



Lenguajes de Programación II

Programación Orientada a Objetos

UNIDAD IV – PARTE 1

ANÁLISIS ORIENTADO A OBJETOS

Docentes: Lic. Rodrigo Velázquez

FACULTAD POLITÉCNICA – UNA

2022



AOO



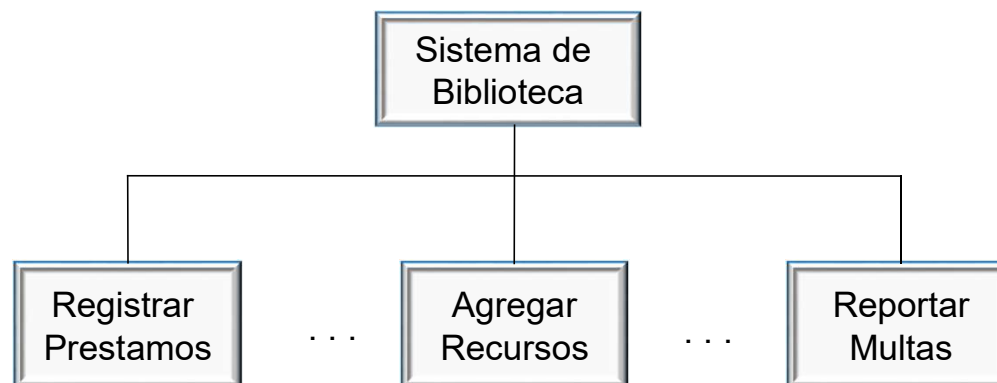
CONTENIDO

Análisis Orientado a Objetos

- Análisis y Diseño
- Tareas en el Análisis
- Requerimientos
- Funciones del Sistema
- Casos de Uso
- Diagramas de caso de uso
- Modelo Conceptual
- Asociación

ANÁLISIS Y DISEÑO

- La mayoría de proyectos de software son complejos.
- La estrategia primaria para superar la complejidad es la descomposición (divide y vencerás).
 - Dividir el problema en unidades más pequeñas que sean manejables.
- En el Análisis y Diseño Estructurados se descompone el problema en sus funciones o procesos.
 - Este método origina una división jerárquica de procesos constituidos por sus subprocesos.



ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

- Esta estrategia busca descomponer un sistema en un conjunto de objetos relacionados.
- Así, durante el Análisis y Diseño Orientado a Objetos un Sistema de Bibliotecas podría descomponerse en, al menos (dependiendo del grado de detalle), los siguientes objetos:





ANÁLISIS ORIENTADO A OBJETOS

- AOO (Análisis Orientado a Objetos). Se basa en conceptos como objetos, atributos, clases y miembros.
- Objetivo
 - Desarrollar una serie de modelos que describan el software de computadora al trabajar para satisfacer un conjunto de requisitos definidos por el cliente.

ANALISIS ORIENTADO A OBJETOS

- En esta fase intenta identificar y describir los conceptos (objetos) del dominio del problema que se intenta resolver.
- Los esfuerzos se centran en realizar una investigación acerca de en que consiste el problema.
- Algunas tareas importantes de todo Análisis son:
 - **1. Definir los Requerimientos.**
 - **2. Definir los Casos de Uso.**
 - **3. Crear y perfeccionar los Diagramas de Casos de Uso.**
 - **4. Crear y perfeccionar el Modelo Conceptual.**

CASO DE ESTUDIO

- Sistema de Terminal de Punto de Venta (TPDV).
 - Consiste en una terminal con un sistema automatizado mediante el cual sea posible registrar las ventas y cobrar los pagos.
 - El Sistema se utiliza en una tienda de venta al detalle de productos (como un Supermercado).

REQUERIMIENTOS

- *Los requerimientos son una descripción de las necesidades o deseos de un producto.*
 - 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



PANORAMA GENERAL

- Aquí se define cual es el objetivo de crear el sistema y definir a grandes rasgos cual será utilidad global.

- Panorama General TPDV:
 - Este proyecto tiene por objeto crear un Sistema TPDV que se utilizará en locales de ventas de productos al menudeo, tipo Supermercados, Ferreterías, Almacenes , etc...
 - El Sistema TPVD deberá ser capaz de:
 - Registrar los productos a ser vendidos,
 - Registrar los clientes comunes,
 - Manejar el inventario de productos,
 - Gestionar un grupo de usuarios ,
 - etc...



METAS

- La Meta debe especificar los beneficios concretos que se obtendrán al usar el Software.

- Meta TPDV:
 - La meta principal es automatizar el pago en las cajas registradoras, y dar soporte a servicios más rápidos, más baratos y mejores.
 - Más concretamente, la meta del sistema TPDV incluye:
 - Pago rápido de los clientes.
 - Análisis rápido y exacto de las ventas.
 - Control automático del inventario.
 - Manejar un registro de clientes y de las cuentas de los clientes.

FUNCIONES DEL SISTEMA

- *La Funciones del Sistema indican lo que el sistema debe hacer.*
 - Durante el Análisis, se deben identificar estas funciones y listarlas en grupos lógicos.
- *Para verificar que “X” es una función del Sistema, la siguiente frase debe tener sentido:*
 - El sistema deberá hacer X. Ej:
 - El sistema deberá registrar el tipo de pago de las ventas.
- *Las funciones deben clasificarse en:*
 - Evidentes: la funciones deben realizarse y el usuario debe notar su realización
 - Ocultas: Las funciones deben realizarse y el usuario puede no notar su realización
 - Superfluas: Las funciones pueden incluirse o no en los requerimientos.

FUNCIONES BÁSICAS

Ref.	Función	Cat.
R1.1	Registra la venta en proceso: los productos comprados.	evidente
R1.2	Calcula el total de la venta actual; se incluye el impuesto.	evidente
R1.3	Captura la información sobre el objeto comprado usando un lector de código de barras, o permitiendo la captura manual del código de producto.	evidente
R1.4	Reduce las cantidades del inventario cuando se realiza una venta.	oculta
R1.5	Registra el detalle de las ventas efectuadas.	oculta
R1.6	Obliga a introducir una identificación y una contraseña para poder utilizar el sistema.	evidente
R1.7	Ofrece un mecanismo de almacenamiento persistente.	oculta
R1.8	Ofrece mecanismos de comunicación entre el sistema TPDV y otros sistemas (POS Banco).	oculta
R1.9	Muestra la descripción y el precio del producto registrado.	evidente
R1.10	Se debe mostrar un mensaje de finalización para cada transacción	Superflua

FUNCIONES DE PAGO

Ref.	Función	Cat.
R2.1	Maneja los pagos en efectivo, capturando la cantidad pagada y calculando el cambio.	evidente
R2.2	Maneja los pagos a crédito, capturando la información crediticia a partir de una lectora de tarjetas de crédito y autorizando los pagos con el servicio de autorización externa a través de una conexión por modem.	evidente
R2.3	Maneja los pagos con cheque, capturando nro. de cédula y teléfono del cliente.	evidente
R2.4	Registra los pagos a crédito en el sistema de cuentas por cobrar.	oculta
R2.5	Informa de la cantidad de operaciones y el promedio del ventas realizadas en la terminal por el usuario actual	superflua

ATRIBUTOS DEL SISTEMA

- **Requerimientos no Funcionales**
- *Representan características deseables en el sistema, pero que no tienen relación con las funciones que debe realizar.*
 - Los atributos pueden tener un conjunto de *detalles*, los cuales tienden a ser valores discretos, simbólicos o descriptivos
 - Algunos atributos también pueden tener *restricciones de frontera*, que son condiciones obligatorias a cumplir.

Atributo	Detalles y restricciones de frontera
tiempo de respuesta	(<i>restricción de frontera</i>) Cuando se registre un producto vendido, la descripción y el precio aparecerán en un segundo.
metáfora de interfaz	(<i>detalle</i>) Ventanas orientadas a la metáfora de un formulario y cuadros de diálogo. (<i>detalle</i>) Maximiza una navegación fácil con teclado y no con mouse.
tolerancia a fallas	(<i>restricción de frontera</i>) Debe registrar los pagos a crédito autorizados que se hagan a las cuentas por cobrar en un plazo de 24 horas, aun cuando se produzcan fallas de energía o del equipo.
plataformas del sistema operativo	(<i>detalle</i>) Microsoft Windows 95, 98, 2000 y NT.

ATRIBUTOS DEL SISTEMA

- **Requerimientos no Funcionales**
- *Se deben describir/especificar los atributos del sistema que se relacionan con las funciones del sistema.*
- *Los detalles y las restricciones de frontera pueden catalogarse como obligatorios u opcionales.*

Ref.	Función	Atributo	Detalles y restricciones	Cat.
R1.9	Mostrar la descripción y el precio del producto registrado.	tiempo de respuesta	1 segundo como máximo	obligatorio
		metáfora de interfaz	Pantallas basadas en formularios. Con colores.	obligatorio
R2.4	Registrar los pagos a crédito en el sistema de cuentas por cobrar, pues el servicio de autorización de crédito debe a la tienda el importe del pago.	tolerancia a fallas	Debe registrar en las cuentas por cobrar en un plazo de 24 horas, aun cuando se produzcan fallas de energía o del equipo.	obligatorio
		tiempo de respuesta	10 segundos como máximo	obligatorio

CASOS DE USO

- *Consisten en descripciones narrativas de los procesos del sistema, describiéndolos de manera clara y concisa.*
- *Requieren tener al menos un conocimiento parcial de los requerimientos del sistema.*
- *Para especificar los casos de uso en el lenguaje UML, se utiliza una elipse que encierra el nombre del caso.*



Venta de Productos

CASOS DE USO DE ALTO NIVEL

- Describen los procesos del Sistema sin entrar en demasiados detalles.
- Permiten definir y entender rápidamente los procesos principales del Sistema.
- Los Casos de Uso de Alto Nivel utilizan el siguiente formato:
 - Caso de Uso: Nombre del Caso de Uso.
 - Actores: Agentes que participan del caso de uso, se indica cual es el iniciador.
 - Tipo: Primario, secundario u opcional. Esencial o Real.
 - Descripción: Narración de cómo sucede el caso de uso.
- Un Caso de Uso Primario describe un proceso importante del Sistema. Ej. Comprar Productos.
- Un Caso de Uso Secundario describe un proceso de menor importancia dentro del Sistema. Ej. Alertar de existencia mínima.
- Un Caso de Uso Opcional se refiere a procesos que pueden no ser abordados (implementados) por el Sistema. Ej. Controlar Vencimientos.

CASOS DE USO DE ALTO NIVEL

- Caso de uso: Venta de Productos

Actores: Cliente (iniciador), Cajero

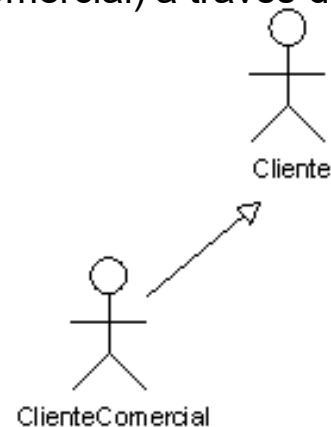
Tipo: Primario.

Descripción: Un Cliente llega a la caja registradora con los artículos que va a comprar. El Cajero registra los artículos. El Sistema registra la venta, sus detalles y genera la factura. El Cajero cobra el importe. Al terminar la operación, el Cliente se marcha con los productos.



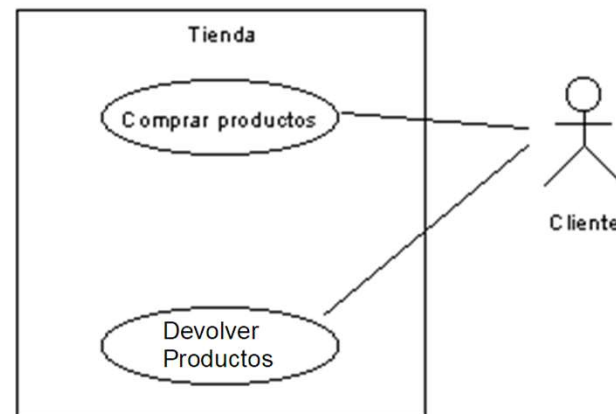
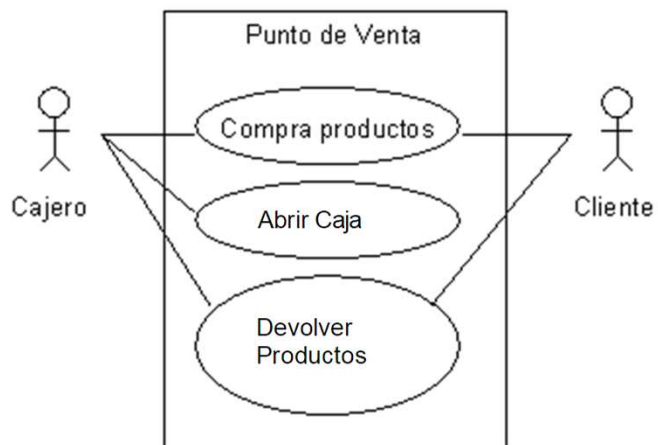
ACTORES

- Un actor representa un conjunto coherente de roles que juegan los usuarios de los casos de uso cuando interactúan con éstos.
- Los actores pueden ser:
 - Personas (roles que desempeñan las personas)
 - Aparatos eléctricos o mecánicos
 - Otros sistemas de cómputo.
- Se pueden definir categorías generales de actores (como cliente en el ejemplo de abajo) y especializarlos (como ClienteComercial) a través de relaciones de generalización (Herencia en OO).



DIAGRAMAS DE CASOS DE USO

- Un Diagrama de Casos de Uso sirve para indicar que Actores participan de cada caso de uso del Sistema.
- La Frontera de un diagrama define el ámbito desde el cual se definen o identifican los casos de uso





DETERMINACIÓN DE CASOS DE USO

- Para determinar los casos de uso de un sistema es necesario:
 - Identificar los actores, sus funciones y la frontera.
 - Describir los casos de uso.
 - Crear el diagrama de casos de uso

IDENTIFICACIÓN

- *La frontera corresponde a una Terminal de Punto de Venta*
- *Los Actores y sus funciones son los siguientes:*

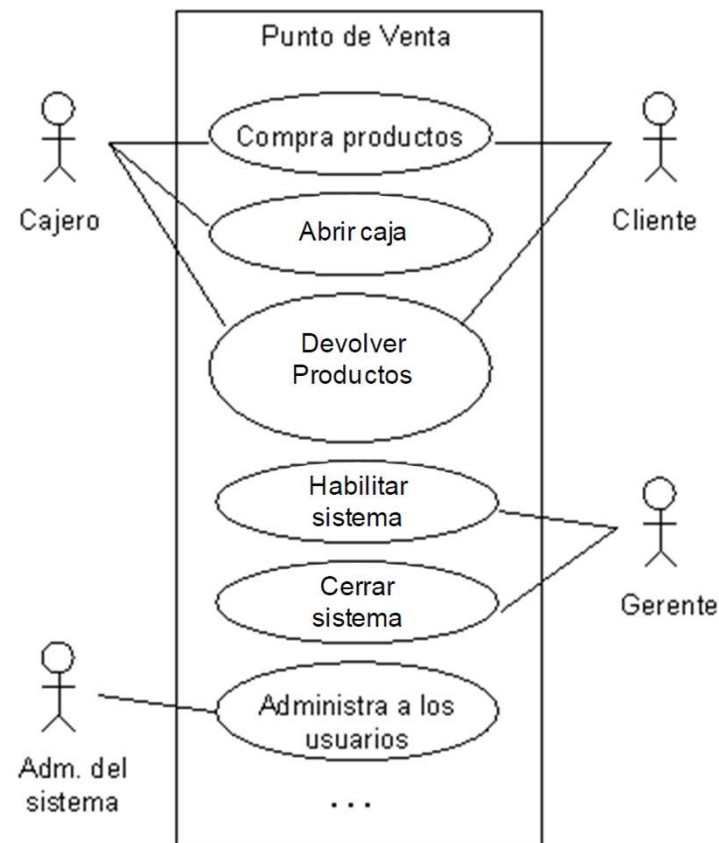
Cajero	Registrar Ventas Abrir/Cerrar una TPDV
Cliente	Comprar productos Devolver los productos
Gerente	Habilitar el sistema Cerrar el Sistema
Administrador del sistema	Incorporar nuevos usuarios



DESCRIPCIÓN DE CASOS DE USO

- *Caso de uso: Habilitación del Sistema*
- *Actores: Gerente.*
- *Tipo: Primario.*
- *Descripción: Un gerente activa las terminales de punto de venta con el fin de permitir que las usen los cajeros. El Gerente comprueba que la fecha y la hora sean correctos, que cada terminal tenga conectividad con el servidor central y elige las terminales que se habilitarán. Hecho esto, las TPDVs habilitadas están listas para ser usadas por los cajero.*

DIAGRAMAS DE CASO DE USO



CASOS DE USO EXPANDIDOS

- *Muestran detalles más profundos acerca de los procesos del sistema, permitiendo conocer mejor como se llevaran a cabo los procesos dentro del sistema y como se cumplen con los requerimientos.*
- *El formato para la descripción es el siguiente:*

Caso de uso: Nombre del caso de uso

Actores: Lista de actores, se indica quién inicia el caso de uso

Propósito: Intención del caso de uso

Resumen: Síntesis del caso de uso de alto nivel.

Tipo: Primario, secundario u opcional. Esencial o real.

Referencias cruzadas: Casos de uso y Funciones del Sistema relacionados.

Descripción: Descripción del caso de uso.

- *En la descripción se indica cuales son la acciones de los actores y la respuestas del sistema*
- *También se indican cuales son los cursos alternos que pueden tomarse.*

CASOS DE USO EXPANDIDOS

■ *Caso de uso: Comprar productos en efectivo*

Actores: Cliente (iniciador), Cajero.

Propósito: Capturar una venta, sus detalles y su pago en efectivo.

Resumen: Un Cliente llega a la caja registradora con artículos que desea comprar. El Cajero registra el dato del cliente y de los productos y recibe un pago en efectivo terminando la operación.

Tipo: Primario y esencial.

Referencias cruzadas: Funciones: R1.1, R1.2, R1.3, R1.7, R1.9, R2.1.

Curso normal de los eventos:

Acción del actor	Respuesta del sistema
1. Este caso de uso comienza cuando un cliente a la TPV con productos para comprar.	
2. El Cajero registra el identificador de cada producto y la cantidad.	3. Determina el precio de cada producto e incorpora la información del monto correspondiente.
4. Al terminar de introducir los productos, el Cajero indica al TPV que se concluyó la captura de los mismos.	5. Calcula y presenta el total de la venta.

CASOS DE USO EXPANDIDOS

Acción del actor	Respuesta del sistema
6. El Cajero le indica el total al cliente.	
7. El Cliente efectúa un pago en efectivo.	
8. El Cajero registra la cantidad de efectivo recibida.	9. Muestra al Cajero y al Cliente el cambio a entregar y genera la boleta.
10. El cajero deposita el efectivo recibido en la caja, y extrae el cambio.	11. Registra la venta concluida.
12. El cliente se marcha con los artículos comprados.	

Cursos alternos:

- A. Item 2: Introducción de identificador inválido. Indicar error.*
- B. Item 7: El cliente no tiene suficiente dinero. Cancelar transacción de venta o restar productos.*

CASOS DE USO - RELACIONES.

- Las relaciones son de dos tipos: de inclusión y de extensión. Ambas se especifican con una flecha de línea de puntos y el estereotipo <<include>> o <<extends>>
- **La extensión es una relación que enriquece un caso de uso sin modificarlo**

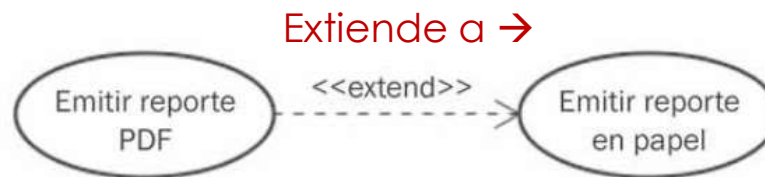


Figura 4.28 Extensión entre casos de uso.

- La inclusión implica que un caso de uso (el incluyente), incorpora el comportamiento de otro caso de uso (el incluido) como parte de su propio comportamiento en un determinado momento de su curso de acción. Modelar la inclusión sirve para evitar la repetición de pasos en distintos casos de uso, cuando son los mismos.



Figura 4.31 Inclusión entre casos de uso.

CONCEPTOS

- *Para identificar los conceptos es necesario de un análisis semántico de las descripciones textuales referentes al dominio del problema.*
- *Por ejemplo, los casos de uso expandidos proveen una buena fuente de conceptos.*
 - En el caso de uso *Comprar productos*:

Acción de los actores	Respuesta del sistema
1. Este caso de uso comienza cuando un Cliente llega a una caja de TPV con los productos que desea comprar.	
2. El Cajero registra el código de barras de cada producto. Si hay más de un producto, el Cajero puede introducir también la cantidad.	3. Determina el precio del producto y a la transacción de venta le agrega la información sobre el producto. Se muestra la descripción y el precio del producto actual.

CONCEPTOS

- *Así, los conceptos determinados por el caso de uso son:*

Cliente

Cajero

TPDV

Venta

Producto

Detalle de
Venta

ATRIBUTOS

- *Un atributo es un valor asociado a un concepto*
- *Los atributos deben referirse a propiedades de un concepto.*
- *No deben incluirse atributos que indiquen una relación*

Cliente
Nombre
ID

Cajero
Nombre
ID

TPDV
ID

Venta
ID
Fecha
Tipo

Producto
Nombre
Precio
Descripción

Detalle de Venta
Cantidad



MECANISMOS DE ABSTRACCIÓN.

➤ **Clasificación.** Es un mecanismo que relaciona individuos con especies. Este mecanismo es extremadamente útil para deducir aspectos generales de un conjunto de individuos, y tiene especial uso en el desarrollo de software.

Así, decimos que José, Julia, María y Juan son individuos particulares de la categoría usuarios (instancias de la clase Usuarios).

MECANISMOS DE ABSTRACCIÓN.

ASOCIACIÓN

➤ **Asociación.** Cada individuo puede estar relacionado de una manera u otra con otros individuos, de su misma clase o de otra. Por ejemplo, si decimos que María es la jefa de José, estamos estableciendo una asociación entre dos usuarios, que muy probablemente se pueda establecer para todas las instancias de la clase, a lo sumo con unas pocas excepciones.

Una asociación es inherentemente bidireccional. Si María es jefa de José, este último es subordinado de aquélla.

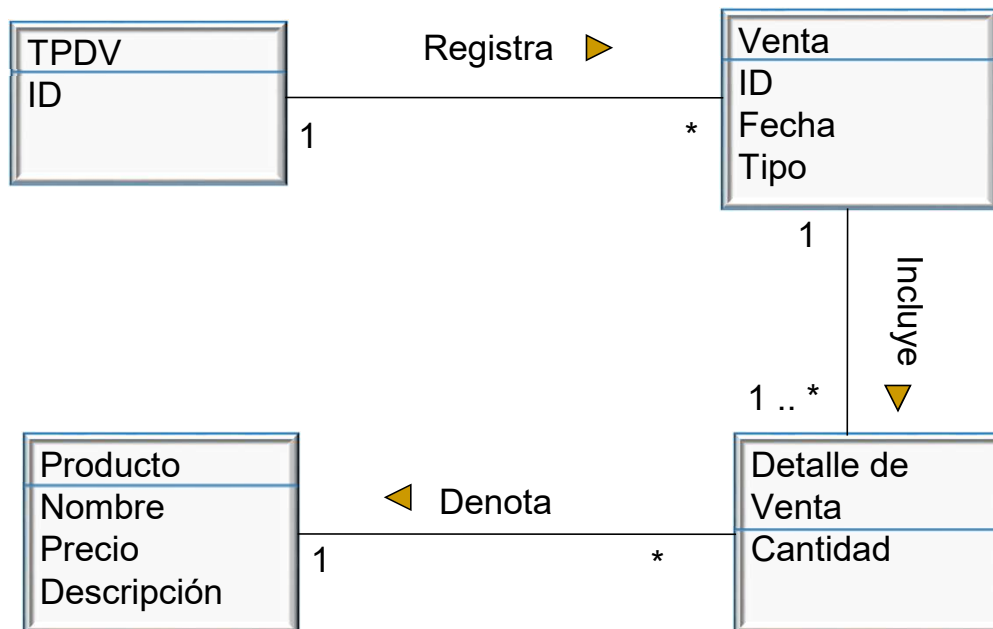
Hay ocasiones en las que las asociaciones tienen una **cardinalidad** esperada, fija o definida mediante un rango. Por ejemplo, una empresa debe tener al menos un empleado. Y cada empleado debe estar asociado a una y solo una empresa.

MECANISMOS DE ABSTRACCIÓN.

ASOCIACIÓN

- Una asociación es una relación entre dos conceptos que indica alguna conexión significativa entre ellos.
- Una asociación se representa como una línea entre conceptos, con el nombre de la asociación.
- Opcionalmente se puede poner una flecha que indique la dirección en que debe leerse la asociación.
- La multiplicidad define cuántas instancias de un tipo A pueden asociarse a una instancia del tipo B. Las expresiones de multiplicidad son:
 - * cero o más, muchos
 - 1..* uno o más
 - 1..40 de uno a cuarenta
 - 5 exactamente cinco
 - 2,4,6 exactamente dos, cuatro o seis
- Una asociación define 2 multiplicidades:
 - De A a B
 - De B a A

ASOCIACIÓN - EJEMPLO



MECANISMOS DE ABSTRACCIÓN.

➤ **Agregación o agrupación.** Es un tipo de asociación especial, es cuando uno de los extremos de la asociación puede considerarse parte del otro extremo.

La agregación es una propiedad transitiva. Si C es parte de B y B es parte de A, entonces C es parte de A.

Como toda asociación, puede haber agregación entre individuos de una misma clase. Por ejemplo, una organización compuesta podrá contener otras organizaciones como partes.



AGREGACIÓN

MECANISMOS DE ABSTRACCIÓN.

➤ **Composición.** hay ocasiones en las que en una agregación las partes no tienen sentido fuera del agregado, o no pueden formar parte de más de un agregado.

Relación todo-partes, donde, partes no puede ser independiente de todo, esto es, si deja de existir el todo, dejan de existir las partes.



COMPOSICIÓN

MECANISMOS DE ABSTRACCIÓN.

➤ **Generalización.** Es la operación por la cual establecemos que una o más clases tienen elementos en común que deseamos agrupar en una clase más genérica.

La generalización es unidireccional, transitiva, ACÍCLICA y asimétrica.

Se trata de un mecanismo de abstracción ideal para separar detalles y concentrarse en aspectos comunes entre conceptos.



HERENCIA

MODELO CONCEPTUAL

- *Un modelo conceptual explica los conceptos más significativos en un dominio del problema, identificando los atributos y las asociaciones de los mismo.*
- *Un modelo conceptual representa cosas del mundo real, no componentes del software. Es una herramienta de comunicación, con la cual se intenta comprender y transmitir los conceptos importantes y sus relaciones*
- *En estos diagramas se muestran:*
 - *Conceptos (objetos)*
 - *Asociaciones entre conceptos (relaciones).*
 - *Atributos de conceptos (atributos).*
- *En UML se representa mediante un grupo de diagramas de estructura estática donde no se define ninguna operación*



CONSTRUCCIÓN DE MODELO CONCEPTUAL

- *Para construir un Modelo Conceptual se deben aplicar los siguientes pasos:*
 - Determinar los conceptos.
 - Dibujar cada concepto.
 - Incorporar las asociaciones.
 - Agregar los atributos.

MODELO CONCEPTUAL

- La siguiente figura muestra un modelo conceptual parcial del dominio de la tienda y las ventas, en donde se utilizara el sistema de TDPV.

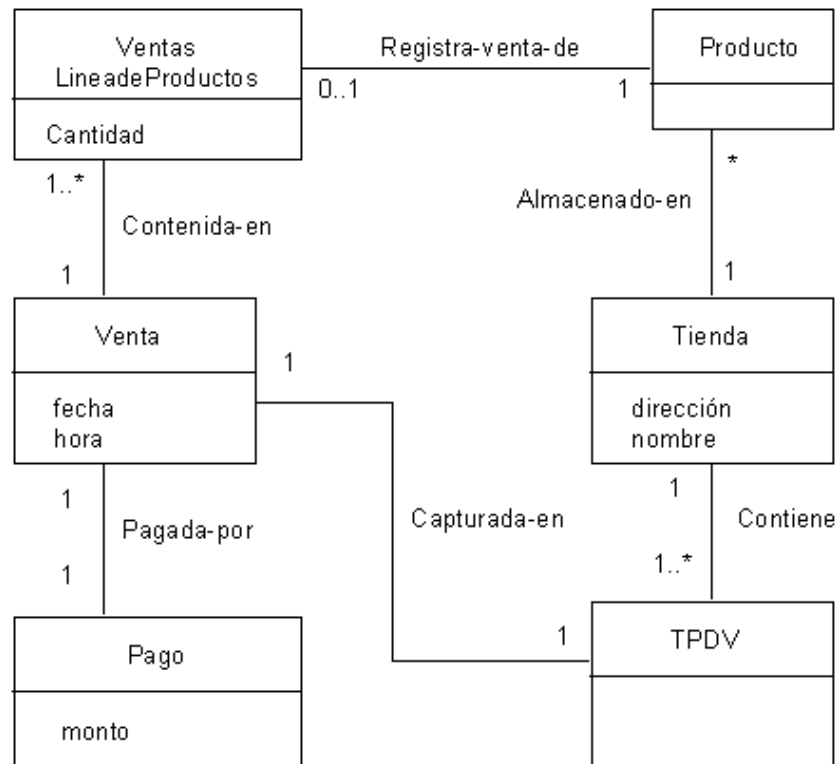
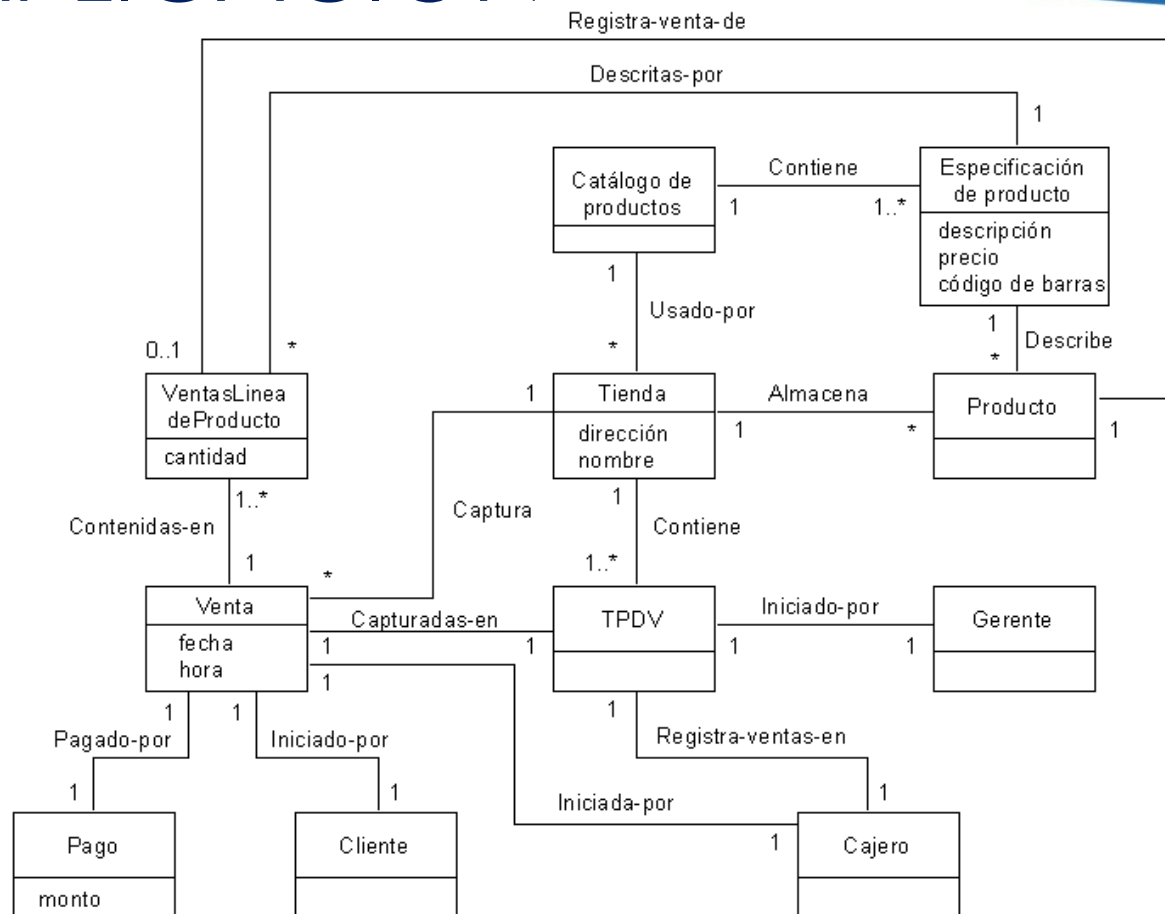


Figura 4.25 Nota aclaratoria.

APLICACIÓN





BIBLIOGRAFÍA

Deitel, P. y Deitel, H. (2016), Cómo programar en Java. Décima edición. México: Pearson educación.

FIN!

