



Unified Modeling Language 2.0

Tomás Balderas Contreras
balderas@ccc.inaoep.mx

Ingeniería de Software
Ciencias Computacionales
INAOE 2011



Contenido

1. La importancia de modelar.
2. Fundamentos de UML.
3. Diagrama de clases.
4. Interfaces.
5. Diagrama de secuencia.
6. Diagrama de casos de uso.
7. Diagrama de componentes.
8. Diagrama de despliegue



I. LA IMPORTANCIA DE MODELAR

Modelado



- ❑ Una técnica de ingeniería probada y aceptada. Es parte central en todas las actividades que conducen a la producción de buen software.
- ❑ Los modelos arquitectónicos de casas y rascacielos ayudan a visualizar el producto final. Los modelos matemáticos permiten analizar los efectos de vientos o terremotos.
- ❑ Aeronáutica, ingeniería automotriz, dispositivos eléctricos y electrónicos, películas, sociología, economía, gestión de empresas.

- ¿Qué es un modelo? Un modelo es una simplificación de la realidad.

- ¿Para qué modelamos?
 - Para controlar riesgos.
 - Para comunicar la estructura deseada y el comportamiento del sistema.
 - Para visualizar y controlar la estructura del sistema.
 - Para comprender mejor el sistema que se está construyendo.



II. FUNDAMENTOS DE UML

Orientación a objetos^[3]



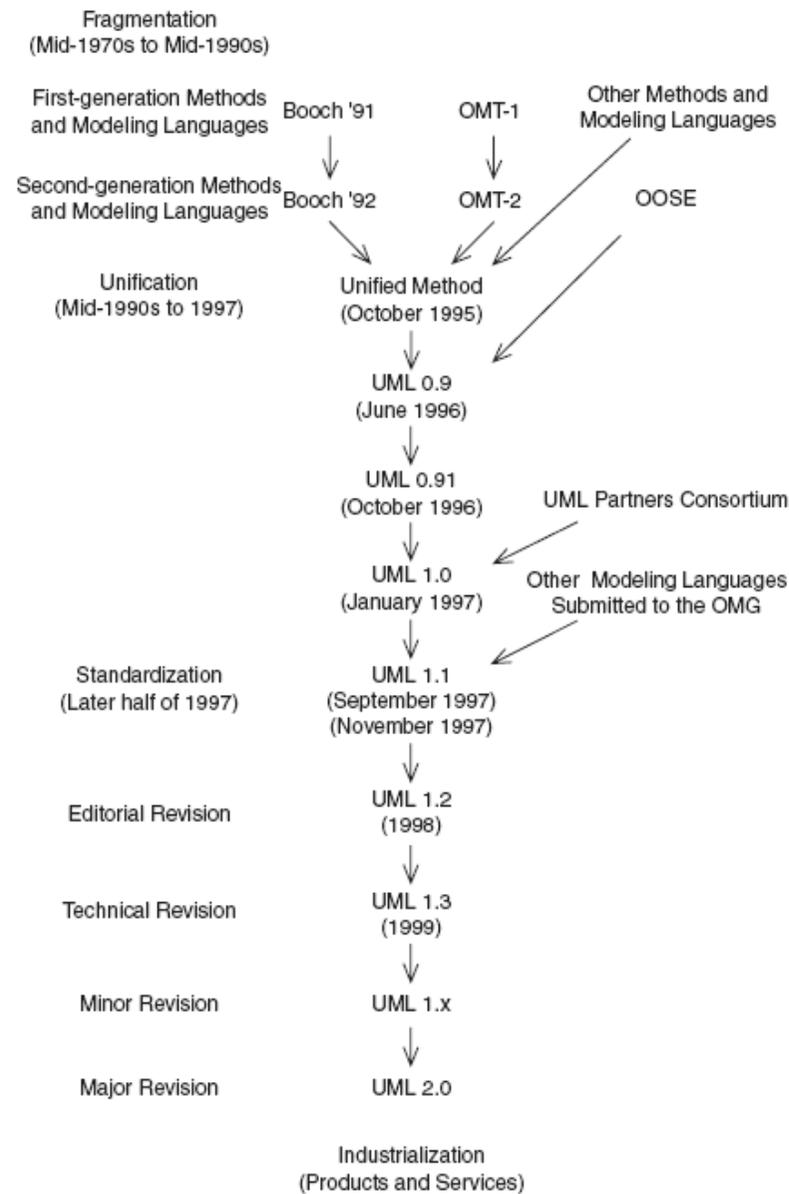
1. ¿Qué es un objeto? Es una representación de una entidad del mundo real que es responsable de tener un estado y llevar a cabo operaciones.
2. ¿Qué es un atributo y qué es una operación? Los atributos corresponden a datos encapsulados en un objeto y las operaciones a los algoritmos que procesan esos datos.



3. ¿Qué es una clase? Es una descripción generalizada que describe una colección de objetos similares.

4. ¿Qué es el análisis y diseño orientado a objetos? Durante el análisis se definen las clases del sistema, incluyendo los atributos y operaciones, y la forma en que se relacionan las clases unas con otras. Durante el diseño se define una arquitectura de software organizada en capas, se especifican subsistemas, se describen las clases y se definen los mecanismos de comunicación entre los componentes.

Evolución^[2]



Definición



1. UML es un lenguaje *estándar* para desarrollar los “planos” de un sistema de software. Debe ser usado junto con un proceso de desarrollo^[1].
2. UML es un meta-modelo basado en Meta-Object Facility (MOF) que consiste de un conjunto de meta-clases que definen a los elementos de modelado del lenguaje.

Aplicaciones^[2]



- ❑ Usos de UML:
 - Visualización.
 - Especificación.
 - Construcción.
 - Documentación.

- ❑ UML combina notaciones provenientes desde:
 - Modelado Orientado a Objetos .
 - Modelado de Datos.
 - Modelado de Componentes .
 - Modelado de Flujos de Trabajo (Workflows).

- ❑ Permite modelar sistemas de información, aplicaciones distribuidas en web, sistemas embedded, entre otros.

Bloques básicos de UML



□ Elementos:

- Estructurales.
- Comportamiento.
- Agrupación.
- Anotación.

□ Relaciones:

- Dependencia.
- Asociación.
- Generalización.
- Realización.

□ Diagramas:

- Clases.
- Objetos.
- Componentes.
- Estructura compuesta.
- Casos de uso.
- Secuencia.
- Comunicación.
- Estados.
- Actividades.
- Despliegue.
- Paquetes.

Herramientas



Rational® software



Herramienta recomendada





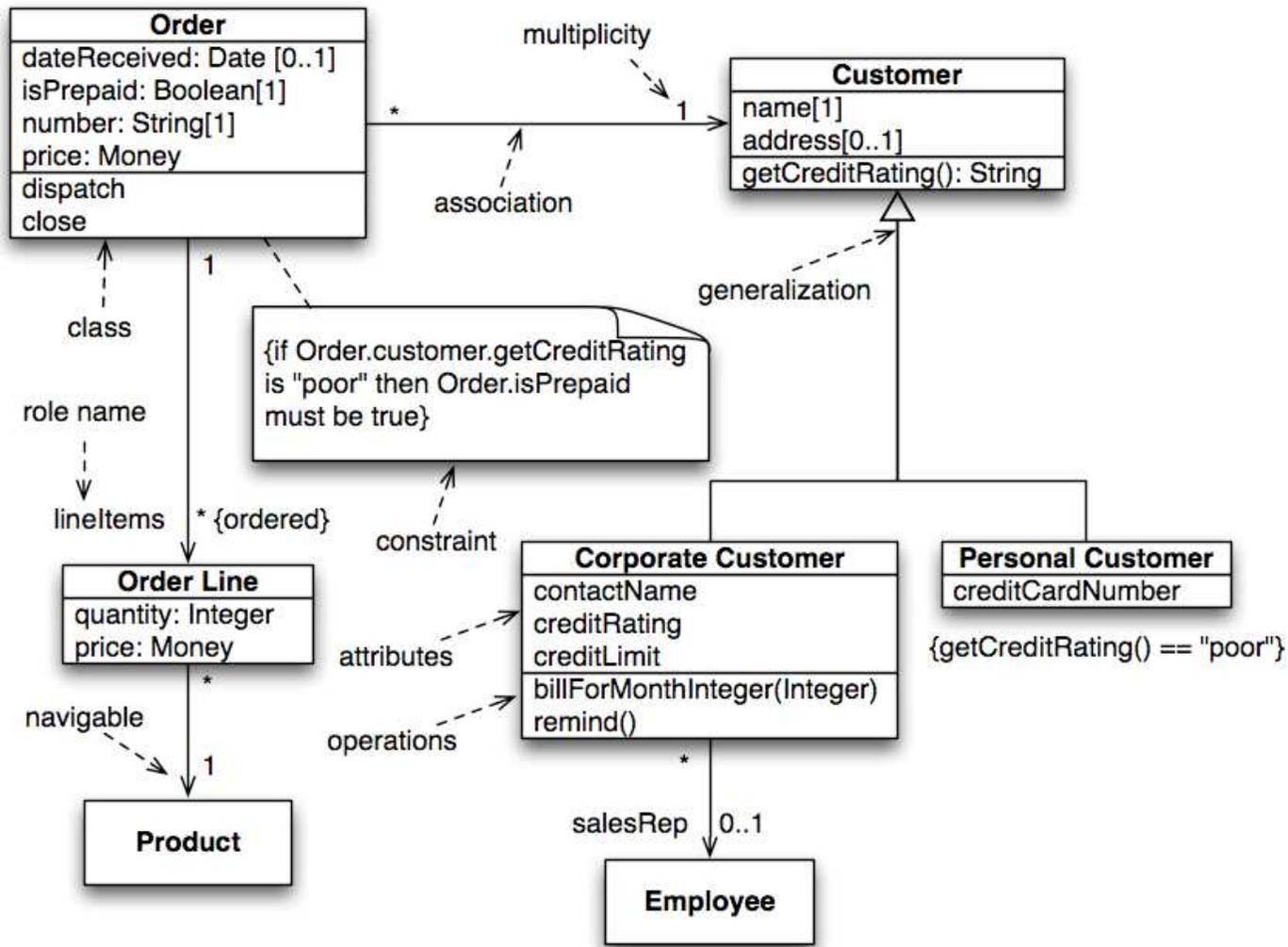
III. DIAGRAMA DE CLASES

Fundamentos



- ❑ Describe el *tipo* de los objetos en el sistema y las diferentes *relaciones* estáticas que existen entre ellos.
- ❑ Muestra los *rasgos* (propiedades y operaciones) de cada clase.
- ❑ Ilustra las *restricciones* que se aplican a la forma en que están conectados los objetos.

Diagrama de clases simple^[2]



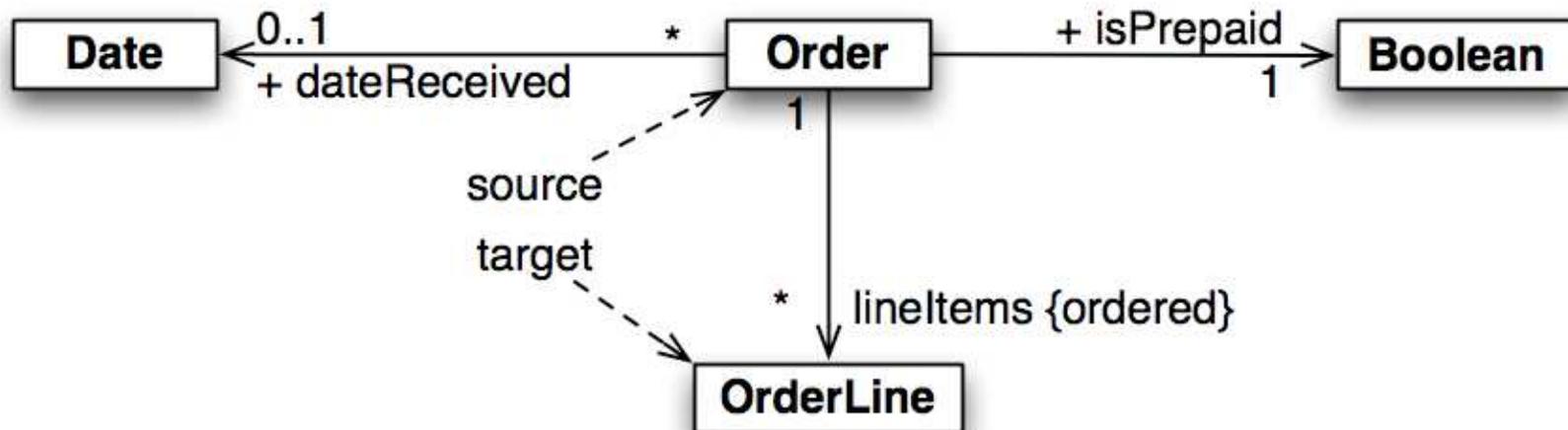
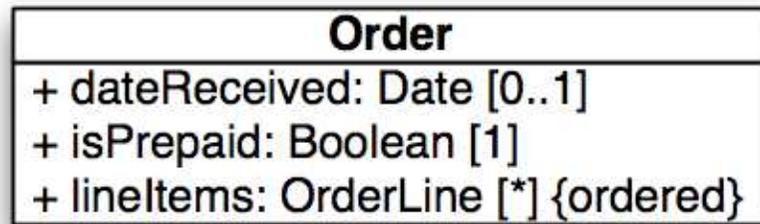


Propiedades como atributos

visibilidad nombre: tipo multiplicidad = default {cadena-propiedad}

- ❑ visibilidad: publico (+), privado (-) o protegido (#).
- ❑ nombre: Cómo se refiere la clase al atributo. Corresponde vagamente a un campo de una estructura.
- ❑ tipo: Indica una restricción sobre qué tipo de objetos pueden colocarse en el atributo. Corresponde al tipo de un campo en una estructura.
- ❑ multiplicidad: Una indicación de a cuántos objetos se refiere el atributo.
- ❑ default value: El valor del atributo para un objeto recién creado si no se especifica un valor para el atributo.
- ❑ {cadena-propiedad}: Especifica propiedades adicionales.

Propiedades como asociaciones



Operaciones



visibilidad nombre (lista-parámetros): tipo {cadena-propiedad}

- ❑ visibilidad: publico (+), privado (-) o protegido (#).
- ❑ nombre: Una cadena de caracteres.
- ❑ lista-parámetros: Es una lista de parámetros para la operación.
- ❑ tipo: Es el tipo del valor regresado por la operación, si lo hay.
- ❑ {cadena-propiedad}: Especifica propiedades adicionales.

dirección nombre: tipo = default

- ❑ dirección: Indica si el parámetro es de entrada, salida o ambos. Si no se especifica dirección, se asume un parámetro de entrada.
- ❑ nombre, tipo, default: Igual que para los atributos.

Sistema de información para el registro de órdenes de compra y clientes



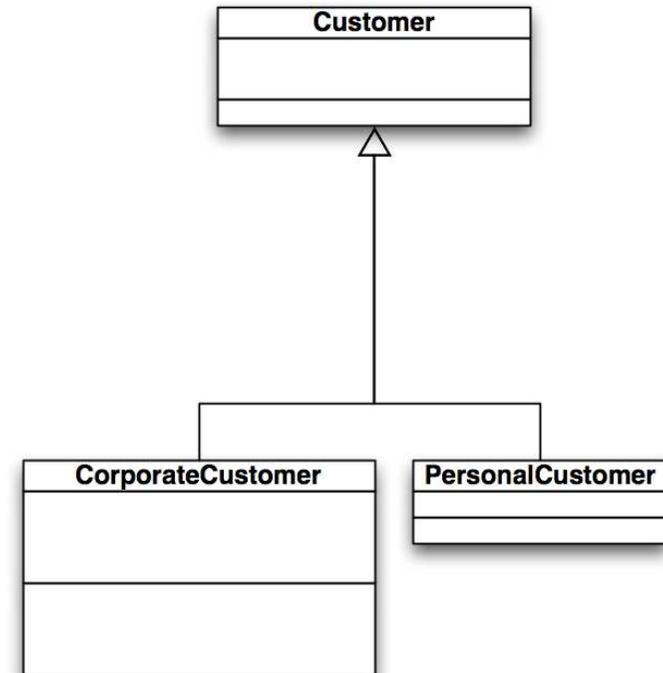
- Requerimientos:
 1. Debe llevar registro de clientes corporativos y de clientes que sean personas físicas.

Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

1. Debe llevar registro de clientes corporativos y de clientes que sean personas físicas.
2. La información general de los clientes incluye su nombre (o razón social) y su dirección.

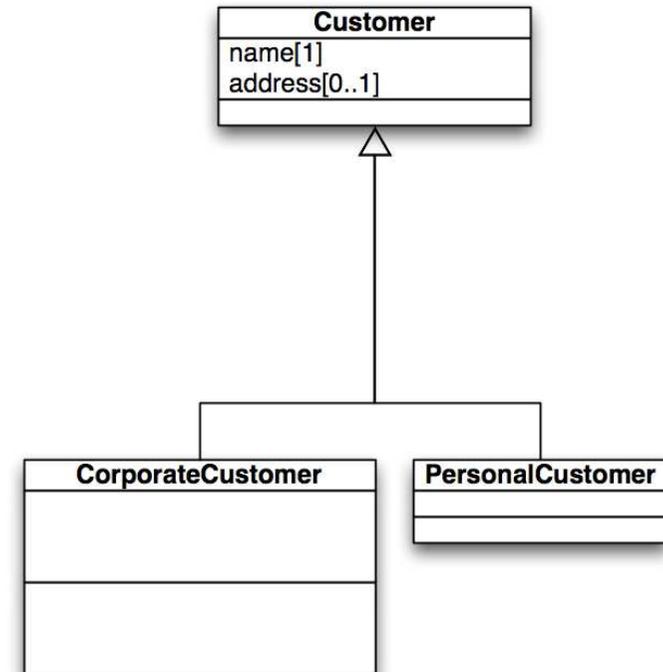


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

1. Debe llevar registro de clientes corporativos y de clientes que sean personas físicas.
2. La información general de los clientes incluye su nombre (o razón social) y su dirección.
3. Para ambos tipos de cliente debe ser posible obtener su calificación de crédito.

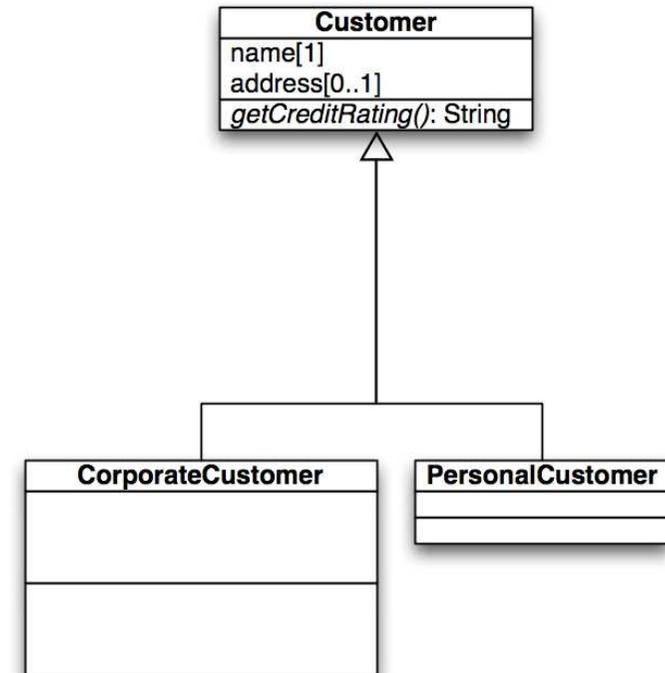


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

1. Debe llevar registro de clientes corporativos y de clientes que sean personas físicas.
2. La información general de los clientes incluye su nombre (o razón social) y su dirección.
3. Para ambos tipos de cliente debe ser posible obtener su calificación de crédito.
4. La información de los clientes corporativos incluye datos de la persona de contacto, la calificación de crédito y su límite de crédito.

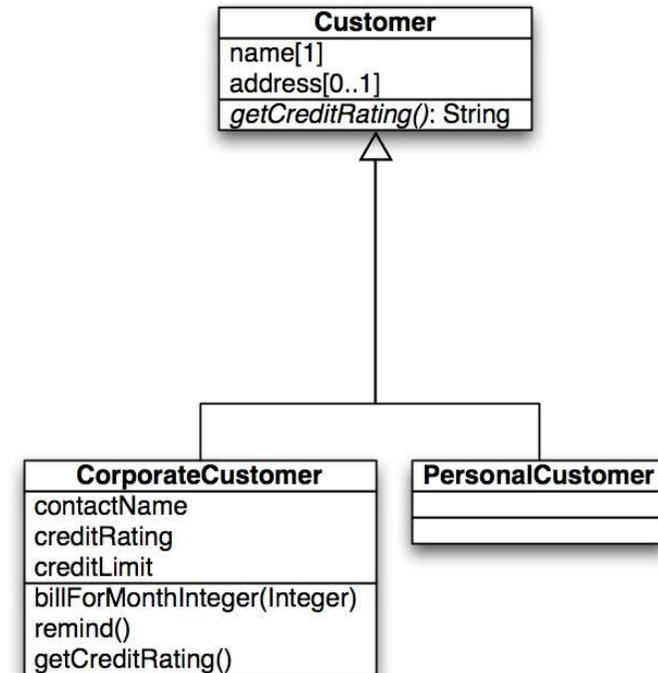


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

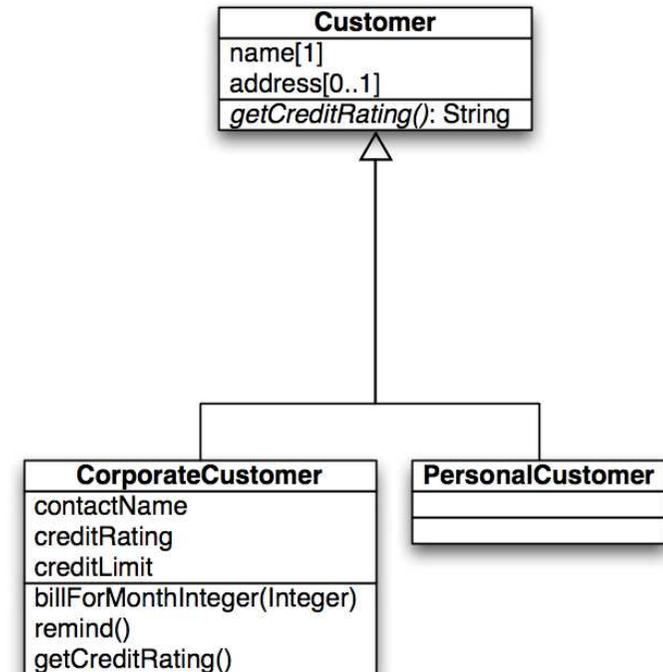
1. Debe llevar registro de clientes corporativos y de clientes que sean personas físicas.
2. La información general de los clientes incluye su nombre (o razón social) y su dirección.
3. Para ambos tipos de cliente debe ser posible obtener su calificación de crédito.
4. La información de los clientes corporativos incluye datos de la persona de contacto, la calificación de crédito y su límite de crédito.



Sistema de información para el registro de órdenes de compra y clientes



- Requerimientos:
 - El sistema debe almacenar el número de tarjeta de crédito de los clientes que sean personas físicas.

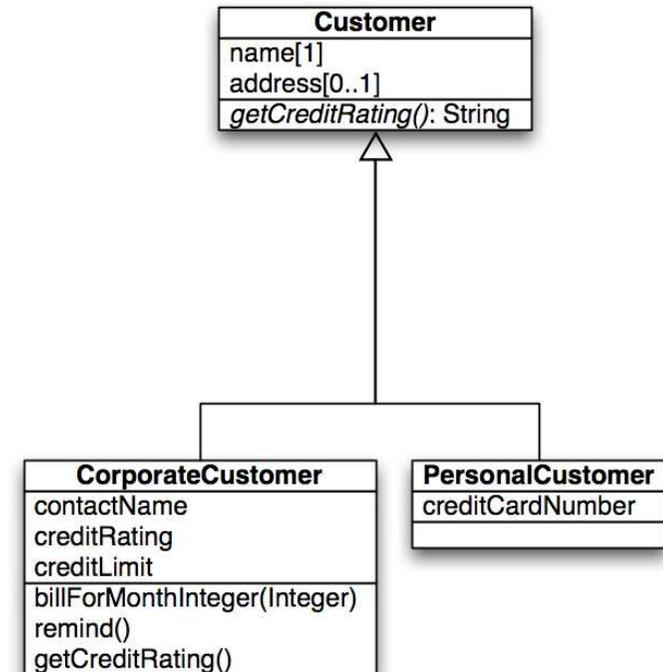


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

5. El sistema debe almacenar el número de tarjeta de crédito de los clientes que sean personas físicas.
6. La calificación de crédito de una persona física es limitada (para la mayoría de ellas al menos).

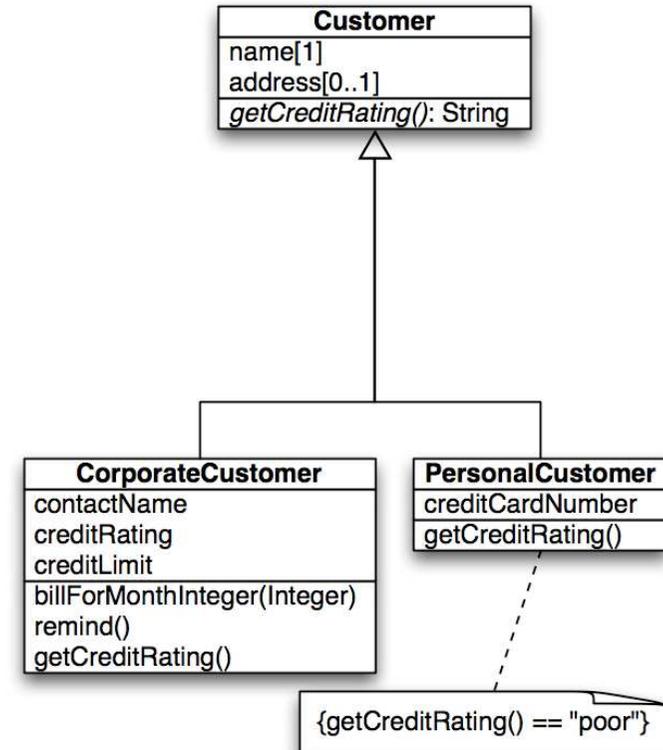


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

5. El sistema debe almacenar el número de tarjeta de crédito de los clientes que sean personas físicas.
6. La calificación de crédito de una persona física es limitada (para la mayoría de ellas al menos).
7. El sistema debe llevar el registro de los agentes de ventas asignados a los clientes corporativos como representantes.

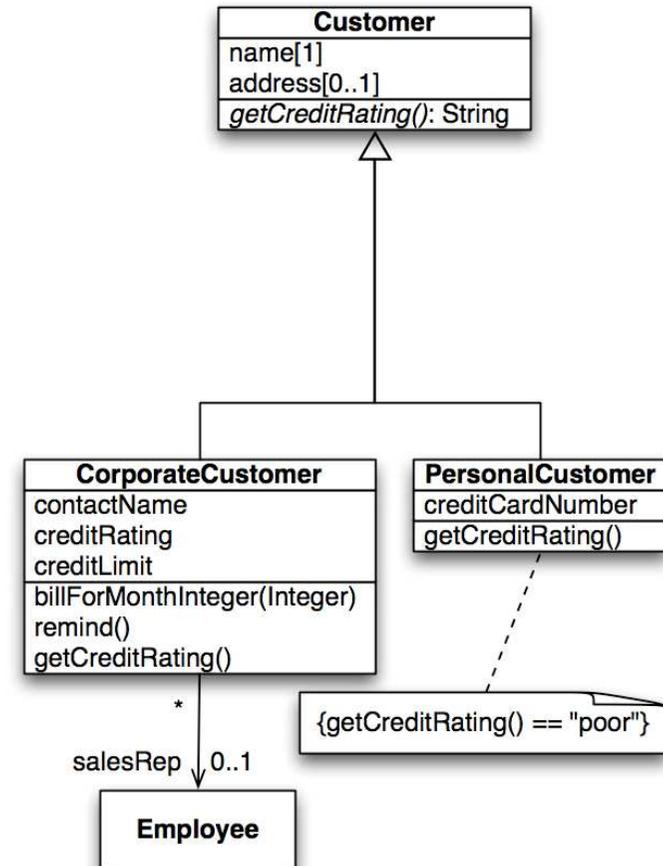


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

5. El sistema debe almacenar el número de tarjeta de crédito de los clientes que sean personas físicas.
6. La calificación de crédito de una persona física es limitada (para la mayoría de ellas al menos).
7. El sistema debe llevar el registro de los agentes de ventas asignados a los clientes corporativos como representantes.

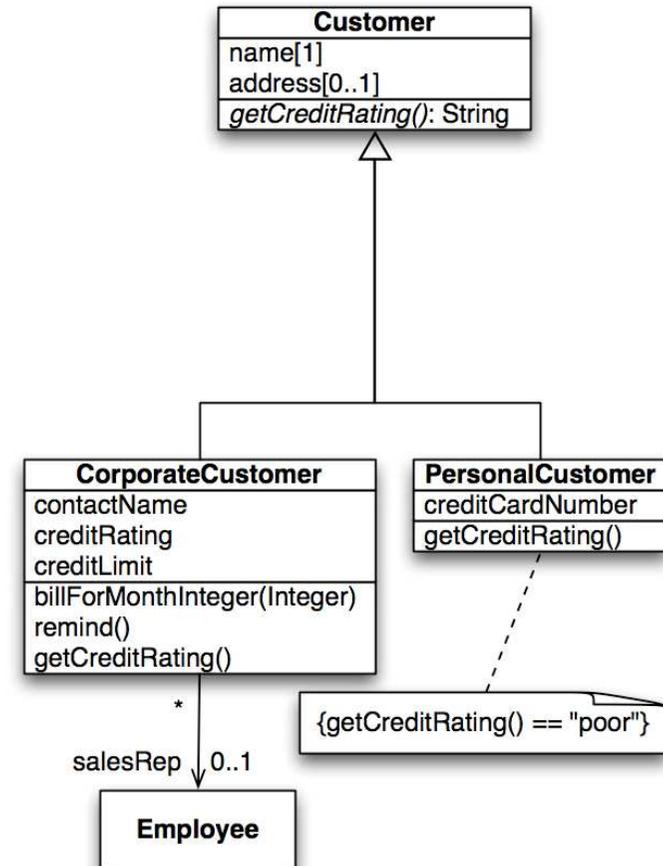


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

- El sistema también registra los productos que la compañía fabrica, los cuales son comercializados mediante órdenes de compra.

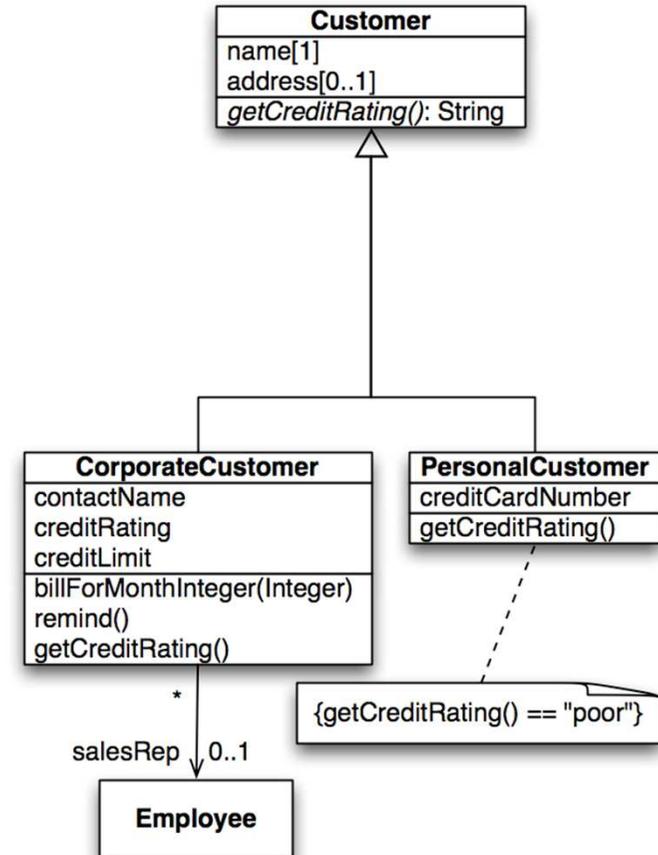
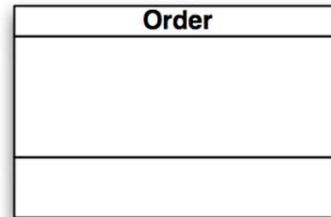


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

- 8. El sistema también registra los productos que la compañía fabrica, los cuales son comercializados mediante órdenes de compra.
- 9. Una orden de compra debe contener la información de la fecha en que se expide, su folio y el monto total del pedido.

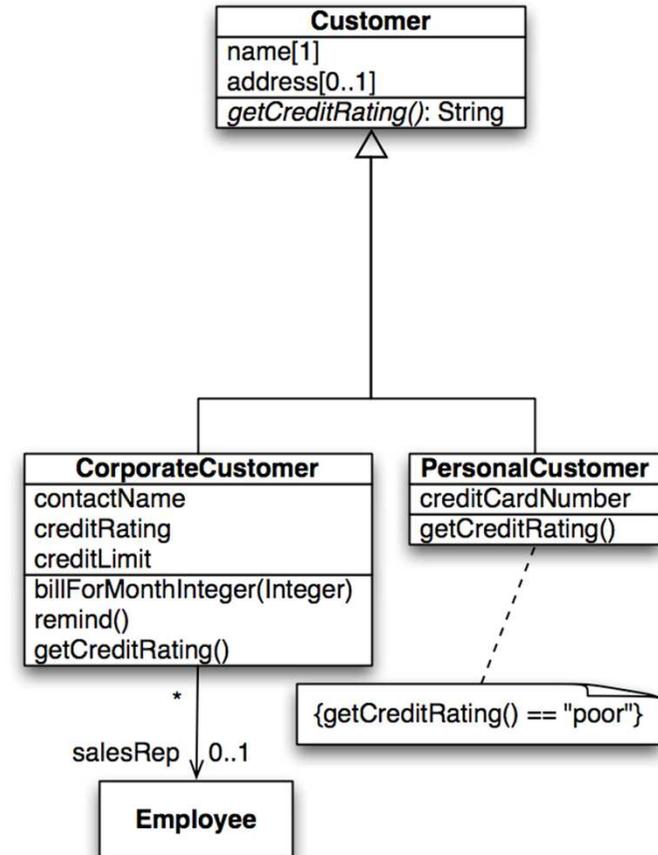
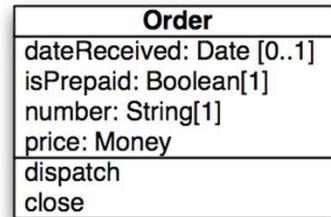


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

8. El sistema también registra los productos que la compañía fabrica, los cuales son comercializados mediante órdenes de compra.
9. Una orden de compra debe contener la información de la fecha en que se expide, su folio y el monto total del pedido.
10. Cada cliente puede haber colocado varias órdenes de compra. Y cada orden de compra (identificada por su folio) ha sido colocada por un único cliente.

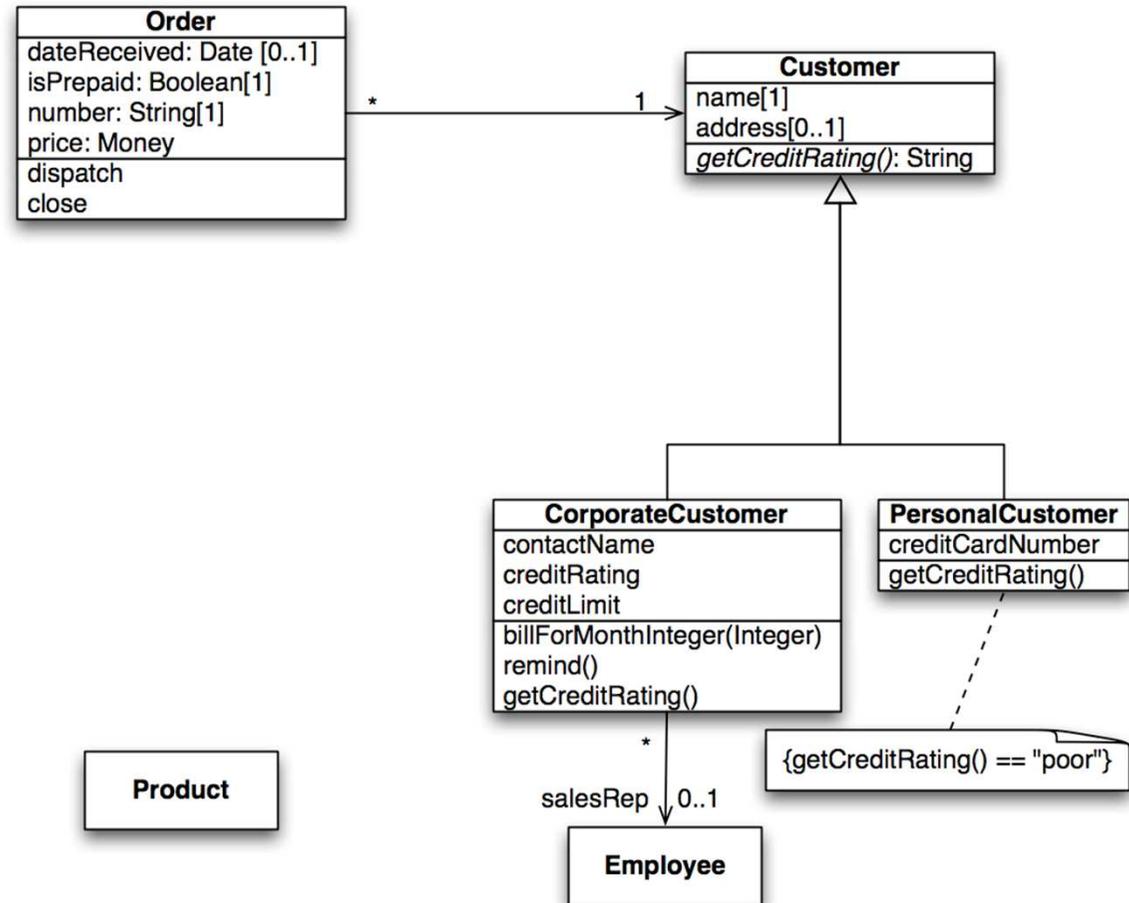


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

8. El sistema también registra los productos que la compañía fabrica, los cuales son comercializados mediante órdenes de compra.
9. Una orden de compra debe contener la información de la fecha en que se expide, su folio y el monto total del pedido.
10. Cada cliente puede haber colocado varias órdenes de compra. Y cada orden de compra (identificada por su folio) ha sido colocada por un único cliente.

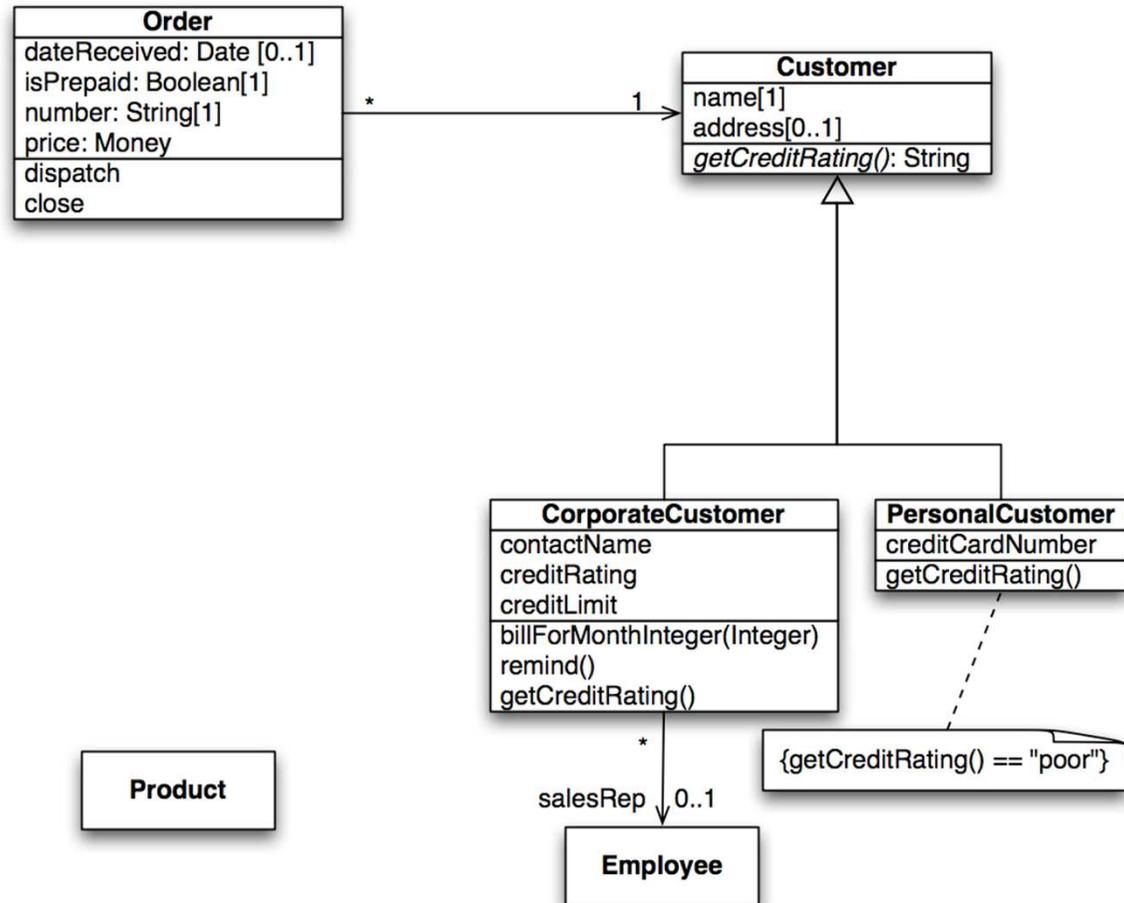


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

- 11. Una entrada de la orden de compra contiene información sobre la cantidad de producto adquirido, la descripción de tal producto y el costo por esa cantidad de producto.

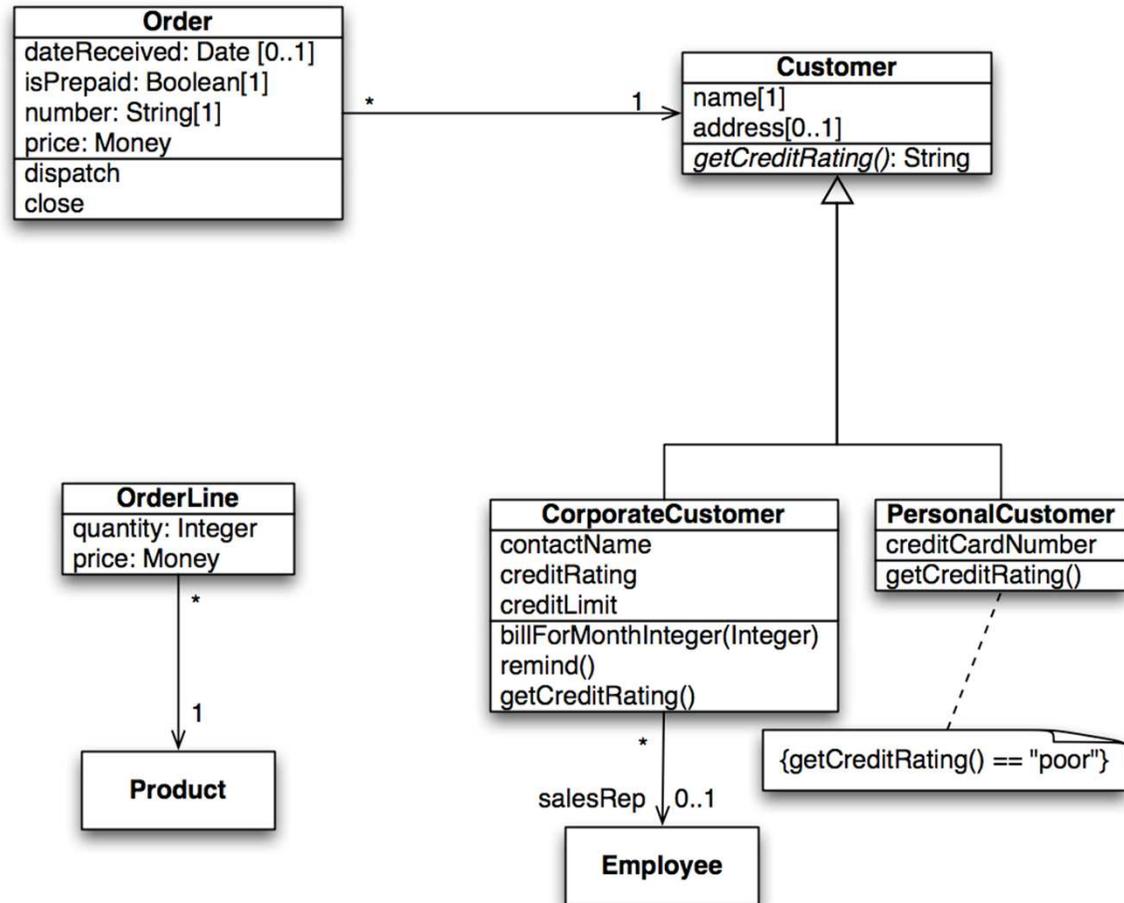


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

- Una entrada de la orden de compra contiene información sobre la cantidad de producto adquirido, la descripción de tal producto y el costo por esa cantidad de producto.
- Cada orden de compra puede contener un conjunto de varias entradas ordenadas, cada una con la información correspondiente a un producto específico.

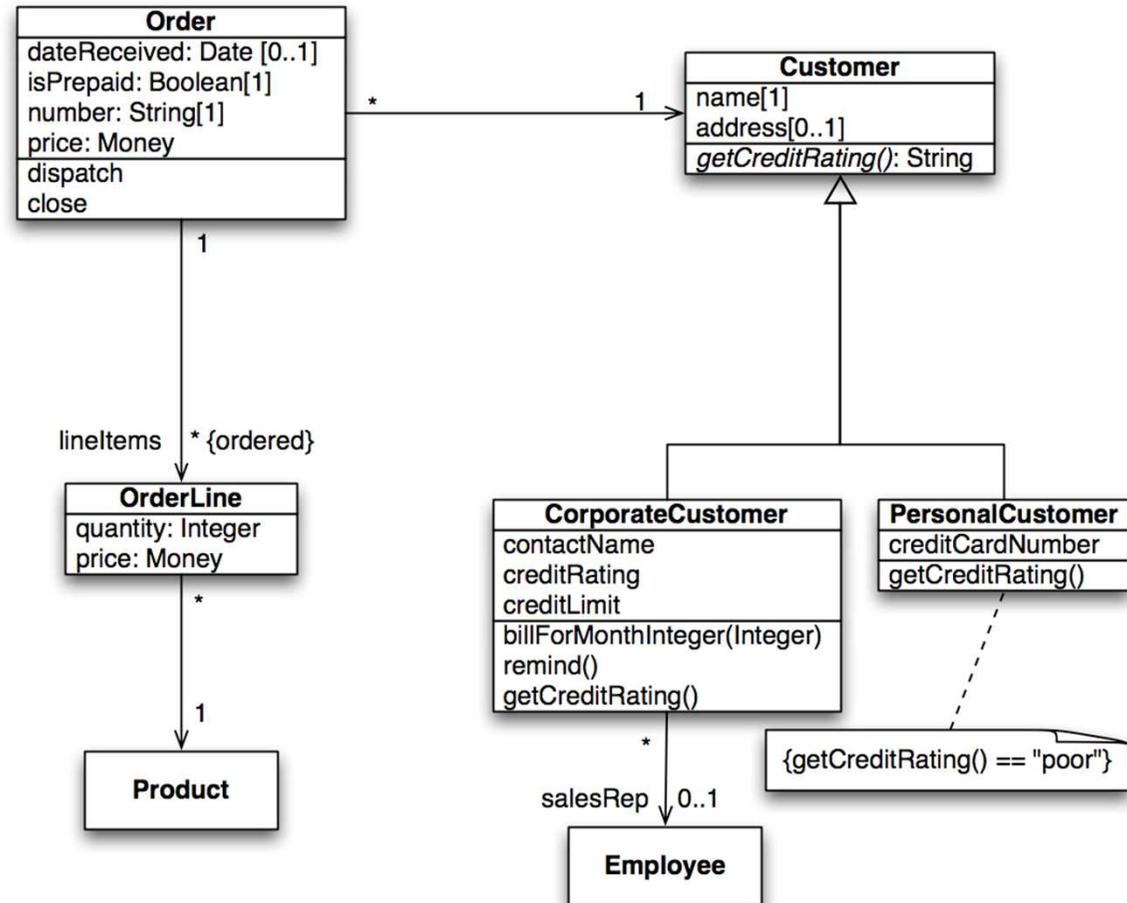


Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

11. Una entrada de la orden de compra contiene información sobre la cantidad de producto adquirido, la descripción de tal producto y el costo por esa cantidad de producto.
12. Cada orden de compra puede contener un conjunto de varias entradas ordenadas, cada una con la información correspondiente a un producto específico.
13. Cuando la calificación de crédito del cliente no es buena el sistema debe registrar que el pago por la orden debe ser anticipado.



Sistema de información para el registro de órdenes de compra y clientes



Requerimientos:

11. Una entrada de la orden de compra contiene información sobre la cantidad de producto adquirido, la descripción de tal producto y el costo por esa cantidad de producto.
12. Cada orden de compra puede contener un conjunto de varias entradas ordenadas, cada una con la información correspondiente a un producto específico.
13. Cuando la calificación de crédito del cliente no es buena el sistema debe registrar que el pago por la orden debe ser anticipado.

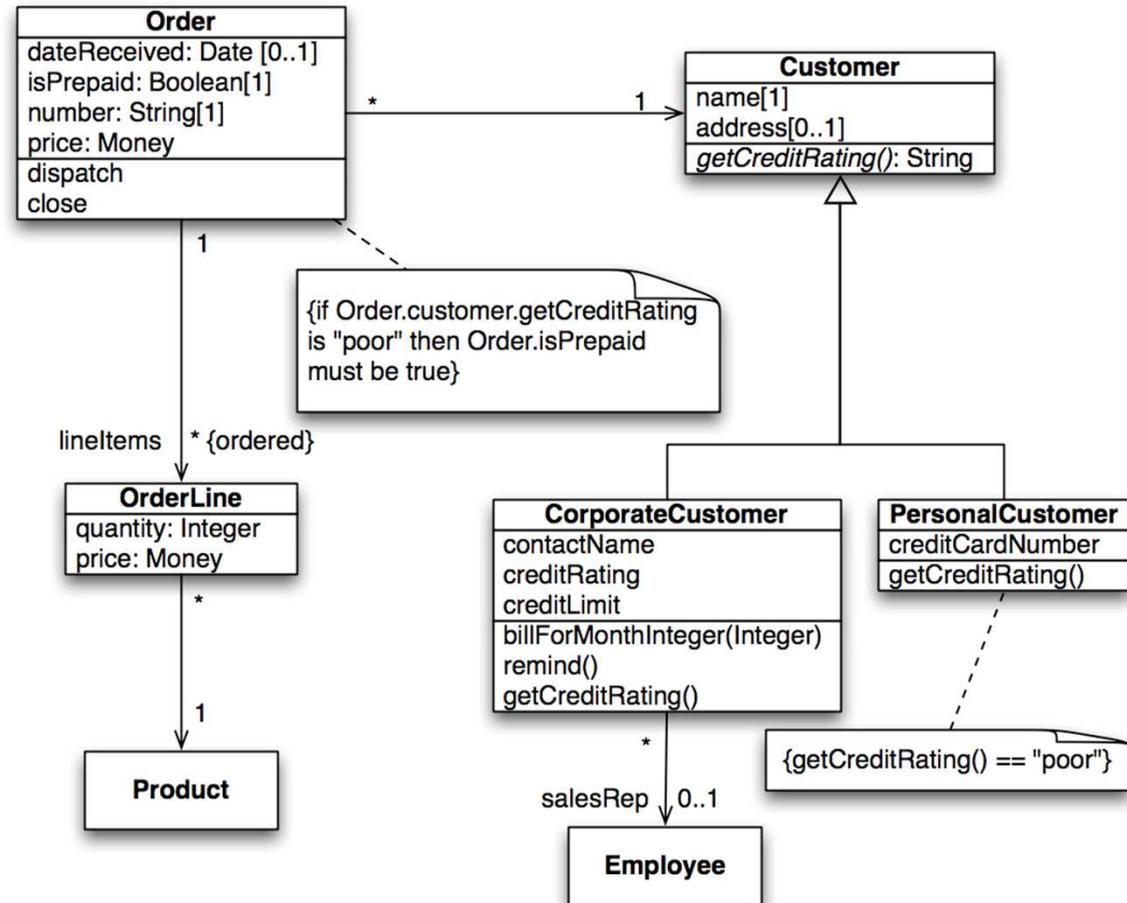
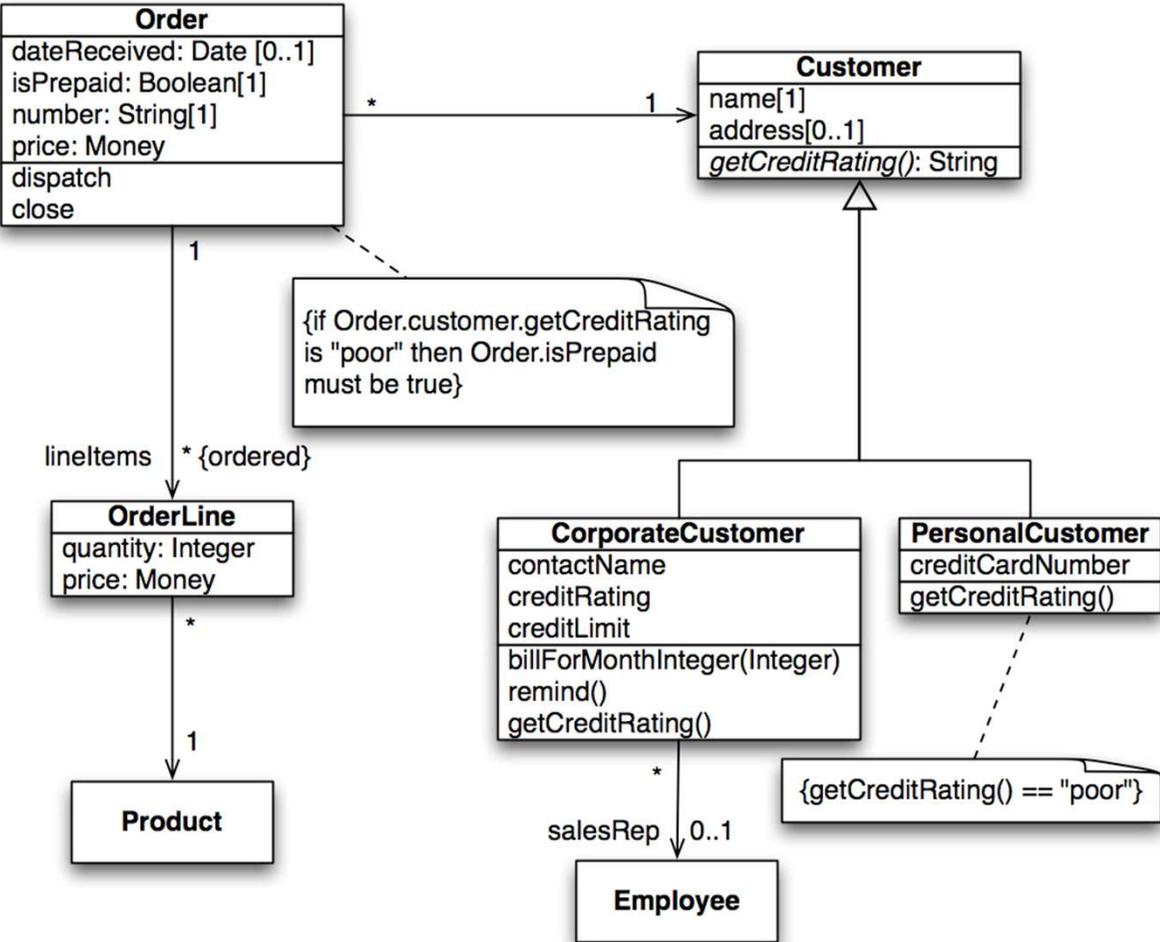
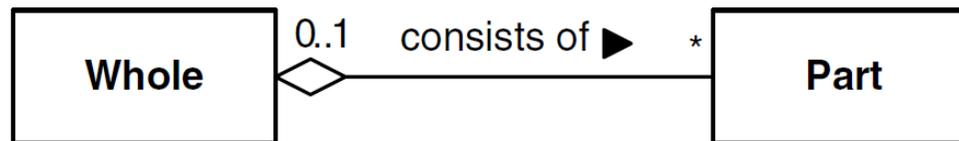
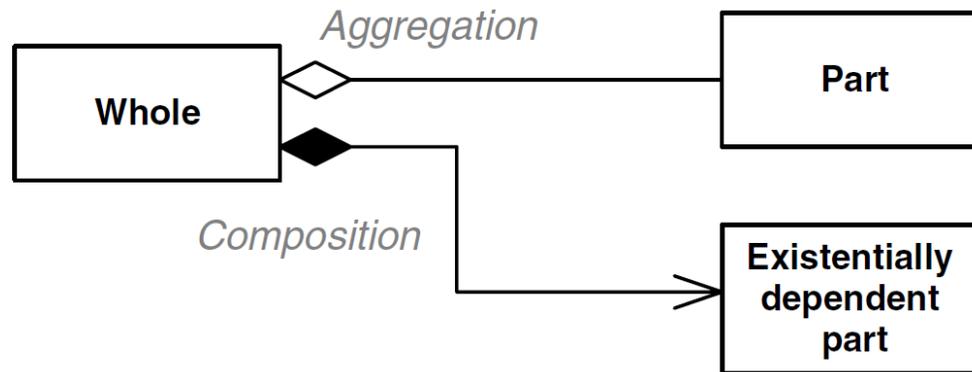


Diagrama de clases del diseño



Relaciones^[4]

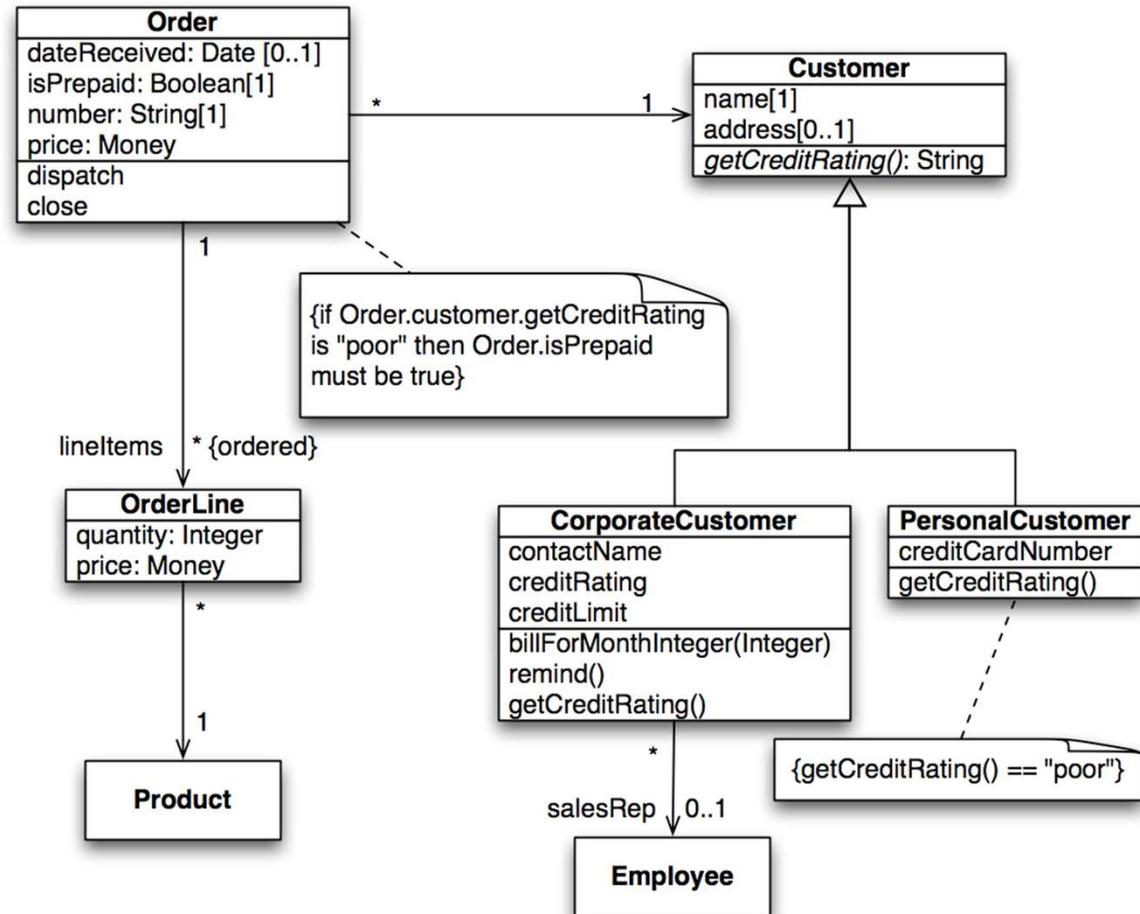


Sistema de información para el registro de órdenes de compra, clientes, empleados y proyectos



Requerimientos:

1. El sistema de información debe llevar un registro de todos los empleados. Los puestos que pueden ocupar son: gerente, ingeniero de diseño, ingeniero de campo, representante de ventas y administrativo.
2. La empresa diseña y manufactura varios tipos de computadoras, teléfonos móviles, reproductores de música y el sistema de información debe llevar un registro de las especificaciones de cada producto.

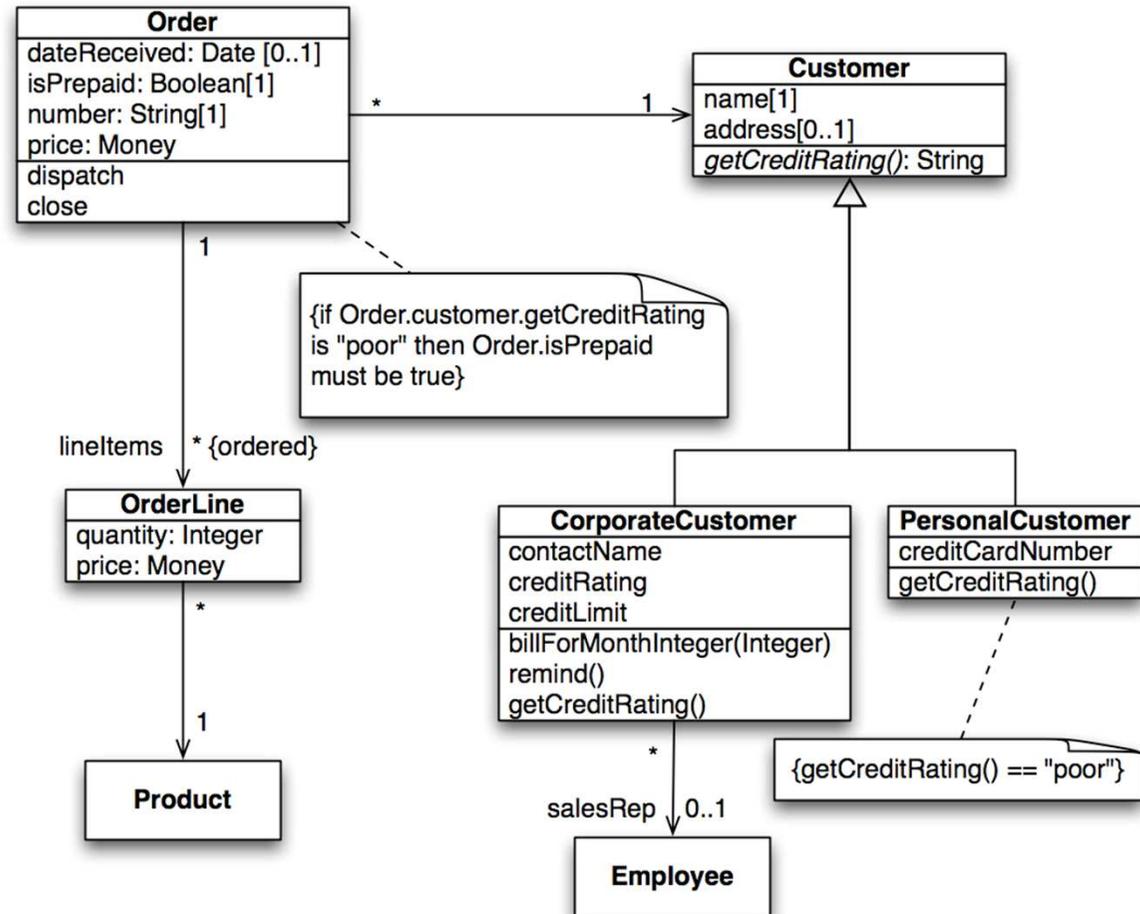


Sistema de información para el registro de órdenes de compra, clientes, empleados y proyectos



Requerimientos:

- 3. El sistema de información debe llevar un registro de qué ingenieros de diseño están asignados al desarrollo de cada producto.
- 4. El sistema de información debe llevar un registro de qué ingenieros de campo proporcionan servicios de consultoría a cada cliente corporativo.
- 5. La compañía está organizada en departamentos de R&D, cada uno de ellos con su gerente y su equipo de ingenieros de diseño.





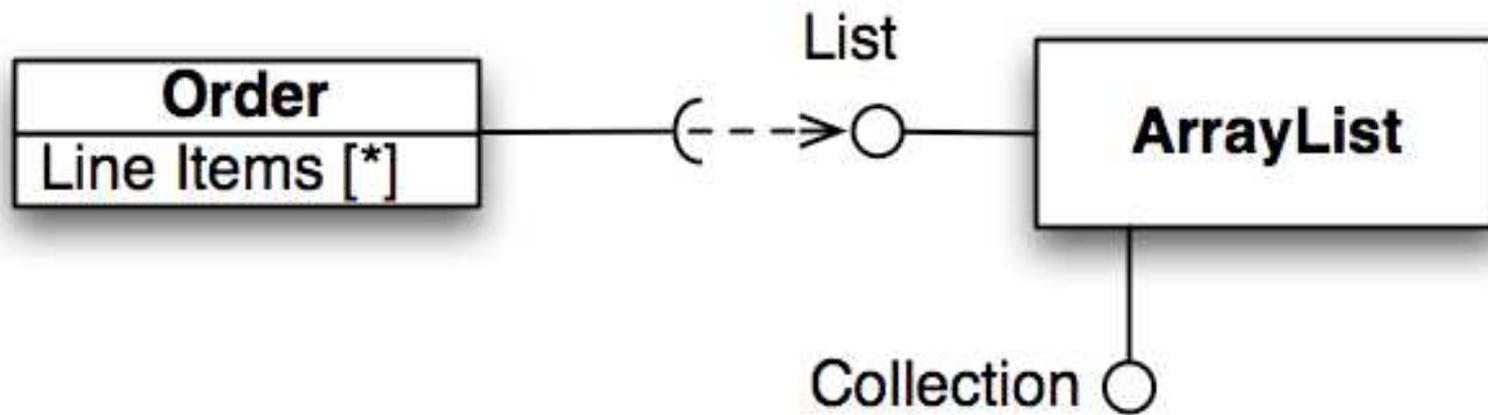
IV. INTERFACES

Fundamentos

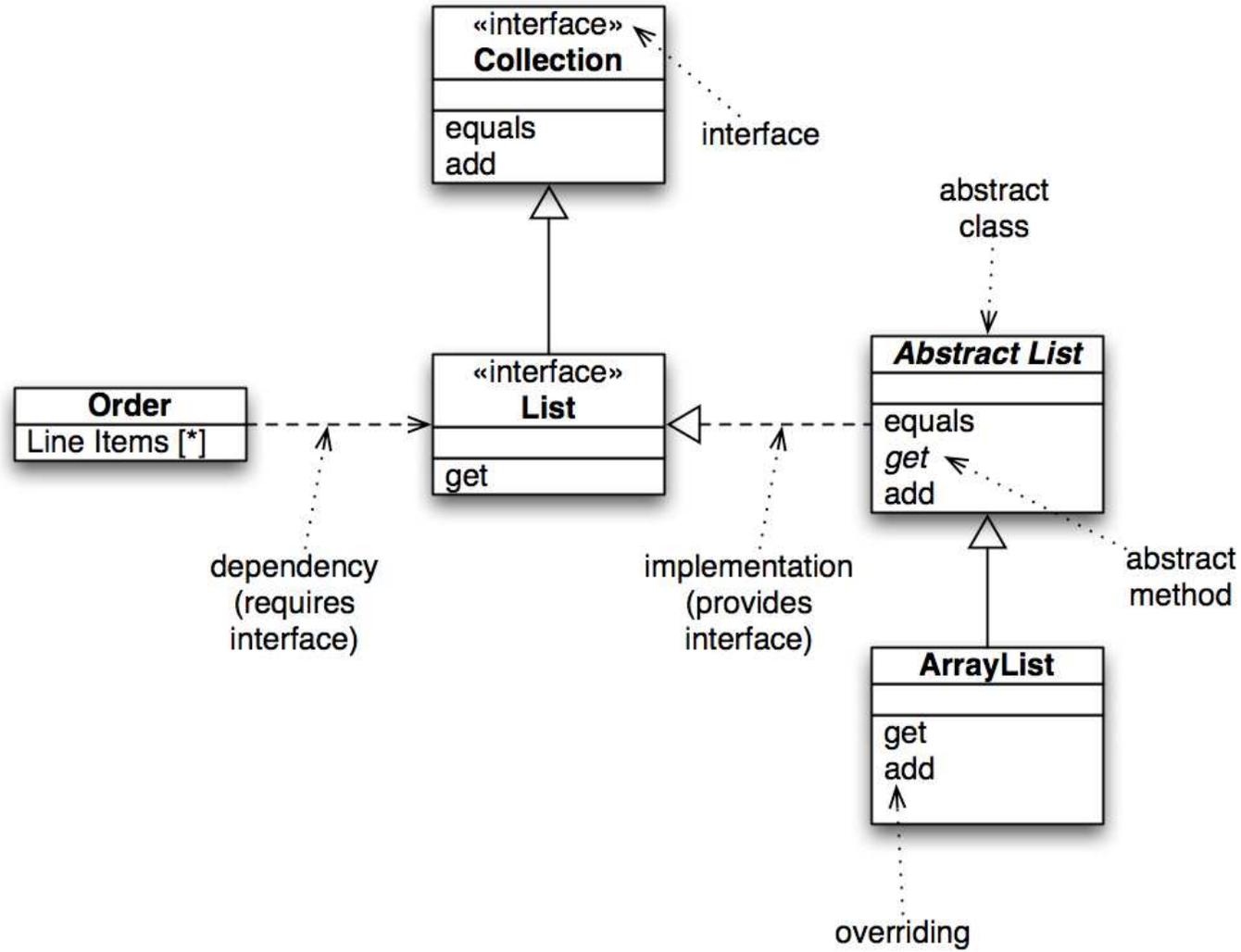


- ❑ Una interfaz es una colección de operaciones que representan el comportamiento completo de una clase o componente, o sólo una parte de ese comportamiento.
- ❑ Una interfaz especifica un conjunto de operaciones (prototipos), pero nunca un conjunto de implementaciones.
- ❑ Una interfaz raramente se encuentra aislada.

Notación^[2]



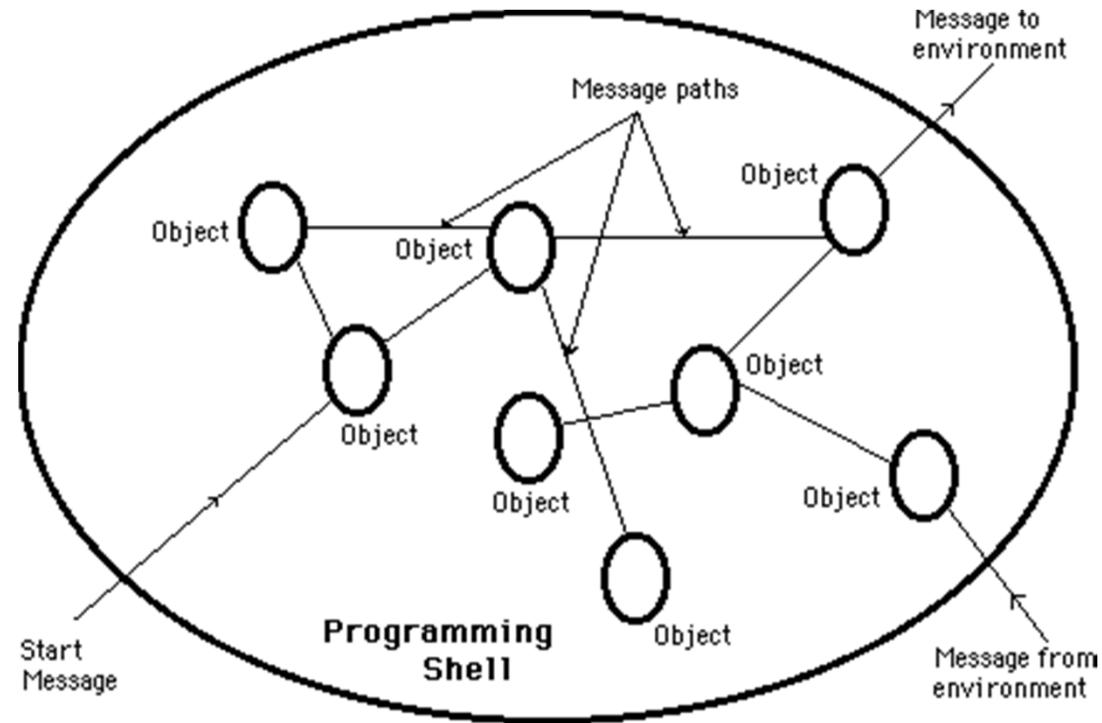
Ejemplo^[2]





V. DIAGRAMA DE SECUENCIA

Mensajes



- ❑ Los objetos son entidades de software que colaboran y se envían mensajes entre ellos.

Fundamentos



- ❑ Es un diagrama que ilustra la interacción entre objetos.
- ❑ Consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes enviados entre ellos.
- ❑ Destaca la ordenación temporal de los mensajes entre los objetos que interaccionan.

Problema: Modelar el proceso de cálculo del precio total de una orden de compra



1. La instancia de la clase Order contiene una operación que al invocarla inicie el proceso de cálculo.
2. Para cada entrada en la orden de compra (instancia de OrderLine) se necesita calcular el costo total por producto, este se calcula en base a:
 - La cantidad de producto solicitado.
 - El precio unitario de tal producto.
3. La instancia de la clase Order suma los costos de todos los productos en la orden y realiza otros posibles cálculos para determinar el precio base.
4. El precio final incluye un descuento que se aplica de forma distinta dependiendo del tipo de cliente. La instancia de la clase Order consulta la información sobre el cliente para determinar el monto de descuento y entonces calcular el costo total de la orden.

Diagrama de clases extendido

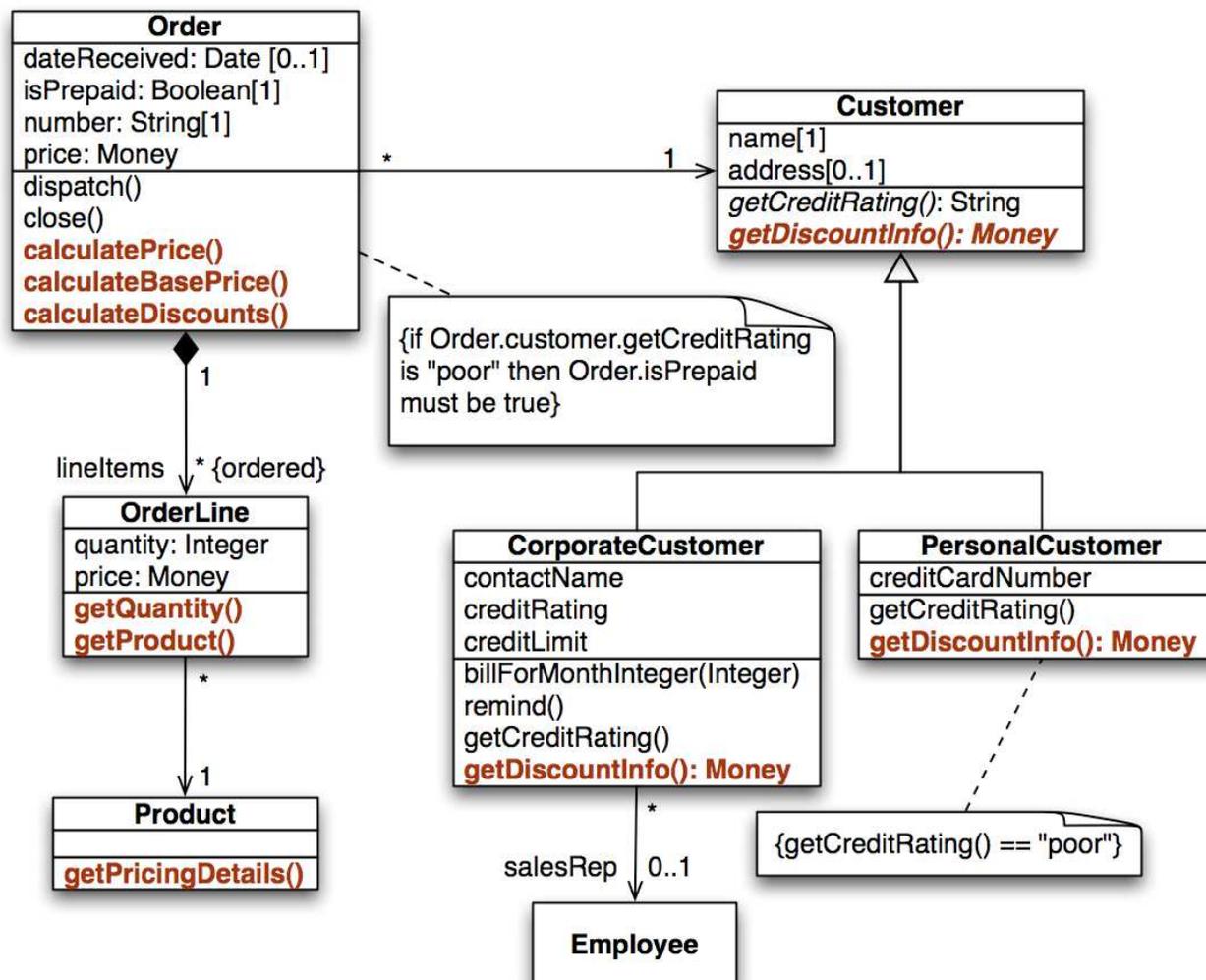
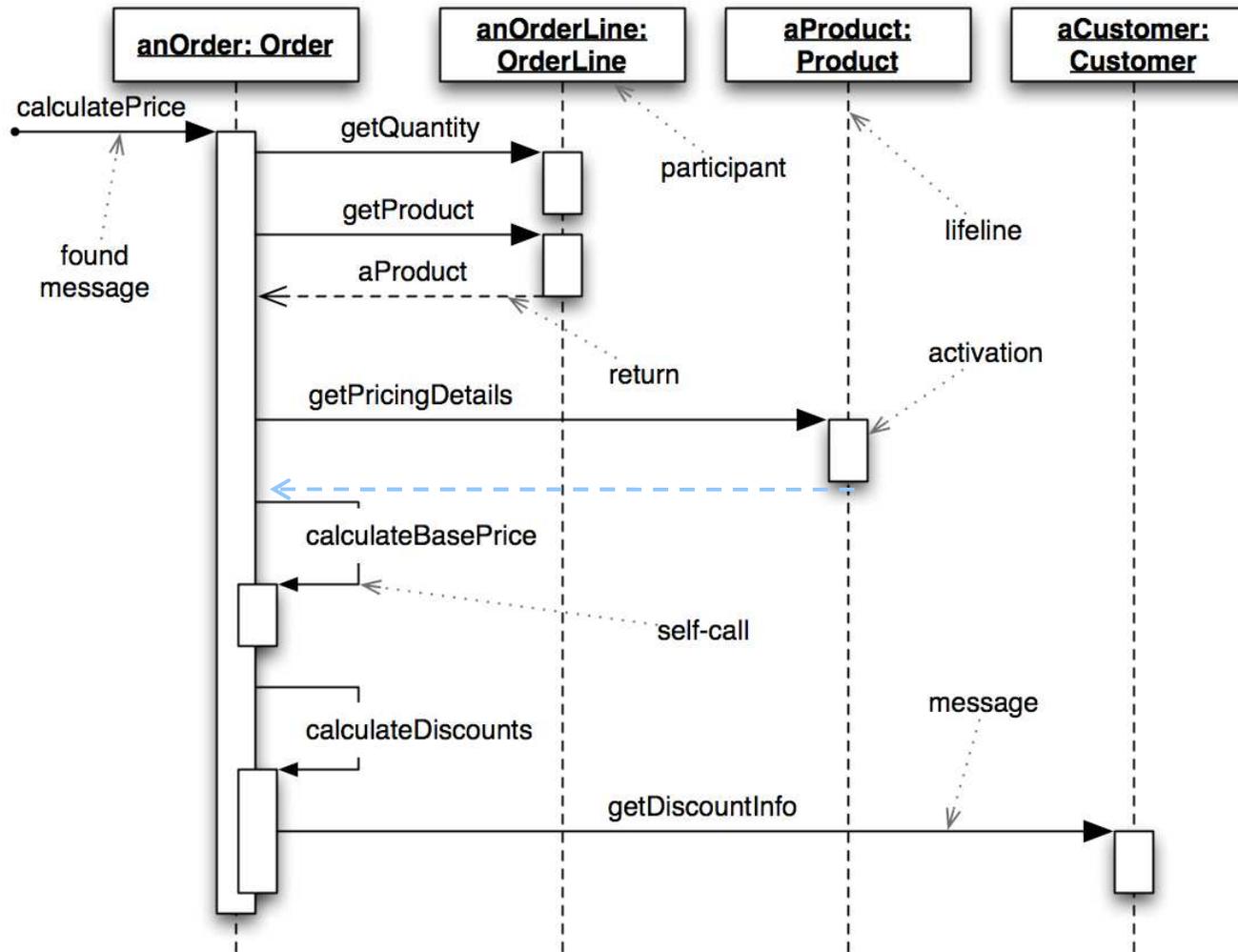


Diagrama de secuencia para el problema

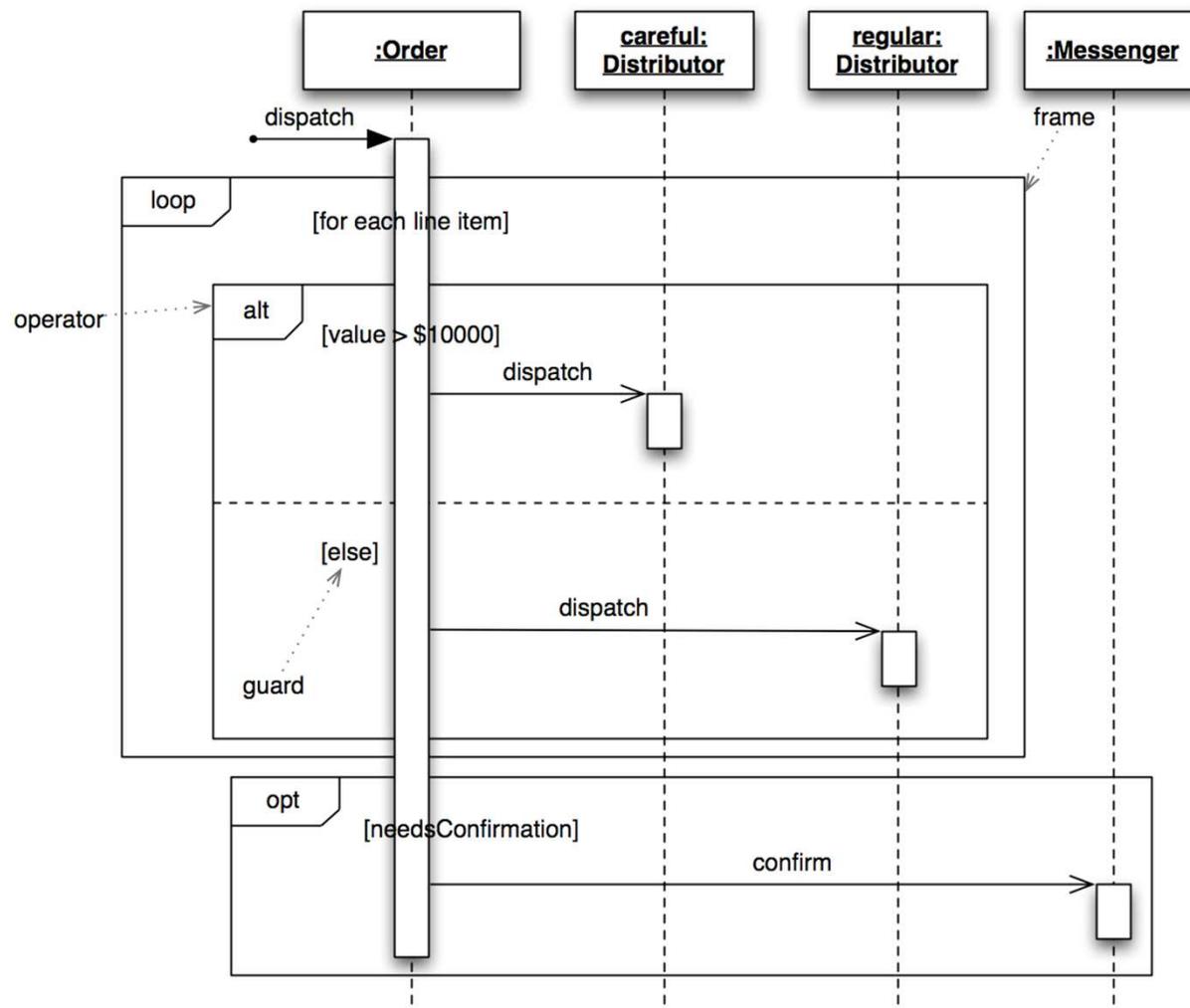


Ejercicio



- La clase Order tiene un método llamado dispatch() que inicia el proceso de entrega de los productos que conforman una orden de compra de acuerdo a los siguientes criterios:
 - Si el costo total por la cantidad de producto es menor que el límite de \$10,000 entonces se indica a un distribuidor regular que haga el reparto del producto, bajo medidas estándar de seguridad.
 - Si el costo total por la cantidad de producto rebasa el límite de \$10,000 entonces se indica a un distribuidor especial (diferente al anterior) que haga el reparto del producto, bajo medidas más rígidas de seguridad.

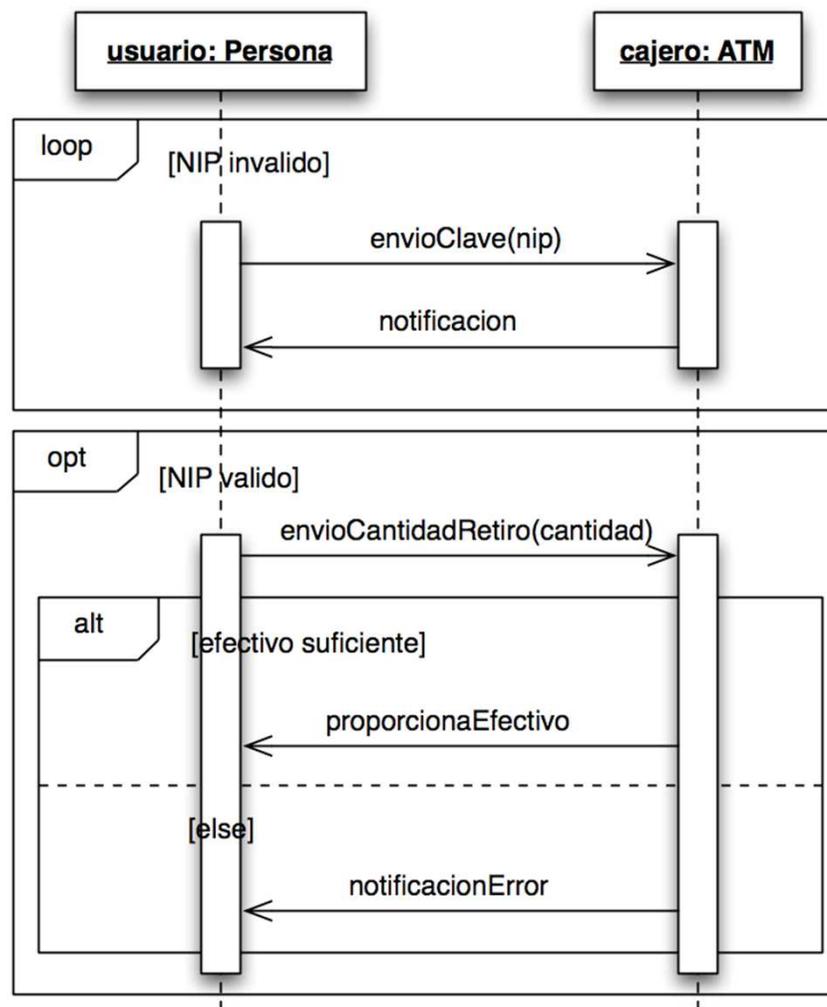
- La empresa tiene un mensajero que, de ser necesario, confirma que los pedidos de cada producto hayan sido despachados de forma correcta.



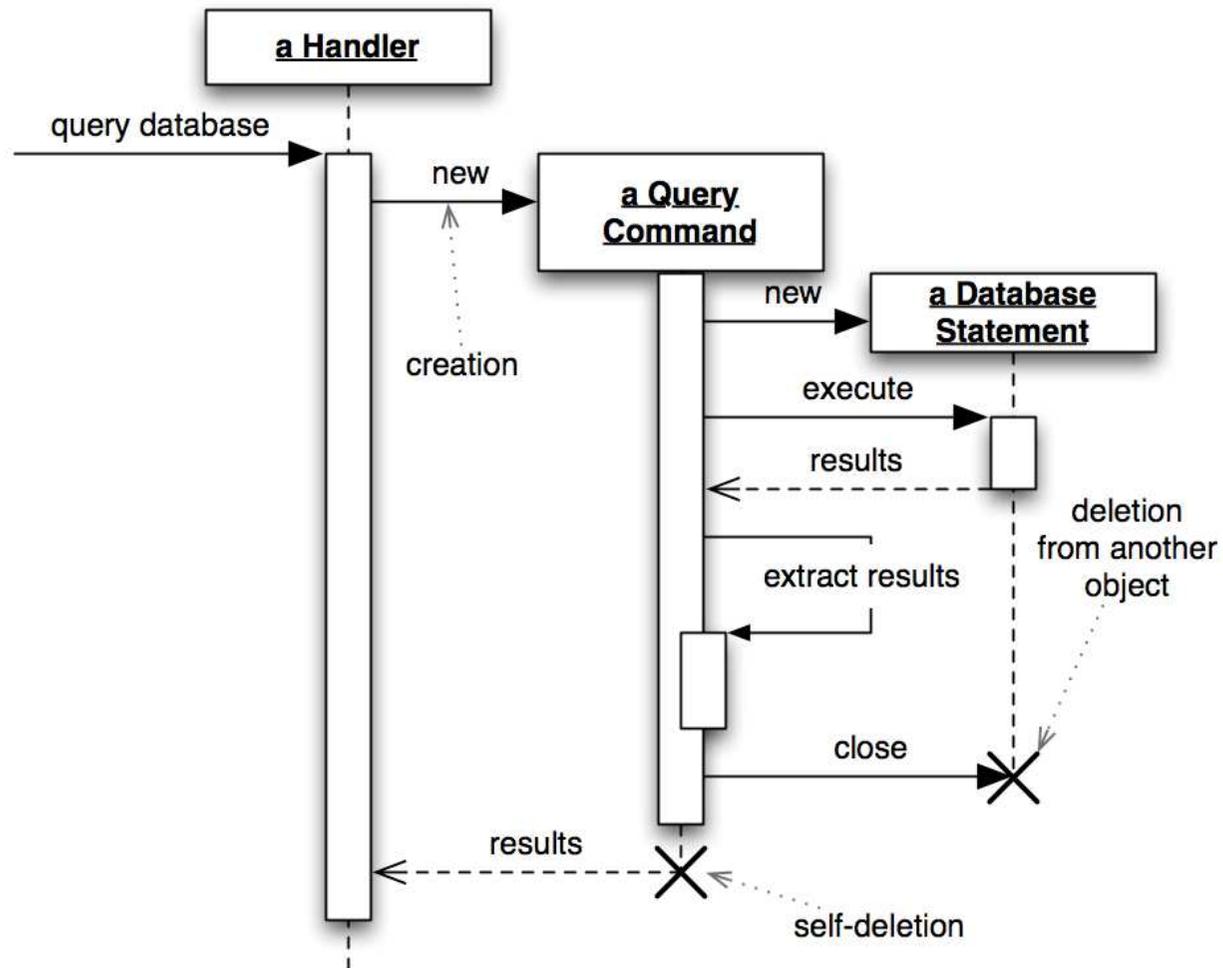
Ejercicio



- Modele, mediante un sencillo diagrama de secuencia en **UML 2.0**, la interacción entre un usuario (instancia de la clase Persona) y un cajero electrónico (instancia de la clase ATM):
 - Al principio, el diagrama debe ilustrar que el usuario proporciona su NIP al cajero y que este le informa si el NIP es válido o no. El diagrama debe ilustrar que este proceso se realiza de forma iterativa mientras el NIP ingresado sea inválido. Por simplicidad asuma que el cajero es capaz de recibir el NIP del usuario un número indeterminado de veces.
 - A continuación, el diagrama debe ilustrar la interacción en la cual el usuario indica la cantidad a retirar y la interacción en la que el cajero entrega el efectivo (si es que cuenta con efectivo suficiente) o un mensaje de error (en caso de que no haya suficiente efectivo). Todas las interacciones en este punto se realizan únicamente si el NIP ingresado por el usuario anteriormente es válido.



Creación dinámica de instancias



Procesamiento de cheques en un sistema de información para banco



- ❑ El **módulo principal** de un sistema de información bancario crea una instancia de la clase **Cheque** cuando algún cajero introduce los datos de un cheque de dicho banco que ha recibido de algún cliente portador. Estos datos incluyen el folio, la fecha de expedición, el número de cuenta de la persona que expidió el cheque y el monto del cheque.
- ❑ El **módulo gestor de cuentas** entra en acción para verificar que el cliente que expidió el cheque dispone de fondos suficientes, mediante una consulta a la **cuenta del cliente que expidió el cheque**.
- ❑ Si el balance en la cuenta anterior es mayor o igual al monto del cheque entonces el gestor resta le resta esta cantidad de la cuenta del titular y después se la suma a la **cuenta del cliente que porta el cheque**.
- ❑ Si el balance en la cuenta de la persona que expidió el cheque es menor al monto del cheque entonces el gestor le aplica una cuota de penalización al titular a través de la cuenta y después le envía un correo electrónico informándole de la situación.



VI. DIAGRAMA DE CASOS DE USO

Escenario



- Un escenario es una secuencia de pasos que describe una interacción entre un usuario y un sistema.

Escenario para una tienda en línea



- ❑ *El cliente revisa el catálogo y añade los artículos deseado a al carrito de compra. Cuando el cliente desea pagar indica la información de su tarjeta de crédito, así como la dirección de envío, y confirma la venta. El sistema verifica la autorización de la tarjeta y confirma la venta tanto inmediatamente y por correo electrónico.*

- ❑ Variantes del mismo escenario:
 - ¿Qué pasa si la autorización de la tarjeta de crédito falla?
 - ¿Es posible que el sistema de la tienda en línea pueda almacenar los datos de usuarios frecuentes?

- ❑ Meta de estos escenarios comunes: comprar un producto.

Casos de uso



- ❑ Un caso de uso es un conjunto de escenarios enlazados juntos por una meta del usuario común.
- ❑ UML describe la construcción de un diagrama de casos de uso, que muestra cómo los casos de uso se relacionan unos con otros.
- ❑ Casi siempre el valor de los casos de uso está en la descripción de su contenido puesto que un diagrama sin descripción es de valor limitado.

Actores



- ❑ A los usuarios se les denomina actores, que representan roles que los usuarios juegan con respecto al sistema.
- ❑ Ejemplos de actores:
 - Cliente.
 - Representante de servicio a cliente.
 - Analista de producto.
- ❑ Los actores llevan a cabo los casos de uso.
- ❑ Un actor puede participar en varios casos de uso; de manera inversa, cada caso de uso puede involucrar a varios actores.
- ❑ Un actor puede representar a una persona o a otro tipo de entidad.
- ❑ El término más adecuado no es actor, sino rol.

Contenido de un caso de uso



- ❑ No hay una forma estándar de redactar los escenarios y se aceptan diferentes formatos.
- ❑ Se puede comenzar seleccionando uno de los escenarios del caso de uso como el escenario exitoso principal (MSS), el cual se redacta como una serie de pasos numerados.
- ❑ Después se pueden redactar las extensiones, descritas en términos de variaciones al escenario exitoso principal.
- ❑ Cada caso de uso tiene asociado un actor principal que invoca al sistema para proporcionar el servicio. El sistema se comunica con los actores secundarios para completar el caso de uso.

Ejemplo de descripción de caso de uso



Buy a Product

Goal Level: Sea Level

Main Success Scenario:

1. Customer browses catalog and selects items to buy
2. Customer goes to check out
3. Customer fills in shipping information (address; next-day or 3-day delivery)
4. System presents full pricing information, including shipping
5. Customer fills in credit card information
6. System authorizes purchase
7. System confirms sale immediately
8. System sends confirming e-mail to customer

Extensions:

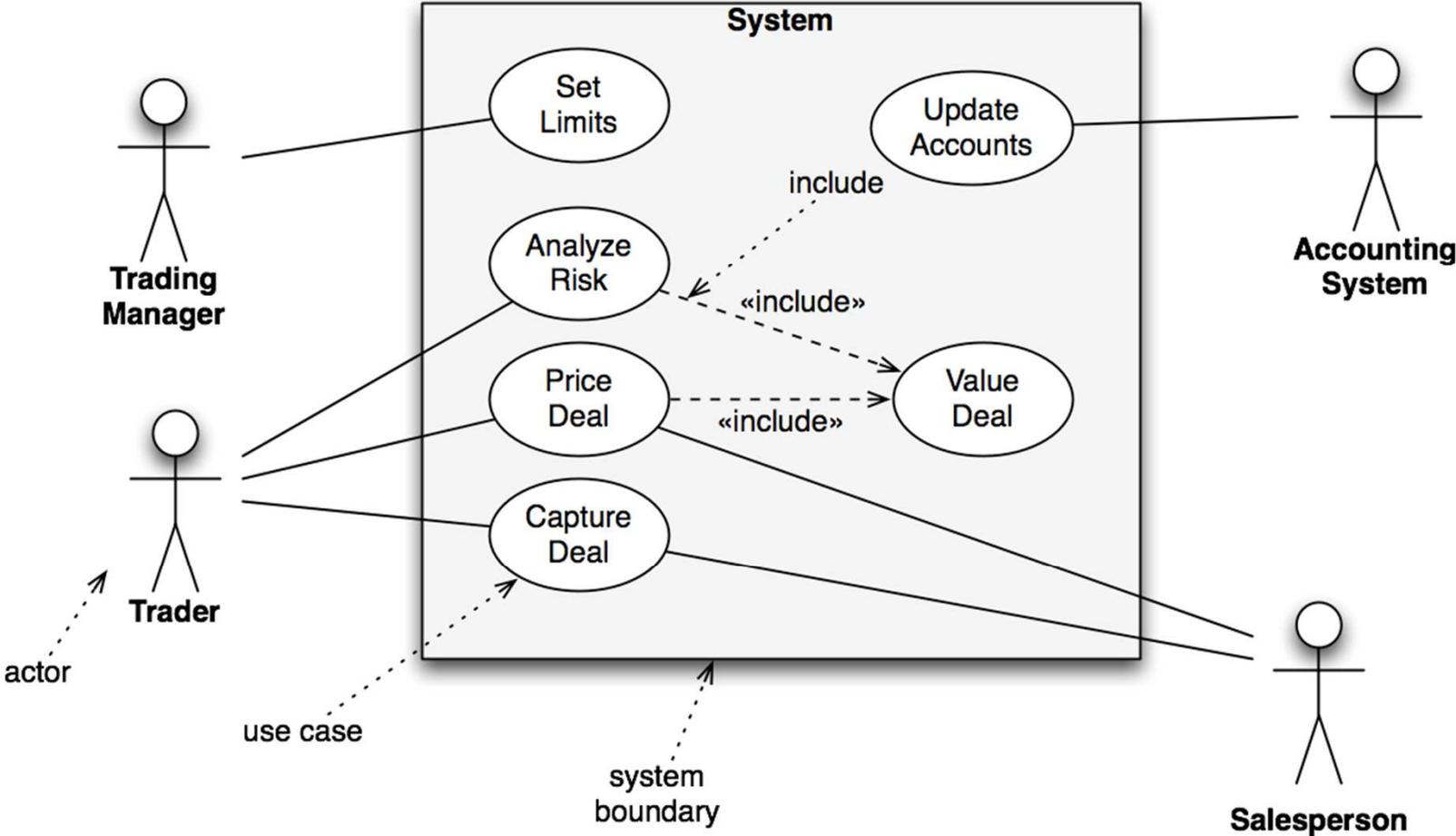
3a: Customer is regular customer

- .1: System displays current shipping, pricing, and billing information
- .2: Customer may accept or override these defaults, returns to MSS at step 6

6a: System fails to authorize credit purchase

- .1: Customer may reenter credit card information or may cancel

Diagrama de casos de uso





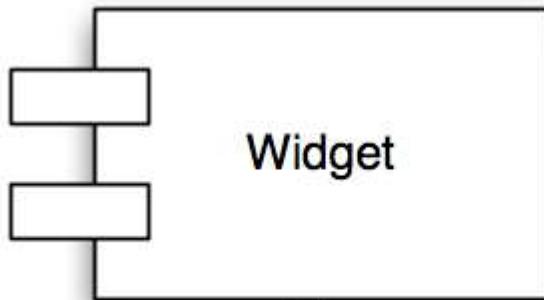
VII. DIAGRAMA DE COMPONENTES



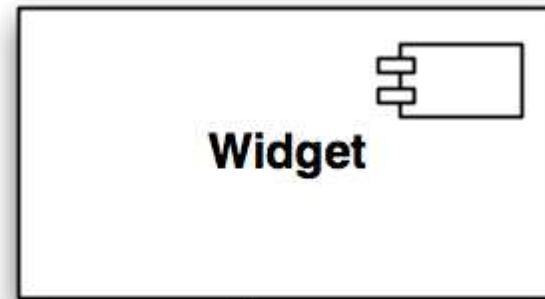
Fundamentos

- ❑ Un componente es una parte modular que oculta su implementación detrás de un conjunto de interfaces externas.
- ❑ Los componentes que comparten las mismas interfaces se pueden sustituir siempre y cuando tengan el mismo comportamiento lógico.
- ❑ La implementación de un componente puede expresarse conectando partes y conectores; las partes pueden incluir componentes más pequeños.

Notación

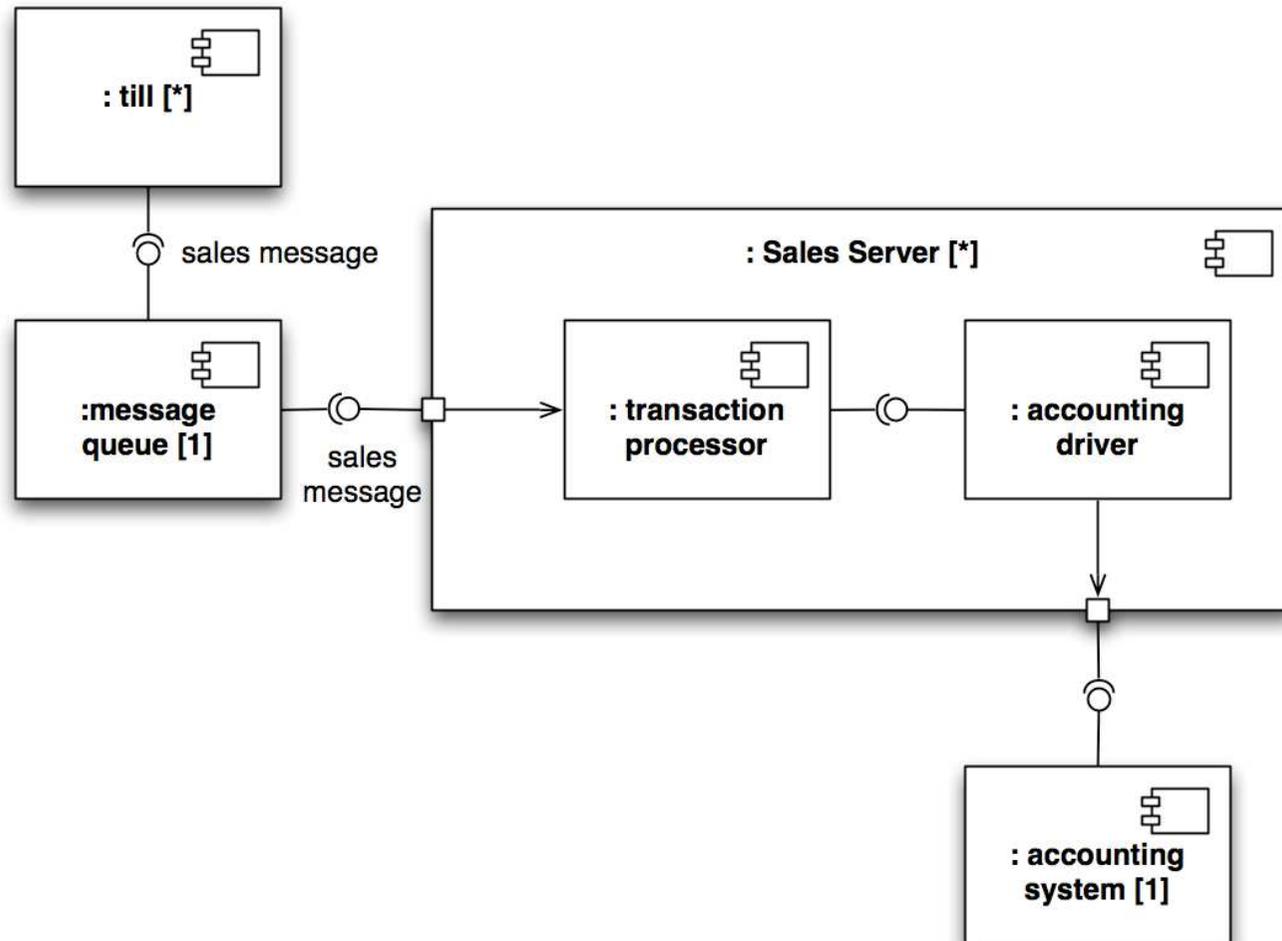


UML 1 notation



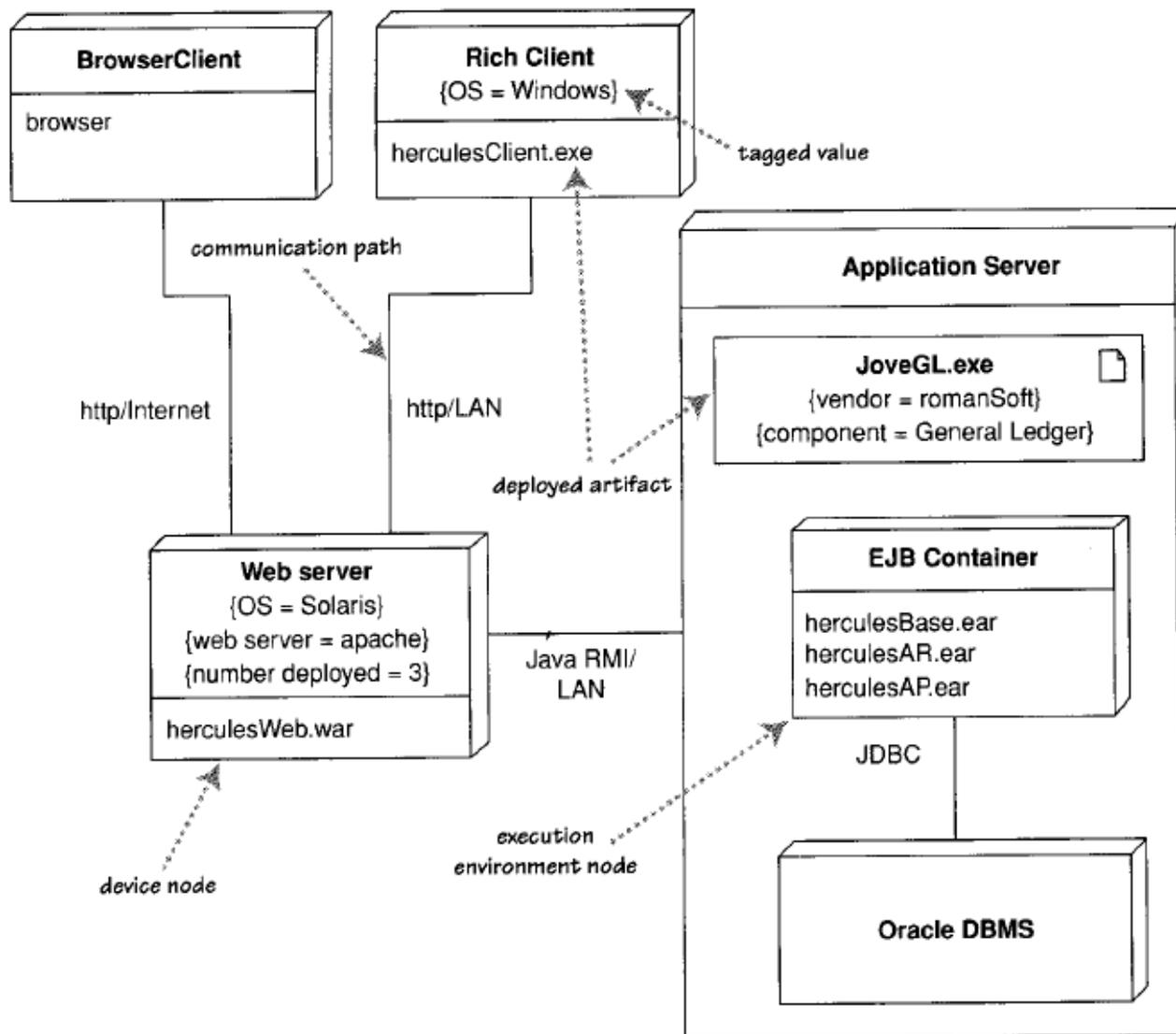
UML 2 notation

Ejemplo





VIII. DIAGRAMA DE DESPLIEGUE



Referencias



1. Booch, G., Rumbaugh, J. y Jacobson, I; El Lenguaje Unificado de Modelado; Segunda Edición; Addison-Wesley; 2006.
2. Fowler, M; UML Distilled. A Brief Guide to the Standard Object Modeling Language; Tercera Edición; Addison-Wesley; 2004.
3. Pressman, R; Software Engineering: A Practitioner's Approach; Séptima Edición; Mc-Graw Hill; 2010.
4. Weilkieins, T. y Oestereich, B; UML 2 Certification Guide: Fundamental and Intermediate Exams; Morgan Kaufmann; 2006.



Unified Modeling Language 2.0

Tomás Balderas Contreras
balderas@ccc.inaoep.mx

PREGUNTAS & RESPUESTAS