradora con artículos que desea comprar. El Cajero registra los productos y recibe un pago en efectivo. Al terminar la operación, el Cliente se marcha con los productos comprados.
- **Tipo:** Primario y esencial.
- **Referencias cruzadas:** Funciones: R1.1, R1.2, R1.3, R1.7, R1.9, R2.1.
- **Descripción:** Un Cliente llega a la caja registradora con los artículos que va a comprar. El Cajero registra los artículos y cobra el importe. Al terminar la operación, el Cliente se marcha con los productos.

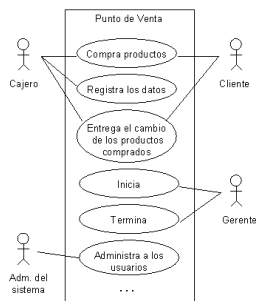
45

## Diagrama de CU básico



46

## Ejemplo: Diagrama de Casos de Uso



47

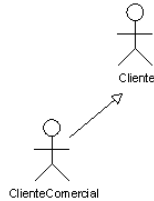
## Más sobre Actores

- Se pueden definir categorías generales de actores (como *cliente* en el ejemplo de abajo) y especializarlos (como *ClienteComercial*) a través de relaciones de generalización.

48



## Ejemplo: generalización Actores



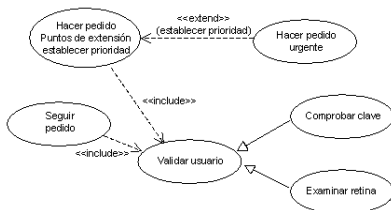
49

## Organización de casos de uso

- Los casos de uso pueden organizarse agrupándolos en paquetes.
- También se pueden especificar relaciones de generalización, inclusión y extensión.
- **Generalización** significa que el caso de uso hijo hereda el comportamiento y el significado del caso de uso padre, donde el hijo puede agregar o redefinir el comportamiento del padre. La generalización entre casos de uso se representa como una línea continua con una punta de flecha vacía.

50

## Ejemplo:



51

## Organización de casos de uso (2)

- Una relación de **inclusión** entre dos casos de uso significa que un caso de uso base incorpora explícitamente el comportamiento de otro caso de uso en el lugar especificado en el caso base.
- Aquí el caso de uso base toma el comportamiento del caso de uso proveedor. Esta relación se usa para evitar describir el mismo flujo de eventos repetidas veces, poniendo el comportamiento común en un caso de uso aparte (que será incluido por un caso base).
- Una relación de inclusión se representa como una dependencia, usando la palabra *include*. Para especificar la posición en un flujo de eventos, se usa la palabra *include* seguido del caso de uso que se quiere incluir.

52

## Ej. Include

- Por ejemplo, para describir el flujo de *Seguir pedido*:
- **Flujo de eventos principal:** Obtener y verificar el número de pedido. include (validar usuario). Examinar el estado de cada parte del pedido y preparar un informe para el usuario.

53

## Organización de casos de uso (3)

- Una relación de extensión entre casos de uso significa que un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por el caso de uso que extiende al base.
- Un caso de uso puede extenderse solamente en ciertos puntos, llamados *puntos de extensión*. La extensión se puede ver como que el caso de uso que extiende, incorpora su comportamiento en el caso de uso base. Se representa como una dependencia con la palabra *extend*.
- Los puntos de extensión sólo son etiquetas que pueden aparecer en el flujo del caso de uso base.

54

## Ej. Extend

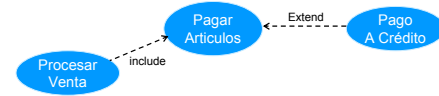
- Por ejemplo, el flujo de *Hacer pedido Urgente* podría escribirse así:

*Flujo de eventos principal:* incluye (Validar usuario). Recoger los ítems del pedido del usuario. (establecer prioridad). Enviar el pedido para ser procesado.

- Una relación de extensión se usa para modelar la parte de un caso de uso que el usuario puede ver como comportamiento opcional del sistema. De esta forma se separa el comportamiento opcional del obligatorio. Es decir, un caso de uso base puede, bajo ciertas condiciones, incorporar el comportamiento de otro caso de uso en el lugar especificado.

55

## Otro Ejemplo usando PDV



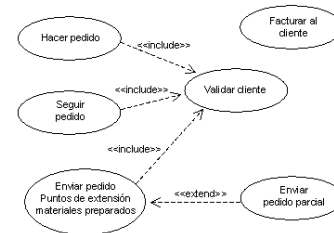
56

## Ejemplo: Sistema de ventas.

- Un sistema de ventas debe interactuar con clientes, los cuales efectúan pedidos. Además los clientes pueden hacer un seguimiento de sus propios pedidos. El sistema envía los pedidos y las facturas a los clientes. En algunos casos, según la urgencia de los clientes, se puede adelantar parte del pedido (pedidos parciales).

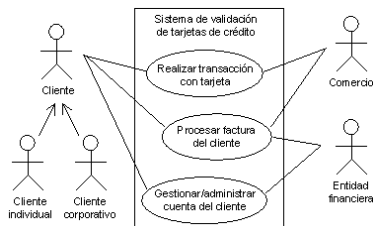
57

## Ejemplo: Sistema de ventas. (cont)



58

## Ejemplo 2: Validación de tarjetas de crédito



59

## Ejemplo 2: Validación de tarjetas de crédito

- La figura muestra el contexto de un sistema de validación de tarjetas de crédito. Se puede ver que existen dos categorías de clientes: clientes individuales, clientes corporativos
  - Ojo, hay un error en la notación UML de la figura, ¿cuál es?
- Estos actores representan los roles que juegan las personas que interactúan con el sistema
- También hay actores que representan otras instituciones, como Comercio, con el cual el cliente realiza una transacción para comprar un artículo o servicio, y Entidad financiera, que por lo general es una entidad bancaria donde el usuario tiene la tarjeta de crédito.

60

## Diagramas de Secuencia de Sistema (DSS)

Sebastián Ríos

Fuente: "UML y PATRONES", Craig Larman, 1Ed y 2 Ed.

### Objetivos

- Se pretende encontrar las interacciones entre los Actores y el Sistema en cuestión
- A nivel de analisis la idea es abordar los escenarios esenciales de éxito para poder aclarar varios puntos del sistema:
  - Complejidad
  - Tamaño
  - Coordinaciones, entre otras

62

### Modelamiento de Caja Negra

- Para poder cumplir el objetivo anterior y dado que aún no hemos diseñado nada, lo que conviene es utilizar el enfoque de Caja Negra.
- Nos enfocamos en "que" hace el sistema y no en "como" lo hace.

63

### DSS y Analisis

- Claramente en la fase de análisis, puede que aun nos falten escenarios, por eso debe focalizarse en el escenario principal de cada caso de uso.
- Con esta información se mejoran ciclicamente los casos de uso.

64

### DSS

- En un CU podemos ver como interactúan los actores externos con el sistema para obtener un resultado de valor observable.
- Un DSS es un dibujo que muestra los eventos que generan los actores externos, el orden y los eventos entre otros sistemas.
- Los DSS se realizan para un escenario específico

65

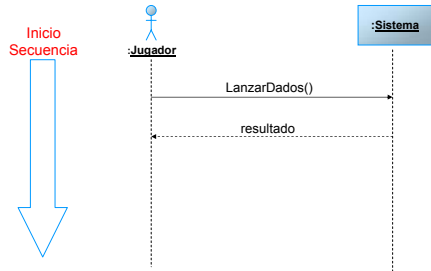
### Ejemplo para Jugar Binario

- **Caso de uso: Jugar Binario**
- Actores: Jugador
- Escenario principal de Éxito:
  1. Este caso de uso comienza cuando el jugador recoge y tira 2 dados.
  2. Si los puntos suman siete, gana y pierde si suman cualquier otro número.
- Escenario de fallo:
  - A. En cualquier momento el sistema falla en este caso, se debe.
    1. Permitir reiniciar el juego, (volver al paso 1)

66

## DSS para caso de uso “Jugar Binario”

### ■ Escenario Principal de Éxito.



67

## Ciclos en los eventos

- Muchas veces hay un conjunto de eventos que se desea repetir, hasta que una cierta condición se cumpla
- Para representar esta situación, se utiliza un rectángulo sobre las acciones a repetir y un parentesis cuadrado, especificando la condición

68

## Ejemplo modificado para Jugar Binario

### ■ Caso de uso: Jugar Binario

#### ■ Actores: Jugador

#### ■ Escenario principal de Éxito:

1. Este caso de uso comienza cuando el jugador recoge y tira 2 dados.
2. El sistema despliega el resultado. Si los puntos suman siete, gana y pierde si suman cualquier otro número.
  - Se repiten los pasos 1 y 2 si el usuario lo desea y si ha ganado el juego actual
3. Se finaliza el juego

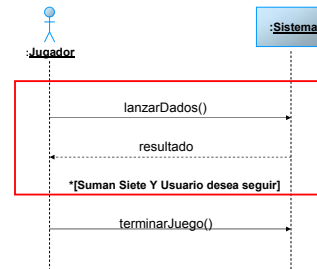
#### ■ Escenario de fallo:

- A. En cualquier momento el sistema falla en este caso, se debe.
  1. Permitir reiniciar el juego, (volver al paso 1)

69

## DSS modificado para C.U. “Jugar Binario”

### ■ Escenario Principal de Éxito.



70

## Eventos y limites del Sistema

- Estos diagramas deben representar todos los eventos entre los actores y el sistema
- Para esto es muy importante tener claros cuales son los limites del sistema
  - Usualmente como limite el sistema y el hardware, si es que es el caso
  - La mala elección del limite puede llevarnos a no considerar algunos eventos o a ciniderar mas de los que son necesarios.
- Para evitar esto debemos indentificar muy bien a los actores externos.
  - Estos son los que interactúan directamente con el sistema enviando eventos a este.

71

## Ejemplo: Limites de sistema

### ■ Caso de uso: Procesar venta

#### □ Actor Principal: Cajero

#### □ Personal Involucrado e intereses:

- Cajero: quiere entradas precisas, rápidas y sin errores de pago, ya que las perdidas se deducen de su salario
- Vendedor: quiere que las comisiones de lqas ventas esten acytualizadas
- Compañía: quiere registrar la transacción con precisión y satisfacer los intereses de los clientes...
- Servicio de Impuestos Internos: quiere recopilar los impuestos de cada venta...

.....

72

## Ejemplo: Limites de sistema (cont)

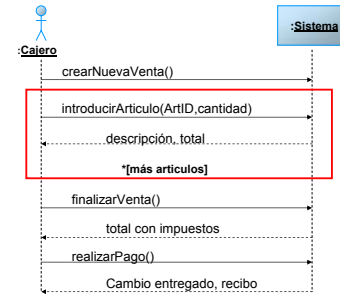
Escenario principal de Éxito (Flujo básico):

- 1.El cliente llega a un terminal de PDV con artículos y/o servicios que desea comprar
- 2.El cajero inicia una nueva vena
- 3.El cajero introduce el identificador del artículo
- 4.El sistema registra la línea de la venta y presenta la descripción del artículo, precio y suma parcial. El precio se calcula de acuerdo a un conjunto de reglas de precios
  - El cajero repite los pasos 3 y 4 hasta que se indique
- 5.El sistema presenta el total calculado
- 6.El cajero le dice al cliente el total de la venta y le pide que le pague
- 7.El cliente paga y el sistema gestiona el pago
- 8.El sistema registra la venta completa y envía información de la venta y del pago al sistema de contabilidad externo (Para la contabilidad y las comisiones) y al sistema de inventario (Para actualizar el inventario)
- 9.El sistema imprime el recibo
- 10.El cliente se va con el recibo y los artículos (si es el caso)

75

## DSS de Pocesar Venta

### Escenario principal de Éxito



74

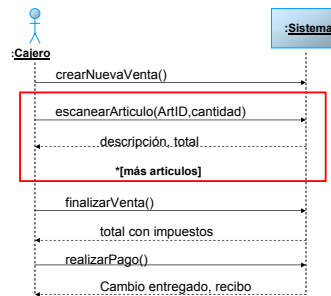
## Como asignar nombres a los eventos y operaciones

- Los nombres de los eventos y sus operaciones asociadas, deben estar nombradas de acuerdo con las motivaciones que dan lugar a ellas
  - Por lo general un verbo
  - introducirArticulo, cambiarNombre, realizarPago...
- NO USE!!
  - Nombres que esten relacionados con los medios de entrada físicos.
  - O con la Interfaz de Usuario

75

## Ejemplo: ¿Cuál es el Error?

### Escenario principal de Éxito



Escanear, es un nombre que Esta ligado al hardware con el que se esta pensando realizar esa labor.

- Se pierde Generalidad
- Se limita el diseño en una fase que no corresponde
- IntroducirArticulo, deja ver cual Es la intención del cajero

76

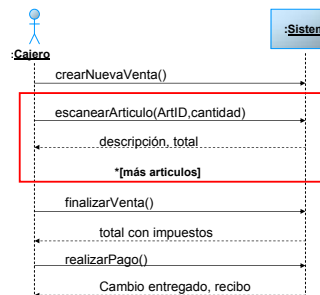
## Observación/Consejo práctico

- Muchas veces es muy bueno, colocar el texto del escenario de éxito junto al diagrama para facilitar la lectura.

77

## Ejemplo:

### Escenario principal de Éxito



Escenario principal de Éxito (Flujo básico):

- 1.El cliente llega a un terminal de PDV con artículos y/o servicios que desea comprar
- 2.El cajero inicia una nueva vena
- 3.El cajero introduce el identificador del artículo
- 4.El sistema registra la línea de la venta y presenta la descripción del artículo, precio y suma parcial. El precio se calcula de acuerdo a un conjunto de reglas de precios
- El cajero repite los pasos 3 y 4 hasta que se indique

78

## DSS y los Glosarios

- Por lo general los eventos, operaciones, parámetros y valores de retorno en un DSS son concisos. Por lo que podría requerirse de una explicación más detallada
  - Para esto utilizamos el Glosario
- Sin embargo, esto debería haber quedado ya definido y explicado en los casos de uso. Si no, podrían agregarse a un Glosario.
  - Sin embargo, piense si será de real utilidad hacer el glosario, pues a veces puede ser trabajo desperdiciado.

79

## Modelo del Dominio

Sebastián Ríos

Fuente: "UML y PATRONES", Craig Larman, 1Ed y 2 Ed.

## Modelo de Dominio

- Este es un modelo muy simple pero importante. Tiene por objetivo mostrar a los modeladores las *clases conceptuales* significativas en el *dominio de un problema*.
- Es el diagrama más importante creado durante el AOO.
  - Ojo, Casos de Uso, corresponden al análisis de requisitos y nos son OO.
  - Sin embargo ponen de relieve el punto de vista de del proceso de dominio

81

## Modelo del Dominio (Cont)

- No muestra componentes de software
- No muestra clases de software
- Ni objetos de software
- Lo que se buscan son las clases de conceptos del mundo real dentro del dominio del problema que son importantes tener presentes.

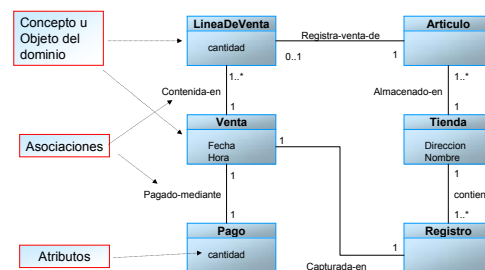
82

## Modelo del Dominio (Cont)

- Se utiliza notación UML para poder representarlo
- Se diagrama como un *diagrama de clases*, pero sin ninguna operación
- Pueden mostrar:
  - Clases conceptuales u objetos del dominio
  - Asociaciones entre estas clases conceptuales
  - Atributos de las clases conceptuales

83

## Ejemplo: Para las ventas del PDV



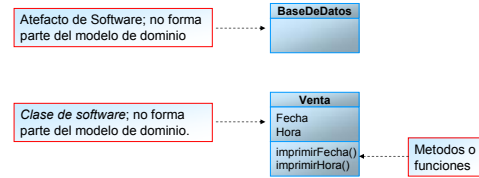
84

## Observaciones importante

- El modelo de dominio no es un modelo de componentes de software
- Representa cosas del mundo real, por lo que debe evitar componentes u objetos de software

85

## Ejemplo: ¿qué debe evitar?



86

## Descomposición del Dominio

- Existen varias formas de descomponer el dominio de un problema, para reducir su complejidad.
  - Descomposición funcional, donde el problema se descompone por funciones o procesos
  - Análisis y diseño orientado a objetos, el foco está en descomponer por COSAS o ENTIDADES, del dominio del problema
- Por lo tanto, lo más importante es lograr identificar todos los conceptos y documentarlos en un diagrama de dominio.

87

## Ejemplo: Dominio de Ventas

- Para las ventas de una tienda del mundo real, existen varias clases conceptuales o conceptos
  - Tienda
  - Registro
  - Pago
  - Venta
  - Cajero
  - Artículo..... Etc..

88

## Identificando clases conceptuales

- Se deben identificar las clases conceptuales por escenario de los Casos de Uso
- Hay varias estrategias para lograrlo:
  1. Utilización de listas de Categorías de clases conceptuales
  2. Identificación de frases nominales
  3. Utilización de patrones de análisis

89

## Lista de categoría de clases conceptuales

Categoría de clase conceptual	Ejemplos
Objetos tangibles o físicos	Registro, Avión
Especificaciones, diseños o descripciones de las cosas	EspecificaciónDelProducto, DescripciónDelVuelo
Lugares	Tienda, Oficina, Bodega
Transacciones	Venta, Pago, Reserva
Líneas de Transacción	LíneaDeVenta
Roles de la gente	Cajero, Piloto, Conductor
Contenedores de otras cosas	Tienda, Lata, Avión, estante
Cosas en un contenedor	Artículo, Pasajero, libro
Otros sistemas informáticos o eléctricos externos	SistemaAutorizacionPagoCredito, SII, ControlDeTráficoAereo
Conceptos abstractos	Ansia, Acrofobia
Organizaciones	DepartamentoDeVentas, CompañíaAerea
Hechos	Venta, Pago, Reunión, vuelo, colisión, aterrizaje
Procesos (normalmente no se presentan como conceptos)	VentaDeProductos, ReservaDeAsiento
Reglas y políticas	PolíticaDeReingreso, PolíticaDeCobranza
Catálogos	CatálogoDeProductos, CatálogoDePiezas
Registros de Finanzas, trabajo, contratos, cuestiones legales	LíneaDeCrédito, Stock
Manuales, documentos, artículos de referencia, libros	ListaDeCambiosDePreciosDiarios, ManualReparaciones

90

## Frases nominales

- La idea fundamental es tratar de identificar, mediante el análisis lingüístico, nombres y frases nominales, en las descripciones textuales de un dominio. Y considerarlas clases conceptuales
  - Frase nominal, es aquella que no tiene verbo
- Problemas:
  - No se puede realizar una correspondencia mecánica entre estas frases y los nombres de las clases
  - Muchas veces las frases son ambiguas

91

## Ejemplo: PDV

### Escenario principal de Éxito (Flujo básico):

1. El **cliente** llega a un **terminal de PDV** con **artículos y/o servicios** que desea comprar
2. El **cajero** inicia una nueva **venta**
3. El **cajero** introduce el **identificador del artículo**
4. El sistema registra la **línea de la venta** y presenta la **descripción del artículo, precio y suma** parcial. El precio se calcula de acuerdo a un conjunto de **reglas de precios**
  - El cajero repite los pasos 3 y 4 hasta que se indique
5. El sistema presenta el total calculado
6. El cajero le dice al cliente el total de la **venta** y le pide que le pague
7. El cliente paga y el sistema gestiona el **pago**
8. El sistema registra la venta completa y envía información de la venta y del pago al sistema de **contabilidad** externo (Para la contabilidad y las **comisiones**) y al sistema de **inventario** (Para actualizar el inventario)
9. El sistema imprime el **recibo**
10. El cliente se va con el recibo y los artículos (si es el caso)

92

## Patrones de Análisis

- Son modelos de Dominio parciales y que han sido desarrollados por expertos.
- Hay varios libros publicados:
  - Analysis Patterns
  - Data Model Patterns

93

## Clases conceptuales candidatas

- Después de aplicar alguno de los métodos anteriores, es posible generar una lista con clases conceptuales candidatas
- Para el dominio de Ventas
  - Registro
  - Artículo
  - Tienda
  - Venta
  - Pago
  - CatalogoDeProductos
  - EspecificacionDelProducto
  - LineaDeVenta
  - Cajero
  - Cliente
  - Encargado

94

## Un detalle importante

- Y que paso con el recibo??, no esta en el modelo!
  - Un recibo es un informe de venta, y toda la información que contiene esta contenida en otras fuentes. Por lo tanto al incluirlo estaríamos duplicando información
  - Sin embargo, el recibo le da al portador la facultad de poder devolver los artículos, por lo tanto, al momento de modelar el caso de uso "Gestionar Devoluciones" lo colocaremos justificadamente

95

## Como Nombrar y Modelar mejor

- Estrategia del Cartógrafo
  - Haga el modelo del dominio con el espíritu de trabajo de los cartógrafos
    - Utilice los nombres existentes en el Territorio (dominio)
    - Excluya las características irrelevantes. (El cartógrafo no coloca la población o la topografía.... No coloque conceptos que no son relevantes)
    - No añada cosas que no están ahí

96

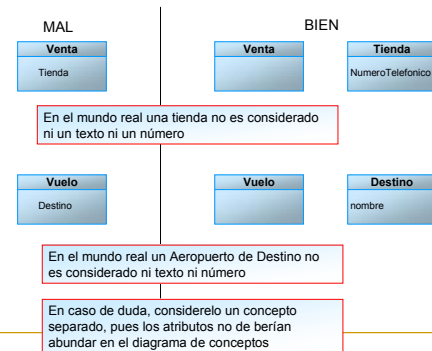


## Típico Error

- Es cuando representamos algo como un atributo cuando en verdad debería ser un concepto
- Para evitar esto hay una sencilla regla
  - Si hay una clase conceptual X, que no se considera Número o Texto, entonces probablemente es una clase conceptual y no un atributo

97

## Ejemplo



98

## Clases conceptuales similares.

- En algunas ocasiones nos encontramos con 2 clases conceptuales que desempeñan la misma función pero con nombre distinto.
- Por ejemplo:

En este caso lo que normalmente se hace es utilizar el nombre del sistema. En este caso se considera, al sistema, más bien como una entidad abstracta que registra, lee, etc.



99

## Modelos del mundo Irreal

- Algunos dominios de sistemas, poseen dominios muy abstractos. Que tienen muy poca analogía con dominios naturales o de negocios.
- Por ejemplo algunos sistemas de telecomunicaciones poseen como clases conceptuales
  - Mensaje, Conexión, Puerto, Dialogo, Ruta, Protocolo, entre otros

100

## Clases conceptuales de Especificación o descripción

- Suponga lo siguiente:
  - Una instancia de un **Artículo** representa una instancia física dentro de la tienda, y por lo mismo cada artículo podría tener un número de serie
  - Un **Artículo** tiene un precio, descripción e identificador del artículo que no se recogen en otro sitio.
  - Todos en la tienda sufren de amnesia
  - Cada vez que se vende un artículo físico real, se elimina una instancia de **Artículo**

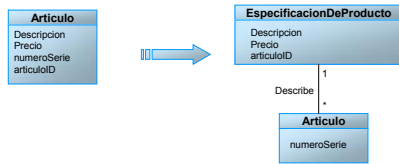
101

## Clases conceptuales de Especificación o descripción (cont)

- Entonces bajo estos supuestos si el Artículo fuera lapiz de pasta. Y la tienda vende todos los lapices de pasta.
- No queda ningún "Lapiz de pasta" en la memoria del Computador
- ¿cuánto cuestan los lapices de pasta?.....nadie puede responder!!!
  - El precio estaba en las instancias del lapiz de pasta.
- Por lo tanto se requieren Clases que Describan o Especificquen a otras clases conceptuales.

102

## Ejemplo: Clases de Especificación



103

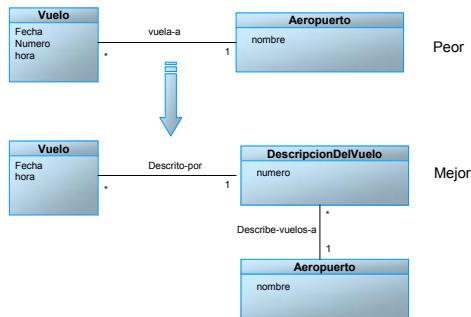
## Cuando agregar clases conceptuales de Especificación

### ■ Cuando:

- Se necesita de la descripción o especificación de un articulo o producto independiente de la existencia de un articulo en particular
- La eliminación de instancias de las cosas que se describen (Articulo) dan como resultado una perdida de información que necesita mantenerse.
- Desea reducir información redundante o duplicada.

104

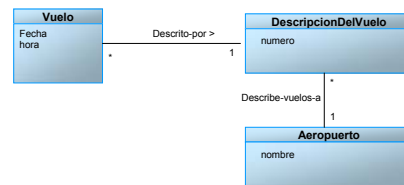
## Ejemplo:



105

## Asociaciones y Notación

- Una Asociación es una relación entre tipos (o más concretamente, son instancias de estos tipos)
- Se representan por una línea, puede tener dirección y cardinalidad



106

## ¿cuándo colocar una relación?

- Considere las siguientes:
  - Asociaciones de las que es necesario conservar el conocimiento de la relación durante algún tiempo.
    - Por ejemplo, la asociación entre LineaDeVenta y Venta es necesaria, sin ella no se podría reconstruir la venta.
  - Asociaciones derivadas de la ListaDeAsociaciones

107

## Lista de Asociaciones Comunes

Categoría	Ejemplos
A es una parte física de B	Cajón - Registro, Ala - avión
A es parte lógica de B	LineaDeVenta - Venta, RutaVuelo - Vuelo
A esta contenido físicamente en B	Registro - Tienda, Artículo - Estantería
A esta contenido lógicamente en B	DescripciónDelArtículo - Catálogo
A es una descripción de B	DescripciónDeArtículo - Artículo
A es una línea de transacción o un informe de B	TrabajoMantenimiento - RegistroDeMantenimiento
A se conoce/registra/recoge/informa/captura en B	Venta - RegistroVenta, Reserva - ListaDePasajeros
A es miembro de B	Cajero - Tienda, Artículos - Bodega
A es sub-unidad organizativa de B	Departamento - Tienda, Mantenimiento - CODELCO
A utiliza o gestiona B	Cajero - RegistroVenta, Piloto - Avión
A se comunica con B	Cliente - Cajero, AgenteDeReservas - Pasajero
A esta relacionado con una transacción B	Cliente - Pago, Pasajero - Pasaje
A es una transacción relacionada con otra transacción B	Pago - Venta, Reserva - Cancelación
A está al lado de B	Ciudad - Ciudad
A es propiedad de B	RegistroDeVenta - Tienda, Avión - CompañíaAerea
A es un evento relacionado con B	Venta - Cliente, Venta - Tienda

108

## Asociaciones de alta prioridad

- Son asociaciones que es invariante útil incluirlas en el modelo del dominio.
  - A es una parte lógica o física de B
  - A esta contenida física o lógicamente en B
  - A se registra en B

109

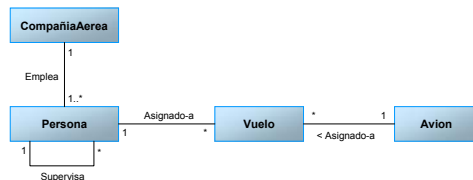
## Guía para las asociaciones

- Centrese en las asociaciones para las que se necesita conservar el conocimiento de la relación durante algún tiempo
- Es más importante identificar clases conceptuales que asociaciones
- Demasiadas asociaciones tienden a confundir un modelo de dominio
- Evite mostrar relaciones redundante o derivadas

110

## Roles de las Asociaciones

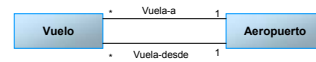
- Por lo general se respeta el formato:
  - NombreTipo – FraseVerbal – NombreTipo
  - Se leen típicamente de derecha a izquierda
  - Pueden tener direccionalidad



111

## Múltiples Asociaciones entre 2 Tipos

- A veces pueden existir más de una relación entre 2 clases conceptuales. Cuando es necesario anotarlas.
  - Cuando la naturaleza de las relaciones sea distinta
  - Cuando valga la pena hacer la distinción



112

## Asociaciones e Implementación

- Las asociaciones en el modelo de dominio son solo semánticas no es una declaración del flujo de los datos, conexiones entre objetos o instancias de clases. Solo es una declaración de que aquella relación es importante en el mundo real.
- Sin embargo, muchas de ellas serán implementadas posteriormente.

113

## Los Atributos

- Los atributos, son datos relevantes que es importante tener presentes con respecto a una clase conceptual
- Estos deberían ser simples o de tipo de datos
  - Boolean, Fechas, Horas, Números, Textos
  - Ejemplos: Dirección, Color, Teléfono, UPC (Universal Product Code)

114

## ¿Qué sucede con los atributos en el código?

- La restricción de que los datos sean simples, no implica que los atributos en C++ o Java, solo deban ser de datos primitivos o simples.
- Es más, durante la fase de diseño, muchas de las Asociaciones del modelo del dominio, se implementarán como atributos que referencian a otros objetos de software complejos.

115

## No a las Llaves Foraneas

- Ojo, este es un modelo del dominio, se están registrando en el modelo las clases conceptuales
- Por lo tanto no debe NUNCA relacionar una clase via un atributo de otra clase conceptual
  - Nuevamente....Esto no es una Base de Datos!!!!

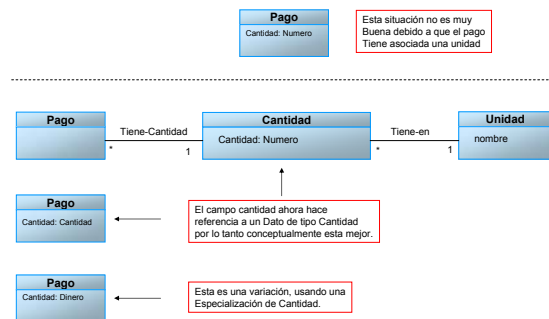
116

## Modelado de Cantidades

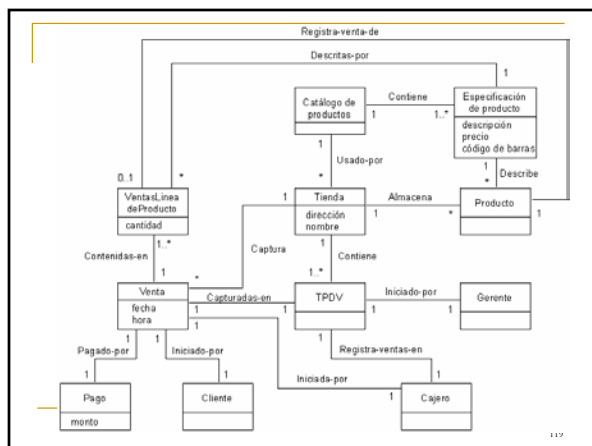
- Las cantidades numéricas en general no son solo números, si no que también tienen asociadas una unidad
  - Velocidad, Precio, Cantidad de Material
- Por lo tanto muchas veces es bueno modelar algunas de estas cantidades como Clase Conceptual no como un atributo.

117

## Ejemplo



118



119

## Contratos de Operaciones

## Introducción

- Los contratos de operaciones sirven para ayudar a completar y detallar los casos de uso
- Para lo cual se describe el comportamiento detallado del sistema en función de los cambios de estado de los objetos del modelo de Dominio después de la ejecución de una operación

121

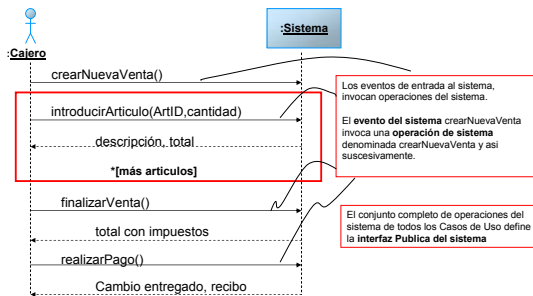
## ¿Como se generan los contratos?

- Se genera un contrato por cada operación del sistema (como caja negra)
- Se utiliza el los DSS y en base a estos, para cada uno de los mensajes se debe generar una ficha que se denomina contrato.

122

## Ej: DSS de Pocosar Venta

- Escenario principal de Exito



123

## Secciones del Contrato

- **Operación:** Nombre de la operación y parámetros
- **Referencias Cruzadas:** (Opcional) Algún caso de uso que pueda hacer uso de esta operación
- **Precondiciones:** Suposiciones relevantes sobre el estado del sistema o de los objetos del Modelo de Dominio, antes de la ejecución de la Operación. Estas condiciones no se comprobaran en la lógica de esta operación, se asumen ciertas, y son suposiciones no triviales que deben notarse que se han realizado
- **Postcondiciones:** El estado de los objetos del Modelo de Dominio después que se completa la operación. (escriba las postcondiciones en pasado)

124

## El escenario y el telón

- Suponga que el Sistema y sus clases conceptuales se presentan en un escenario
- Al levantarse el telón usted le toma una foto
- Luego el telón se baja..... Y se aplica la operación al sistema!!!!
- Al levantar el telón saca otra foto....
- Compare las fotos y exprese como postcondiciones las diferencias que existen en las fotografías. ☺

125

## Postcondiciones

- Las post condiciones deben hacer claros los siguientes cambios de estado en los objetos del dominio:
  - Creación de instancias
  - Creación o Disolución de asociaciones
  - Cambio en los atributos
- Recordar que “se quiere reflejar el estado de los objetos del modelo de dominio al termino de la ejecución de una operación!!!”

126

### Relación con entre postcondiciones y Modelo de Dominio

- Que relaciones pueden crearse o destruirse?
- Que instancias pueden de objetos pueden crearse o destruirse?
- “Las que usted especifico en el Modelo de Dominio, trate de ser consistente”

127

### Ejemplo de Postcondiciones

- Operación: crearLineaDeVenta

.....  
.....

- Postcondiciones:

- Se creo una instancia de LineaDeVenta **ldv**
- **ldv** se asocio con la venta actual (nueva asoci)
- **ldv.cantidad** paso a ser cantidad (modif. De atrib)
- **ldv** se asoció con una EspecificaciónDelProducto, en base a la coincidencia del articuloID (nueva asoci).

128

### Ventaja de las postcondiciones

- Nuevamente nos enfocamos en que hace el sistema en vez de en como lo realizará, que es lo que nos importa en esta fase.
- Además, este análisis soporta el poder llegar al detalle fino. Especificando cual debe ser el resultado de la operación.

129

### Actualizaciones al Modelo de Dominio

- Al generar los contratos de Operaciones usualmente puede descubrir nuevas cosas:
  - Registrar nuevas clases conceptuales
  - Atributos o asociaciones
- Por lo tanto no se limite a la definición inicial del Modelo de Dominio, realice las modificaciones que sean necesarias

130

### ¿Cuándo son útiles los Casos de Uso?

- Se supone que debería ser suficiente solo con los casos de uso
- Sin embargo, algunas veces las complejidad de algunas operaciones
  - añadirNuevaReserva(), en un sistema de vuelo
- Por lo tanto puede generar el Contrato para aportar mayor claridad al equipo de desarrollo sobre esta operación

131

### ¿Cuándo no son útiles los contratos?

- Dejan de ser útiles si se hacen contratos para todas las operaciones del sistema
  - Esto implica que el modelo de casos de uso (el documento de CU) es DEFICIENTE y por lo tanto el esfuerzo debería estar centrado en los Casos de Uso
- A veces puede en lugar de aclarar el modelo de casos de uso lo complica más con información que no aporta valor

132

## Guía para la construcción de Contratos

1. Identifique las operaciones del sistema utilizando el DSS
2. Construya contratos para las operaciones complejas o que no están claras en los casos de uso
3. Para describir las postcondiciones utilice las siguientes categorías
  1. Creación y eliminación de instancias
  2. Modificación de Atributos
  3. Formación y rotura de asociaciones

133

## Ejemplo de contrato para el PDV

Operación: realizarPago(cantidad: Dinero)

Referencias Cruzadas: CU Procesar Venta

Precondiciones: Hay una venta en curso

Postcondiciones:

- Se creó una instancia de pago p (creación de instancias)
- P.cantidadEntregada pasó a ser cantidad (modificación de atributos)
- P se asoció con la Venta actual (formación de asociaciones)
- La venta actual se asoció con la Tienda (formación de asociaciones)

134