

Análisis Orientado a Objetos

Asociaciones: Identificación de casos comunes [1]

- ☞ Comience por agregar las asociaciones utilizando la lista de anexa.
 - Categorías comunes que normalmente vale la pena incluir.

Categoría	Ejemplos
A es una parte física de B	Caja-TPDV Ala-Avió
A es una parte lógica de B	VentasLineaDeProducto-Vent TramoDeVuelo-RutaDeVuelo
A está físicamente contenido en B	TPDV-Tienda Producto-Estante Pasajero - Avió
A está lógicamente contenido en B	DescripcionDeProducto - Catálogo Vuelo ProgramaDeVuelo

Análisis Orientado a Objetos

Asociaciones: Identificación de casos comunes [4]

- ☞ Las categorías de alta prioridad que siempre vale la pena incluir son:
 - A es una *parte física o lógica* de B
 - A está *físicamente o lógicamente contenido* en B
 - A está *registrado* en B
- ☞ Las asociaciones son importantes, pero un error común es dedicar demasiado tiempo a descubrirlas.
 - *Es mucho más importante identificar conceptos que asociaciones.* El tiempo asignado a crear el modelo conceptual debería destinarse a identificar conceptos, no asociaciones.
 - Demasiadas asociaciones tienden a confundir el modelo conceptual en vez de aclararlo. A veces se requiere mucho tiempo para descubrirlas, y los beneficios son escasos.

Análisis Orientado a Objetos

Asociaciones: Identificación de casos comunes [2]

A es una descripción de B	DescripcionDeProducto - Product DescripcionDeVuelo - Vuelo
A es un elemento de línea en un transacción o reporte B	VentasLineaDeProducto-Vent TrabajoDeMantenimiento-Mantenimiento
A se conoce/ introduce/ registra/ presenta/ captura en B	Venta - TPDV Reservacion - ListaDePasajeros
A es miembro de B	Cajero - Tienda Piloto - Avio
A es una subunidad organizacional de B	Departamento - Tienda Mantenimiento - LineaAerea
A usa o dirige a B	Cajero - TPDV Piloto - Avio
A se comunica con B	Cliente - Cajero AgenteDeReservaciones - Pasajero

Análisis Orientado a Objetos

Asociaciones: Directrices

- ☞ Concentrarse en las asociaciones en que el conocimiento de la relación ha de preservarse durante algún tiempo (asociaciones que "es necesario conocer").
- ☞ No incluir las asociaciones redundantes, ni las derivables.
 - Redundantes: repiten lo dicho por otras asociaciones.
 - Derivables: pueden ser deducidas a partir de otras asociaciones.

Análisis Orientado a Objetos

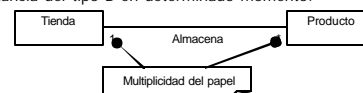
Asociaciones: Identificación de casos comunes [3]

A se relaciona con una transacción B	Pago - Boleto Pasajero - Boleto
A es una transacción relacionada con otra transacción B	Pago - Venta Reservacion - Cancelacion
A está contiguo a B	TPDV - TPDV Ciudad - Ciudad
A es propiedad de B	TPDV - Tienda Avion - LineaAerea

Análisis Orientado a Objetos

Asociaciones: Multiplicidad

- ☞ La *multiplicidad* define cuántas instancias de tipo A pueden asociarse a una instancia del tipo B en determinado momento.



- ☞ Algunos ejemplos de multiplicidad:

*	cero o más, "muchos"	1..40	de uno a 40
1..*	uno o más	5	exactamente 5
3,5,8	exactamente 3, 5 u 8		

- ☞ En UML, el valor de multiplicidad *depende del contexto*.
 - P.ej., la multiplicidad de Trabaja-Para entre Persona y Compañía será diferente si se está haciendo el sistema al SII ("..") o una empresa (1..*).

Análisis Orientado a Objetos

Asociaciones: Notación

- ⌘ Se asigna un nombre a una asociación basándose en el esquema *NombreDeTipo – FraseNominal – NombreDeTipo*.
 - La idea es genera una secuencia (con al frase nominal) que sea legible y significativa dentro del contexto del modelo.
 - Los nombres de las asociaciones comienzan con mayúscula.
 - Una frase nominal (verbo) debe construirse con guiones.
 - La dirección de lectura es de izquierda a derecha y de arriba hacia abajo.

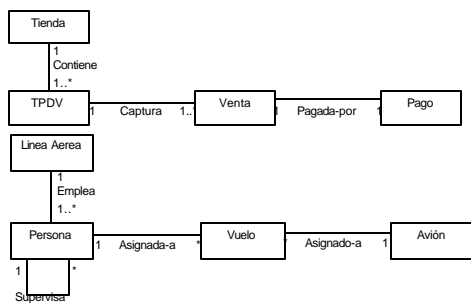
Análisis Orientado a Objetos

Asociaciones: Implementación

- ⌘ Durante la fase de análisis, una asociación no es una proposición sobre flujos de datos, variables de instancia, ni conexiones de objetos en una solución de software; *es una proposición de que una relación es significativa en el mundo real.*
- ⌘ Una asociación no necesariamente debe ser implementada durante la construcción.
 - Es posible que una asociación sea reemplazada por una clase al e laborar el modelo de software.

Análisis Orientado a Objetos

Asociaciones: Ejemplo



Análisis Orientado a Objetos

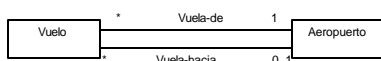
Ej. punto de venta: Asociaciones del dominio

- ⌘ Deberíamos incorporar las asociaciones que indican los requerimientos (los casos de uso, por ejemplo), las que conllevan la necesidad de recordar o que de alguna otra forma nos sugiere nuestra percepción del dominio del problema.
- ⌘ Conceptos:
 - TPDV, Producto, Tienda, Venta, Pago
 - Catalogo De Producto, Especificación De Producto
 - Ventas Linea De Productos
 - Cajero, Cliente, Gerente

Análisis Orientado a Objetos

Asociaciones múltiples entre 2 conceptos

- ⌘ Dos conceptos pueden tener varias asociaciones entre ellos.
 - P.ej., en el dominio de la línea aérea encontramos varias relaciones entre Vuelo y Aeropuerto.
 - Las asociaciones volar-hacia y volar-de son netamente diferentes que deben mostrarse por separado.
 - Nota: no se garantiza que todos los vuelos aterricen en un aeropuerto.



Análisis Orientado a Objetos

Ej. punto de venta: Asociaciones del dominio

- ⌘ Relaciones inolvidables en la Tienda

<i>TPDV captura Venta</i>	Para conocer la venta actual genera un total e imprime un recibo.
<i>Venta pagada en efectivo</i>	Para saber si se pagó la venta, relaciona la cantidad ofrecida con el total de la venta e imprime un recibo.
<i>CatalogoDeProductos registra EspecificacionDeProducto</i>	Para recuperar la especificación de producto con un código universal de producto

Análisis Orientado a Objetos Ej. punto de venta: Asociaciones del dominio

☞ Recorreremos la lista de comprobación, basándonos en tipos anteriormente identificados y teniendo presentes los requerimientos actuales del caso de uso.

Categoría	Sistema TPDV
A es una parte física de B	no se aplica
A es una parte lógica de B	VentasLineaDeProducto-Venta
A está contenido físicamente en B	TPDV-Tienda Producto-Tienda
A está contenido lógicamente en B	EspecificacionDeProducto - CatalogoDeProductos CatalogoDeProductos - Tienda
A es una descripción de B	EspecificacionDeProducto - Producto
A es un elemento de línea en una transacción o reporte B	VentasLineaDeProducto-Venta

Análisis Orientado a Objetos Ej. punto de venta: Modelo Conceptual

☞ El conjunto de asociaciones que se incluye en el modelo se obtuvo de manera bastante mecánica a partir de la lista de comprobación. Pero tal vez hay que ser más restrictivos con las asociaciones.

☞ Venta Capturada por Cajero

- Los requerimientos no indican la necesidad de conocer, ni de registrar al cajero actual. Además es derivable si existe la asociación TPDV usado-por Cajero.

☞ TPDV Usado-por Cajero

- Los requerimientos no indican la necesidad de conocer, ni de registrar al cajero actual.

Análisis Orientado a Objetos Ej. punto de venta: Asociaciones del dominio

A se conoce/ introduce/ registra/ presenta/ captura en B	Venta (terminada) - Tienda Venta (actual) - TPDV
A es miembro de B	Cajero - Tienda
A es una subunidad organizacional de B	no aplicable
A usa o dirige a B	Cajero - TPDV Gerente - TPDV Gerente - Cajero, probablemente no aplicable
A se comunica con B	Ciente - Cajero
A se relaciona con una transacción B	Ciente - Pago Cajero - Pago
A es una transacción relacionada con otra transacción B	Pago - Venta
A está contiguo a B	TPDV - TPDV, probablemente no aplicable
A es propiedad de B	TPDV - Tienda

Análisis Orientado a Objetos Ej. punto de venta: Modelo Conceptual

☞ TPDV Iniciado-por Gerente

- Los requerimientos no indican la necesidad de conocer, ni de registrar al gerente que inició un TPDV.

☞ Venta Iniciada por Cliente

- Los requerimientos no indican la necesidad de conocer, ni de registrar al cliente actual.

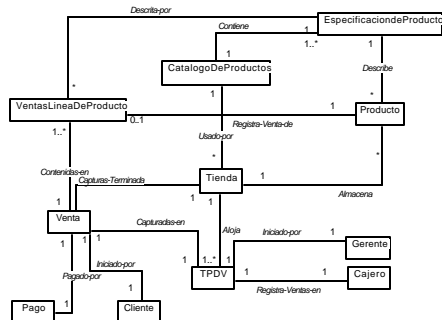
☞ Tienda Almacena Producto

- Los requerimientos no indican la necesidad de conocer o mantener la información del inventario.

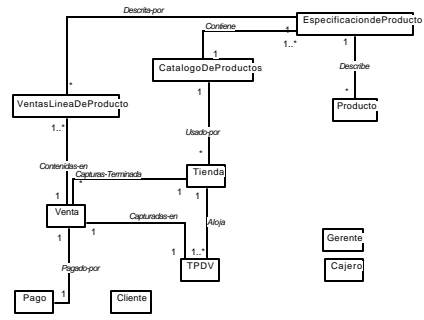
☞ Ventas Línea De Producto Registra-venta-de Producto

- Los requerimientos no indican la necesidad de mantener la información del inventario.

Análisis Orientado a Objetos Ej. punto de venta: Modelo Conceptual



Análisis Orientado a Objetos Ej. punto de venta: Modelo Conceptual

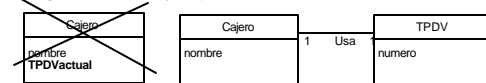


Análisis Orientado a Objetos Ej. punto de venta: Modelo Conceptual

- ⌘ Nótese que la necesidad de conocer una asociación depende de los requerimientos; un cambio de ellos (p.ej., exigir que la identificación del cajero aparezca en el recibo) altera la necesidad de recordar la relación.
- ⌘ Hay asociaciones que *deben* conocerse, pero hay otras *opcionales* que son útiles sólo para la comprensión.
 - Para documentar un conocimiento enriquecido del dominio.
 - El propósito típico de un modelo es ser leído por personas.

Análisis Orientado a Objetos Atributos [2]

- ⌘ En un modelo conceptual:
 - Los atributos que aparecen en un modelo conceptual derivan de requerimientos (p.ej., casos de uso) que indican o implican la necesidad de recordar información.
 - ❑ P.ej., si un recibo de ventas incluya fecha y hora, el concepto *Venta* requiere atributos "fecha" y "hora".
 - Los atributos deberían ser atributos simples o valores puros de datos.
 - ❑ Normalmente, el tipo de un atributo no debería ser un concepto complejo del dominio (como *Venta* o *Aeropuerto*).
 - ❑ Si un atributo parece requerir un tipo complejo (p.ej. el TPDV actual), generalmente es mejor expresarlo con una asociación.



Análisis Orientado a Objetos Contenido

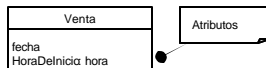
- ⌘ Modelo conceptual
- ⌘ Identificación de conceptos
- ⌘ -- Asociaciones
- ⌘ Atributos

Análisis Orientado a Objetos Ej. punto de venta: Atributos

- ⌘ Es necesario producir una lista de atributos para los conceptos del dominio de punto de venta. Debería estar reservada específicamente a los requerimientos a las simplificaciones en cuestión: Comparar productos versión 1.
- ⌘ Para eso habrá que leer los siguientes documentos:
 - especificación de requerimientos
 - casos de uso en cuestión
 - documentos de simplificaciones, clarificaciones y suposiciones
- ⌘ P.ej., podemos identificar los atributos:
 - Tienda: dirección, nombre
 - Venta: fecha, hora
 - VentasLineaDeProducto : cantidad
 - Pago: monto
 - EspecificacionDeProducto : descripción, precio, cup

Análisis Orientado a Objetos Atributos [1]

- ⌘ Un *atributo* es un valor lógico de un dato u objeto.
 - "Pertenece" a un concepto.
 - Tiene un *tipo*, que define sus posibles valores.
 - En UML, los atributos aparecen en la 2da sección de un concepto (con tipo opcional).



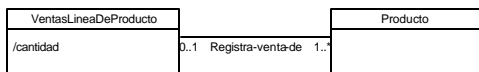
- ⌘ Los tipos más simples de atributos corresponden a tipos primitivos de datos; los tipos simples más comunes son:
 - Booleano, Tiempo (Fecha, Hora, ...), Número, String (Texto)
 - Color, Geometría (Punto, Rectángulo, ...)
 - Dirección, Teléfono, RUT, Código (de producto, postal, ...), enumeraciones

Análisis Orientado a Objetos Ej. punto de venta: Atributos

- ⌘ Es posible que el cajero reciba un grupo de productos afines (seis paquetes de pañuelos desechables) y introduzca una sola vez el CUP y la cantidad (6).
 - En consecuencia una instancia de *VentasLineaDeProducto* puede estar asociada a más de una instancia de cada producto.
 - La cantidad que introduce el cajero puede quedar registrada como atributo de *VentasLineaDeProducto*.
- ⌘ Sin embargo, también puede ser calculada a partir del valor real de multiplicidad de la relación; así puede caracterizarse como atributo derivado, el cual puede ser deducido de otra información.
 - En UML, un atributo derivado se denota con el símbolo "/".

Análisis Orientado a Objetos

Ej. punto de venta: Atributos del sistema



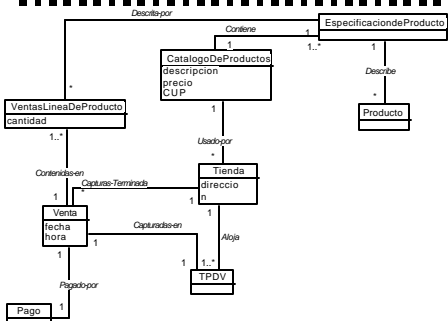
Análisis Orientado a Objetos

Contenido

- ⌘ Modelo conceptual
- ⌘ Identificación de conceptos
- ⌘ Asociaciones
- ⌘ -- Atributos

Análisis Orientado a Objetos

Ej. punto de venta: Modelo Conceptual



Análisis Orientado a Objetos

Quiz

- ⌘ ¿Qué es lo que aparece en el modelo conceptual?
- ⌘ ¿Qué es el principio del cartógrafo?
- ⌘ ¿Qué representa una asociación?
- ⌘ ¿Todas las asociaciones son importantes?
- ⌘ ¿Qué representa la multiplicidad?
- ⌘ ¿Porqué son tan importantes los nombres en el modelo?

Análisis Orientado a Objetos

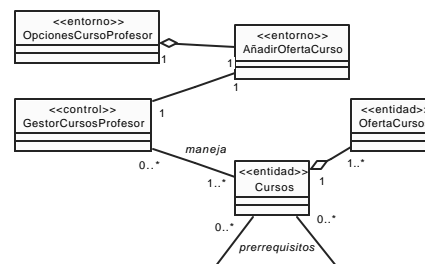
Ej. punto de venta: Modelo Conceptual

- ⌘ Hemos creado un modelo conceptual relativamente útil del dominio del punto de venta.
- ⌘ No existe un modelo apropiado para todos los casos y circunstancias, todos ellos no son más que aproximaciones al dominio que queremos entender.
- ⌘ Un buen modelo conceptual capta las abstracciones esenciales y la información indispensable para comprender el dominio dentro del contexto de los requerimientos actuales.

Análisis Orientado a Objetos

Quiz

- ⌘ ¿Cómo se interpreta este diagrama?



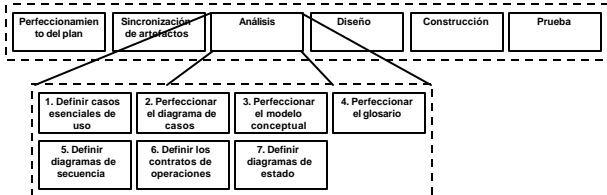
Análisis Orientado a Objetos Contenido

- ⌘ Comportamiento de los sistemas
- ⌘ Diagrama de secuencia del sistema
- ⌘ Contratos
 - Precondiciones
 - Postcondiciones

Análisis Orientado a Objetos Diagrama de Secuencia [2]

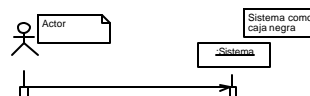
- ⌘ Los casos de uso indican cómo los actores interactúan con el sistema de software que es lo que realmente queremos crear.
- ⌘ Durante la interacción un actor genera eventos dirigidos a un sistema, solicitando alguna operación o cambio.
 - Por ejemplo, cuando un cajero introduce un código universal de producto de un artículo, está pidiendo al sistema TPDV registrar el código.

Análisis Orientado a Objetos Descripción de Comportamiento



Análisis Orientado a Objetos Diagrama de Secuencia [3]

- ⌘ El diagrama de secuencia de un sistema es una representación que muestra, en un determinado escenario, los eventos generados por actores externos, su orden y los eventos externos del sistema.
 - A todos los sistemas se les trata como caja negra; los diagramas se centran en los eventos que fluyen de los actores a los sistemas.
 - En el diagrama el tiempo avanza hacia abajo, y el ordenamiento de los eventos debería seguir el orden indicado en el caso de uso.
 - Los eventos del sistema pueden incluir parámetros.

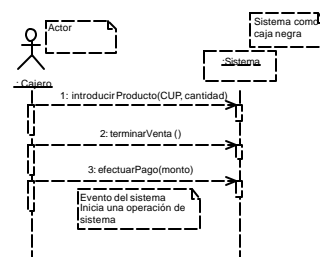


Análisis Orientado a Objetos Diagrama de Secuencia [1]

- ⌘ Antes de iniciar el diseño lógico de cómo funcionará una aplicación de software es necesario investigar y definir su comportamiento como una "caja negra".
- ⌘ El comportamiento del sistema es una descripción de lo que hace, sin explicar la manera en que lo hace. Una parte de esa descripción es un diagrama de secuencia del sistema.

Análisis Orientado a Objetos Diagrama de Secuencia [4]

- ⌘ Curso normal de los eventos en el caso Comprar Productos.



Análisis Orientado a Objetos Eventos y operaciones

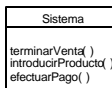
- ⌘ El evento de un sistema es un hecho externo de entrada que un actor produce en un sistema.
- ⌘ La operación de un sistema es una acción que éste ejecuta en respuesta a un evento del sistema.
 - Por ejemplo, cuando un cajero genera un evento introducirProducto, causa la ejecución de la operación introducirProducto ;
- ⌘ El nombre del evento y de la operación son idénticos; la distinción reside en que el evento es el estímulo nombrado y la operación es la respuesta (lo mismo sucede con los mensajes y los métodos).

Análisis Orientado a Objetos Diagrama de Secuencia del Sistema

- ⌘ Para elaborar un diagrama de secuencia del sistema que describa el curso normal de los eventos en un caso de uso:
 - Trace una línea que represente el sistema como una caja negra.
 - Identifique los actores que operan directamente sobre el sistema. Trace una línea para cada uno de ellos.
 - A partir del curso normal de los eventos del caso de uso identifique los eventos ("externos") del sistema que son generados por los actores. Muéstrellos gráficamente en el diagrama.
 - A la izquierda del diagrama puede incluir o no el texto del caso de uso.

Análisis Orientado a Objetos Registro de las operaciones

- ⌘ Para determinar el conjunto de las operaciones requeridas del sistema se identifican sus eventos. Cuando se utilizan los parámetros, las operaciones son las siguientes:
 - EfectuarPago(CUP, Cantidad)
 - TerminarVenta ()
 - EfectuarPago(monto)
- ⌘ ¿Donde deberían registrarse estas operaciones?. En UML se pueden agruparse las operaciones como operaciones de tipo Sistema. Los parámetros son opcionales.



Análisis Orientado a Objetos Diagrama de Secuencia

- ⌘ Consideramos ahora el caso de uso Comprar Productos a fin de identificar los eventos del sistema.
 - Primero debemos determinar los actores que interactúan directamente con el sistema de software.
 - El cliente interactúa con el cajero, pero no directamente con el sistema TPDV; esto sólo lo hace el cajero.
 - Por tanto, el cliente no es un generador de eventos del sistema, sólo el cajero lo es.

Análisis Orientado a Objetos Registro de las operaciones

- ⌘ Observe que la representación del tipo Sistema es muy diferente a lo que se expresó en el modelo conceptual.
- ⌘ Los elementos de éste representan conceptos del mundo real; en cambio, el tipo Sistema es un concepto artificial.
 - Ello se debe a la naturaleza de la información que estamos representando: mientras el modelo conceptual es la información estática, el tipo Sistema representa el comportamiento de sistema, el cual es la información dinámica.

Análisis Orientado a Objetos Diagrama de Secuencia

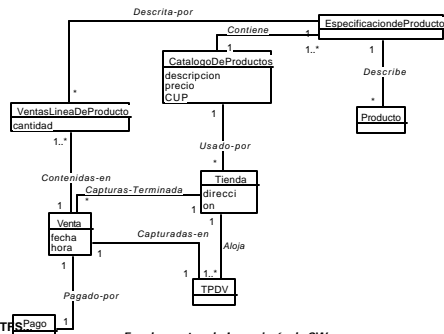
- ⌘ Los eventos de un sistema (y sus operaciones asociadas) deben expresarse en el nivel de propósito y no en el nivel el medio físico de entrada o de elementos de la interfaz.
- ⌘ También mejora la claridad, si el nombre de un evento del sistema comienza con un verbo (agregar, introducir, terminar, efectuar), ya que recalca que los eventos están orientados a comandos.
 - Así, "terminarVenta" es preferible a "IntroducirTeclaOprimida" porque capta mejor el propósito de la operación: mantiene un carácter abstracto y no se pronuncia respecto a las decisiones de diseño sobre cuál interfaz sirve para capturar el evento del sistema.

Análisis Orientado a Objetos Diagrama de Secuencia

En cuanto a expresar las operaciones en el nivel de propósito, procure alcanzar el nivel más alto o la meta final de asignar nombre a la operación. Por ejemplo, respecto a la operación que captura el pago:

- IntroducirImporteOfrecido(monto) – deficiente
- IntroducirPago(monto) – mejor
- EfectuarPago – quizá mejor aún

Análisis Orientado a Objetos Comportamiento de los sistemas: contratos



Análisis Orientado a Objetos Modelo de análisis

Los diagramas de la secuencia de un sistema forman parte del modelo de comportamiento del sistema.

El modelo de análisis se compone de:

- Modelo de casos de uso de análisis (modelo dinámico)
 - ❑ Casos de uso de alto nivel o esenciales
 - ❑ Diagramas de casos de uso
- Modelo conceptual (modelo estático)
 - ❑ Diagramas de estructura estática para los conceptos de dominio
- Modelo de comportamiento (modelo dinámico)
 - ❑ Diagramas de secuencia del sistema
 - ❑ Contratos para las operaciones de sistema
- Modelo de estado del análisis (modelo dinámico)
 - ❑ Diagramas de estado para conceptos y casos de uso

Análisis Orientado a Objetos Comportamiento de los sistemas: contratos

En términos generales, un contrato es un documento que describe lo que una operación se propone lograr.

- Suele redactarse en un estilo declarativo, enfatizando lo que sucederá y no cómo se conseguirá.
- Los contratos suelen expresarse a partir de los cambios de estado de las precondiciones y de las poscondiciones.
- Puede elaborarse un contrato tanto para un método de una clase, como para una operación más global del sistema, aunque por ahora nos concentramos en su uso para las operaciones globales del sistema.

El contrato de operación del sistema describe cambios del estado del sistema total cuando se llama una de sus operaciones.

Análisis Orientado a Objetos Comportamiento de los sistemas: contratos

Los contratos contribuyen a definir el comportamiento de un sistema; describen el efecto de las operaciones sobre el sistema.

- El lenguaje UML (pero no el Rational Rose) ofrece un soporte para definir las precondiciones y las poscondiciones de las operaciones.

Los contratos de operación de sistema se elaboran durante la fase de análisis en un ciclo de desarrollo.

Su desarrollo depende del desarrollo previo del modelo conceptual, de los diagramas de secuencia de sistema y la identificación de sus operaciones.

Análisis Orientado a Objetos Comportamiento de los sistemas: contratos

El siguiente ejemplo describe un contrato de la operación "IntroducirProducto" del sistema:

Contrato

- Nombre:** IntroducirProducto (cup:numero, cantidad:entero)
- Responsabilidades:** Capturar (registrar) la venta de un producto y agregarla a la venta. Desplegar la descripción y el precio del producto.
- Tipo:** Sistema
- Referencias cruzadas:** Funciones del sistema: R1.1, R1.3, R1.6
Casos de uso: Comprar productos
- Notas:** Utilizar el acceso super rápido a la base de datos.
- Excepciones:** Si el CUP no es válido, indicar que se cometió.
- Salida:**
- Precondiciones:** El sistema conoce el CUP.

Análisis Orientado a Objetos Comportamiento de los sistemas: contratos

Poscondiciones:

Si se trata de una nueva venta, se creó una *Venta* (creación de instancia).
Si se trata de una nueva venta, la nueva *Venta* fue asociada a un *TPDV* (asociación formada).
Se creó una instancia *VentasLineadeProducto* (creación de instancia).
Se asoció una instancia de *VentasLineadeProducto* a la *Venta* (asociación formada).
Se asignó una cantidad a *VentasLineadeProducto.cantidad* (modificación de atributo).
Se asoció una instancia *VentasLineadeProducto* a la instancia *EspecificaciondeProducto*, basado en la correspondencia del CUP (asociación formada).

⌘ No todas las secciones del contrato son necesarias; se recomienda llenar Responsabilidades y Poscondiciones.

Análisis Orientado a Objetos Postcondiciones

- ⌘ Después de la sección de Responsabilidades, la parte más importante del contrato son las postcondiciones, que estipulan cómo cambió el sistema tras la operación.
- ⌘ Las postcondiciones se expresan dentro del modelo conceptual.
 - ¿Qué instancias es posible crear? La respuesta es: las provenientes del modelo conceptual.
 - ¿Qué asociaciones es posible formar? La respuesta es: las que están en el modelo conceptual.

Análisis Orientado a Objetos Comportamiento de los sistemas: contratos

Contrato	
Nombre:	Nombre de la operación y sus parámetros.
Responsabilidades:	Descripción informal de las responsabilidades que debe cumplir la operación.
Tipo:	Nombre del tipo (concepto, clase de software, interfaz).
Referencias cruzadas:	Números de referencia de las funciones del sistema, casos de uso, etc.
Notas:	Declaraciones del diseño referentes a la operación. Por ejemplo, si se sabe que se prefiere un algoritmo particular para manejar la operación, esa sección es el sitio indicado.
Excepciones:	Casos excepcionales.
Salida:	Sólo dentro del sistema, no incorpore salidas de interfaz de usuario, mensajes o registros que se envían fuera del sistema.
Precondiciones:	Suposiciones acerca del estado del sistema antes de ejecutar la operación.
Poscondiciones:	El estado del sistema después de la operación.

Análisis Orientado a Objetos Postcondiciones

- ⌘ Cuando se formulan contratos, en general se agregarán al modelo conceptual nuevos conceptos, atributos y asociaciones.
 - Mejore el modelo conforme a los nuevos descubrimientos, mientras reflexiona sobre los contratos de las operaciones.
- ⌘ Las poscondiciones deberían expresar el estado de un sistema, no las acciones a realizar.
 - Expréselas en tiempo pasado para enfatizar que se trata de declaraciones sobre un cambio pretérito de estado. Por ejemplo:
 - Se creó una instancia *VentasLineadeProducto* (mejor)
 - En lugar de
 - Crear una instancia *VentasLineadeProducto* (peor)

Análisis Orientado a Objetos Comportamiento de los sistemas: contratos

- ⌘ Aplique la siguiente sugerencia para elaborar contratos.
 - Identifique las operaciones del sistema a partir de los diagramas de secuencia.
 - Elabore un contrato en cada operación del sistema.
 - Comience redactando la sección Responsabilidades; después describa informalmente el propósito de la operación.
 - Complete la sección de Postcondiciones, describiendo en forma declarativa los cambios de estado de los objetos en el modelo conceptual.
 - Para describir las postcondiciones utilice las siguientes categorías:
 - Creación y eliminación de instancias.
 - Modificación de atributos.
 - Asociaciones formadas y canceladas.

Análisis Orientado a Objetos Postcondiciones

- ⌘ Entonces, mire el contrato desde la perspectiva de un escenario y un telón.
 - Tome una fotografía de escenario antes de la operación
 - Corra el telón y aplique la operación del sistema (ruido de fondo con sonidos).
 - Corra el telón y tome una segunda fotografía.
 - Compare las fotografías de antes y después, y exprese como poscondiciones los cambios del estado del escenario (se creó la instancia *VentasLineadeProducto...*).

Análisis Orientado a Objetos Postocondiciones: Ejemplo

⌘ Postocondiciones:

- Si se trata de una nueva venta, se creó una *Venta* (*creación de instancia*).
- Si se trata de una nueva venta, la nueva *Venta* fue asociada a un *TPDV* (*asociación formada*).
- Se creó una instancia *VentasLineadeProducto* (*creación de instancia*).
- Se asoció una instancia de *VentasLineadeProducto* a la *Venta* (*asociación formada*).
- Se asignó una cantidad a *VentasLineadeProducto.cantidad* (*modificación de atributo*).
- Se asoció una instancia *VentasLineadeProducto* a la instancia *EspecificaciondeProducto*, basado en la correspondencia del CUP (*asociación formada*).

Análisis Orientado a Objetos Postocondiciones: Ejemplo

- ⌘ Una vez que el cajero capturó el CUP y la cantidad del producto, ¿qué asociaciones entre los objetos nuevos y actuales debieron haber sido formadas o canceladas?
- ⌘ Habría que relacionar la nueva instancia de *VentasLineadeProducto* con sus *Ventas* y con su *Producto*. Si se trataba de una nueva *Venta* debió haber sido relacionada con la *TPDV* dentro del cual es registrada.
 - Se asoció una instancia de *VentasLineadeProducto* a la *Venta* (*asociación formada*).
 - Si se trata de una nueva venta, la nueva *Venta* fue asociada a un *TPDV* (*asociación formada*).
 - Se asoció una instancia *VentasLineadeProducto* a la instancia *EspecificaciondeProducto*, basado en la correspondencia del CUP (*asociación formada*).

Análisis Orientado a Objetos Postocondiciones: Ejemplo

- ⌘ Una vez que el cajero capturó el CUP y la cantidad del producto, ¿Qué ha de crearse?
- ⌘ Si se trata de una nueva venta, habría que crear una instancia para una nueva *Venta*. Una instancia *VentasLineadeProducto* debería ser creada de modo incondicional.
 - Si se trata de una nueva venta, se creó una *Venta* (*creación de instancia*).
 - Se creó una instancia *VentasLineadeProducto* (*creación de instancia*).

Análisis Orientado a Objetos Precondiciones

- ⌘ Las precondiciones definen las suposiciones sobre el estado del sistema al iniciarse la operación.
- ⌘ Hay muchas precondiciones que pueden declararse en una operación, pero la experiencia revela que vale la pena mencionar las siguientes:
 - Cosas que son importantes de probar en el software en algún momento de la ejecución de la operación.
 - Cosas que serán sometidas a prueba, pero de las cuales depende el éxito para subrayar la importancia y para hacer notarlas a los otros lectores.

Análisis Orientado a Objetos Postocondiciones: Ejemplo

- ⌘ Una vez que el cajero capturó el CUP y la cantidad del producto, ¿qué atributos de los objetos nuevos o actuales deberían ser modificados? Habría que establecer la cantidad de *VentasLineadeProducto*.
 - Se asignó una cantidad a *VentasLineadeProducto.cantidad* (*modificación de atributo*).

Análisis Orientado a Objetos Contratos

- ⌘ El problema más común consiste en olvidar incluir la formación de asociaciones. Sobre todo cuando se crean nuevas instancias, muy probablemente será necesario haber establecido las asociaciones a varios objetos.

Análisis Orientado a Objetos Contrato para introducirProducto

Contrato

Nombre: IntroducirProducto (p:numero ,cantidad:entero)
Responsabilidades: Capturar (registrar) la venta de un producto y agregarla a la venta. Desplegar la descripción y el precio del producto.
Tipo: Sistema
Referencias cruzadas: Funciones del sistema: R1.1, R1.3, R1.6
 Casos de uso: Comprar productos
Notas: Utilizar el acceso super rápido a la base de datos.
Excepciones: Si el CUP no es válido, indicar que se cometió un error.
Salida:
Precondiciones: El sistema conoce el CUP.
Poscondiciones:

- Si se trata de una nueva venta, fue creada una *venta*.
- Si se trata de una nueva venta, la nueva *venta* fue asociada a un *TPDV*.
- Se creó una instancia *ventas.LineadeProducto*
- Se asoció una instancia de *ventas.LineadeProducto* a la *venta*.
- Se asignó una cantidad a *ventas.LineadeProducto.cantidad*
- Se asoció una instancia *ventas.LineadeProducto* a la instancia *especificaciondeProducto* basado en la correspondencia del CUP.



Análisis Orientado a Objetos Contrato para Inicio

Contrato

Nombre: Inicio()
Responsabilidades: Inicializar el sistema
Tipo: Sistema.
Referencias cruzadas:
Notas:
Excepciones:
Salida:
Precondiciones:
Poscondiciones:
 Se creó una instancia *Tienda*, *TPDV*, *CatalogodeProducto* y *EspecificacionesdeProducto*
 Se asoció *CatalogodeProducto* a *EspecificacionesdeProducto*.
 Se asoció *Tienda* a *CatalogodeProductos*.
 Se asoció *TPDV* a *TPDV*.
 Se asoció *TPDV* a *CatalogodeProductos*.



Análisis Orientado a Objetos Contrato para terminarVenta

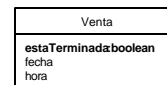
Contrato

Nombre: terminarVenta()
Responsabilidades: Registrar que es el final de la captura de los productos de la venta y desplegar el total de la venta.
Tipo: Sistema
Referencias cruzadas: Funciones del sistema: R1.2
 Casos de uso: Comprar productos
Notas:
Excepciones: Si no esta realizándose una venta, indicar que se cometió un error.
Salida:
Precondiciones:
Poscondiciones:
 Estableció *venta.estaTerminada* en *verdadero*.



Análisis Orientado a Objetos Cambios en el modelo conceptual

- ⌘ Estos contratos sugieren la existencia de un dato que todavía no ha figurado en el modelo conceptual: la terminación de la captura de los productos en la venta. Lo modifica la operación *terminarVenta* y la especificación *efectuarPago* lo toma como precondición.
- ⌘ Una forma de representar esta información es a través de un atributo *estaTerminada* de la *venta*, por medio de un valor booleano.



Análisis Orientado a Objetos Contrato para efectuarPago

Contrato

Nombre: efectuarPago(monto : número)
Responsabilidades: Registrar el pago, calcular el saldo e imprimir el recibo.
Tipo: Sistema.
Referencias cruzadas: Funciones del sistema: R2.1
 Casos de uso: Comprar productos
Notas:
Excepciones: Si la venta no está concluida, indicar que se cometió un error.
Salida:
Precondiciones: La venta esta terminada.
Poscondiciones:
 Se creó una instancia *Pago*.
 Se asoció el *Pago* a una *venta*.
 Se asignó el valor del monto a *Pago.MontoOfrecido*.
 Se asoció la *venta* a la *Tienda* para agregarla al registro histórico de las ventas terminadas.



Análisis Orientado a Objetos Resumen

- ⌘ Comportamiento de los sistemas
- ⌘ Diagrama de secuencia del sistema
- ⌘ Contratos
 - Precondiciones
 - Postcondiciones



Análisis Orientado a Objetos Quiz

- ☒ ¿Qué es el comportamiento de un sistema?
- ☒ ¿Qué representa el diagrama de secuencia del sistema?
- ☒ ¿Cuál es la diferencia entre evento y operación?
- ☒ ¿Porqué se crea el concepto sistema?
- ☒ ¿Cuál es el nombre correcto para una operación?
- ☒ ¿Cuál es la diferencia entre lo estático y lo dinámico?
- ☒ ¿Porqué son necesarios los contratos y cuantos son para un caso de uso?
- ☒ ¿Cuales son las condiciones para elaborar un contrato?
- ☒ ¿Cuales son las partes más importantes del contrato?