2019

Herramientas de productividad HDP115

UNIDAD III: HERRAMIENTAS PARA ANÁLISIS Y DISEÑO

BLADIMIR DIAZ CAMPOS

UNIVERSIDAD DE EL SALVADOR | Facultad de Ingeniería y Arquitectura, Escuela de Sistemas Informáticos

Contenido

Ejemplo de modelado

Se requiere un sistema para un restaurante que permita a los clientes hacer sus pedidos de comida en línea.

El sistema debe permitir a los clientes agregar platillos a un pedido. Para cada platillo ingresado al pedido, el cliente podrá agregar ingredientes adicionales. En el restaurante, un despachador podrá ver en tiempo real los pedidos que vayan siendo realizados por los clientes y atendiéndolos con la política PEPS. Para poder realizar un pedido, los clientes deberán iniciar sesión previamente.

Al momento de enviar su pedido, el cliente deberá ingresar su número de tarjeta de crédito/débito para que el despachador pueda procesar el pedido.

El jefe de cocina podrá cambiar el estado de los menús de *Disponible* a *No disponible* cuando no sea posible servir el menú. Si es posible atender el pedido, cuando el jefe de cocina lo autoriza, el despachador deberá facturar el pedido y despacharlo.

Dado que es posible que un pedido sea realizado mientras el menú solicitado se encontraba Disponible, aunque no haya estado realmente disponible, el jefe de cocina puede denegar el pedido. En cuyo caso, el despachador deberá de ponerse en contacto telefónico con el usuario para ofrecerle alternativas a su pedido.

Lista de actores:

- Cliente.
- Despachador.
- Jefe de cocina.

Lista de casos de uso.

- Ingresar pedido.
- Ingresar pedido.
- Facturar pedido.
- Autorizar pedido.
- Cambiar estado del platillo.

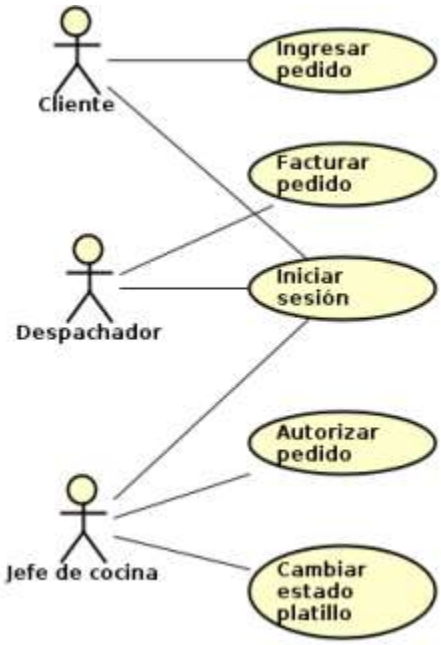


Figura 1: Modelo de casos de uso para el sistema de pedidos de comida en línea

El caso de uso narrado para del caso de uso Ingresar pedido debería ser descrito como se muestra en la tabla 1.

Tabla 1 **Caso de uso narrado para el caso de uso Ingresar pedido.**

Caso de uso:	**Ingresar pedido.**
Descripción:	Permite al usuario realizar un pedido de platillos.
Actores:	Cliente.
Precondiciones:	El cliente debe haber iniciado sesión.
Flujo nombre de los eventos:	1. El Cliente solicita el ingreso de un nuevo pedido 2. El sistema muestra una lista de platillos disponibles para que puedan ser seleccionador por el Cliente. 3a.1 El Cliente selecciona un platillo. 3a.2 El sistema muestra una pantalla para que el usuario indique la cantidad de platillos desea y si desea agregar otros ingredientes complementarios al platillo. 3a.3. El Cliente indica la cantidad de platillos que desea e indica que productos complementarios agregará. 3a.4. El sistema guarda la orden y regresa al paso 2. 3b.1. El Cliente selecciona un platillo previamente ingresado a la orden. 3b.2 El sistema muestra una pantalla con los datos del platillo seleccionado previamente ingresados permitiendo al Cliente modificar la cantidad de platillos desea y si desea agregar otros productos complementarios al platillo. 3b.3. El Cliente modifica los datos y envía las modificaciones para que sean almacenadas por el sistema. 3b.4. El sistema guarda la orden y regresa al paso 2. 3c.1. El Cliente selecciona los platillos que desea eliminar del pedido y solicita al sistema su eliminación. 3c.2. El sistema elimina los platillos seleccionados y retorna al paso 2. 4. El Cliente indica al sistema que ha finalizado la selección de los platillos a ordenar. 5. El sistema le solicita al usuario que ingrese el número de tarjeta de crédito o débito que desea usar para el pago y un número telefónico. 6. El Cliente ingresa los datos solicitados y envía los datos.

	7. El sistema guarda el pedido, le asigna un número y muestra un mensaje al Cliente con el número del pedido.
Flujo alternativo:	6.1 El Cliente cancela la orden. 6.2 El sistema muestra un mensaje de confirmación. 6.3. El Cliente confirma la cancelación de la orden. 6.4. El sistema elimina la orden y retorna al paso 2.
Poscondiciones:	El pedido ingresado será almacenado con estado Ingresado.

Nótese que el flujo de eventos refleja que existe la misma probabilidad de que el usuario elija los pasos 3a, 3b y 3c. En estos casos se dice que el caso de uso tiene *alternativas equiprobables*, y el flujo del caso de uso deberá reflejar este escenario. En estos casos es posible también describir el caso de uso por secciones, en el que cada sección corresponde a uno de los flujos que tienen la misma probabilidad de ser elegido.

Si bien, un caso de uso puede describir el despliegue de más de una pantalla, en este caso podemos observar que los pasos 3b y 3c utilizarán la misma pantalla para su funcionamiento. El flujo descrito indica que la pantalla es mostrada para, al finalizar el proceso, retornar a la pantalla principal.

También es posible observar que la descripción del caso de uso describe un ciclo. Pero este ciclo depende de la acción que desee el usuario. Los casos de uso también persiguen aclarar este tipo de especificaciones.

Para ello, es posible especificar un nivel más detallado de casos de uso, llamados Caso de uso *extendidos*.

Esta especificación, permite aislar flujos que pueden ser reutilizados y que describen de mejor forma los eventos que invocan su realización.

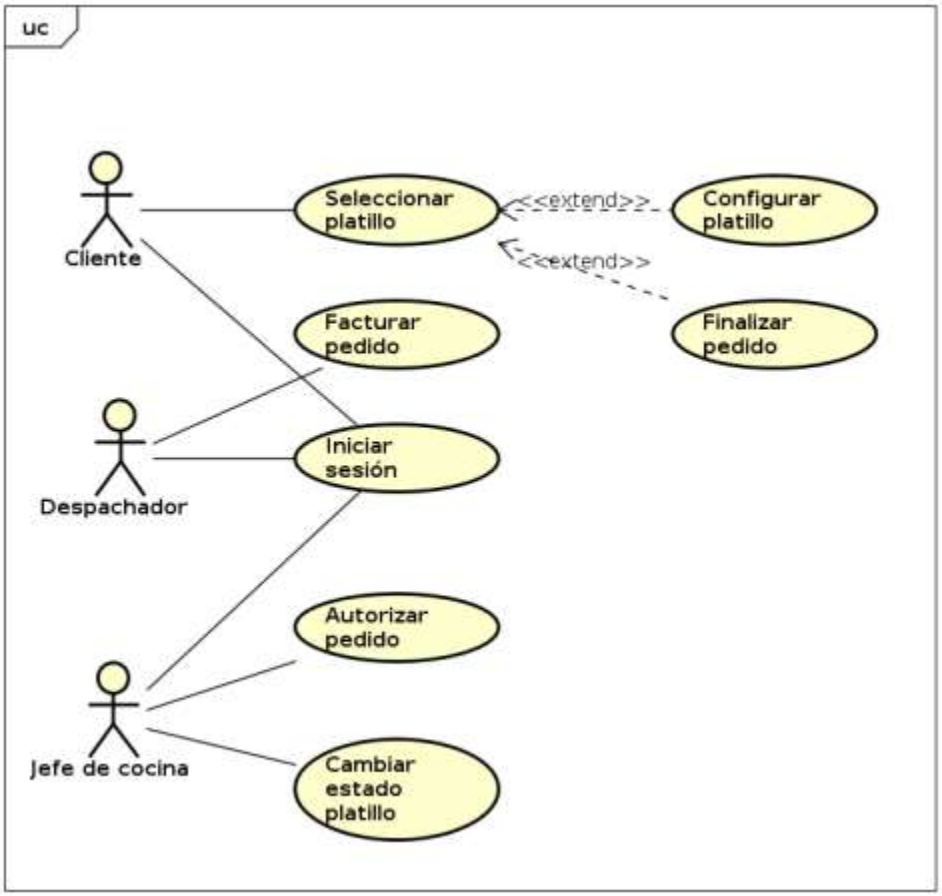


Figura 2: Modelo de casos de uso para el sistema de pedidos de comida en línea

El caso de uso Seleccionar platillo quedará descrito como se muestra en la tabla 2:

Tabla 2. Caso de uso narrado para el caso de uso Seleccionar platillo.

Caso de uso:	**Seleccionar platillo**
Descripción:	Permite al usuario seleccionar un platillo
Actores:	Cliente
Precondiciones:	El cliente debe haber iniciado sesión
Flujo nombre de los eventos:	1. El Cliente solicita el ingreso de un nuevo pedido. 2. El sistema muestra una lista de platillos disponibles para que puedan ser seleccionador por el Cliente.

	3. El Cliente selecciona el platillo que desea agregar al pedido e indica al sistema que agregue el platillo. 4. El sistema agrega el platillo al pedido. 5.a.1 El cliente indica al sistema que desea agregar un ingrediente extra al platillo seleccionado. 5.a.2. Se inicia el caso de uso Configurar platillo. 5.b.1 El cliente indica al sistema la cantidad de platillos que desea. 5.b.2 El sistema calcula el nuevo subtotal de los platillos solicitados y el total del pedido.
Flujo alternativo:	3.1 El Cliente indica al sistema que desea cancelar el pedido. 3.2. El sistema solicita al Cliente que confirme su opción 3.3 El Cliente confirma la cancelación del pedido. 3.4 El sistema elimina los datos ingresados.
Poscondiciones:	El pedido ingresado será almacenado con estado Ingresado.

Nótese que el flujo alternativo de cancelación de la orden, ahora será realizado en el caso de uso *Finalizar pedido.*

2. Prototipado basado en casos de uso

Usualmente solemos asociar los prototipos como un primer ejemplar de algún artefacto, que luego será tomado como modelo para crear otros de la misma clase. Sin embargo, un prototipo puede referirse a otro tipo de artefacto que puede ser desarrollado para otros usos.

Definiremos un prototipo como un modelo, versión incompleta o versión final de un producto, construido para probar un concepto o un proceso, o para ser posteriormente replicado.

En ingeniería de software, usualmente el prototipado consiste en desarrollar versiones incompletas del software a desarrollar, con el objetivo de verificar el cumplimiento con los requerimientos en fases tempranas del desarrollo.

Los prototipos de software suelen tener funcionalidad limitada. Sin embargo, su objetivo es verificar el cumplimiento de los requerimientos y mejorar la propuesta de funcionalidad.

2.1 Tipos de prototipo

Por su funcionalidad:

1. **Baja funcionalidad**: El prototipo consiste de un conjunto de dibujos (por ejemplo, una presentación de escenarios) que constituye una maqueta estática, no programada y no operativa de las interfaces de usuario.
2. **Alta funcionalidad:** El prototipo consiste de un conjunto de pantallas que proporcionan un modelo dinámico, programado y operativo del software.

Por su objetivo:

1. **Exploratorio**: Se trata de un prototipo no reutilizable, cuyo objetivo es clarificar las metas del proyecto, identificar requerimientos, examinar alternativas de diseño o investigar un sistema extenso y complejo.
2. **Experimental**: Se trata de un prototipo utilizado para la validación de las especificaciones de software.
3. **Evolutivo**: Es un prototipo iterativo que es progresivamente refinado hasta que se convierte en el sistema final.

Por su nivel de funcionalidad:

1. **Horizontal**: Se implementan, de forma incompleta, todas las funcionalidades requeridas para el software.
2. **Vertical**: Se implementan de forma completa algunas funcionalidades requeridas para el software.
3. **Diagonal**: Se trata de un prototipo horizontal hasta un cierto nivel, a partir del que se puede considerar vertical.

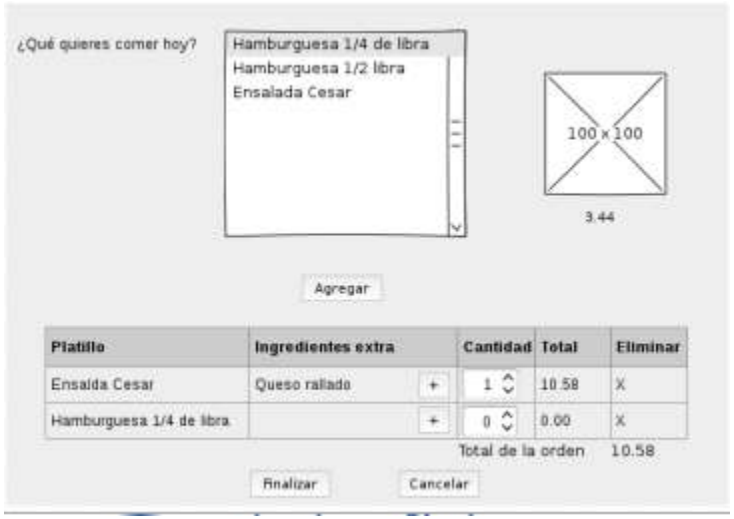
Por el nivel de componentes:

1. **Global**: Es un prototipo horizontal expandido, que cubre un amplio rango de funcionalidades de todo el software.
2. **Local**: Se elige un único componente o característica crítica del software.

La construcción de prototipos en el proceso de desarrollo de software depende de la metodología de desarrollo adoptada para el proyecto y la organización del equipo de desarrollo. Sin embargo, los casos de uso narrados persiguen crear especificaciones que permitan después al programador, tener una idea más clara de los componentes de las interfaces de usuario y de la funcionalidad esperada, para luego ser validada por el usuario.

Retomaremos el ejemplo de pedidos de comida en línea para realizar un prototipo de baja fidelidad.

Caso de uso:	Seleccionar platillo
Descripción:	Permite al usuario seleccionar un platillo
Actores:	Cliente
Precondiciones:	El cliente debe haber iniciado sesión
Flujo nombre de los eventos:	1. El Cliente solicita el ingreso de un nuevo pedido. 2. El sistema muestra una lista de platillos disponibles para que puedan ser seleccionados por el Cliente. 3. El Cliente selecciona el platillo que desea agregar al pedido e indica al sistema que agregue el platillo. 4. El sistema agrega el platillo al pedido. 5.a.1 El cliente indica al sistema que desea agregar un ingrediente extra a uno de los platillos agregados. 5.a.2. Se inicia el caso de uso Configurar platillo. 5.b.1 El cliente indica al sistema la cantidad de platillos que desea, de uno de los platillos agregados. 5.b.2 El sistema calcula el nuevo subtotal de los platillos solicitados y el total del pedido. 6. El cliente indica al sistema que desea finalizar el pedido 7. Se inicia el caso de uso Finalizar pedido.
Flujo alternativo:	3.1 El Cliente indica al sistema que desea cancelar el pedido. 3.2. El sistema solicita al Cliente que confirme su opción. 3.3 El Cliente confirma la cancelación del pedido. 3.4 El sistema elimina los datos ingresados.
Poscondiciones:	El pedido ingresado será almacenado con estado Ingresado



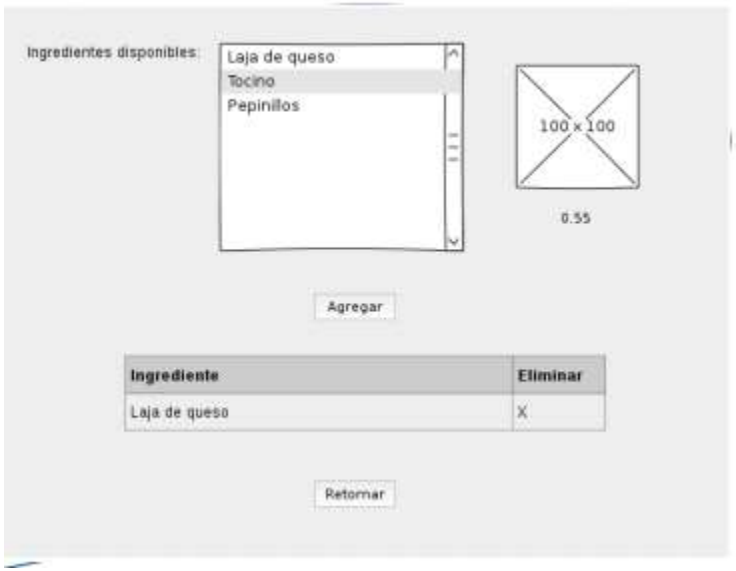
Nótese que la descripción del flujo de eventos tiene una forma de acción-reacción. Es decir, por cada acción del actor, debe haber una respuesta del sistema. Además, la especificación del caso de uso, aparecen nuevamente alternativas equiprobables. Sin embargo, en este caso no es necesario especificar ningún ciclo de iteración, ya que las respuestas del sistema dependerán de los eventos que genere el usuario.

La descripción del caso de uso nos da, no solamente una idea de cómo debe la interfaz actual permitir que el usuario llegue al resto de las interfaces, sino además una idea del diseño esperado.

Mientras más detallada sea la especificación del caso de uso narrado, más sencillo será al programador saber qué controles se requieren en la interfaz y la funcionalidad esperada. Sin embargo, el analista puede evitar este nivel de detalle de especificaciones dejando a criterio y creatividad del programador de las interfaces, apoyado en los estándares de diseño de interfaces de usuario.

El caso de uso Configurar platillo, quedará documentado de la siguiente forma:

Caso de uso:	Configurar platillo
Descripción:	Permite al usuario configurar un platillo.
Actores:	Cliente
Precondiciones:	El cliente debe haber iniciado sesión.
Flujo nombre de los eventos:	1. El sistema muestra una lista de ingredientes adicionales que pueden agregarse al platillo seleccionado. 2. El cliente agrega un ingrediente adicional 3. El sistema agrega a la lista de ingredientes adicionales, el ingrediente seleccionado. 4. El cliente indica al sistema que ha finalizado la configuración del platillo
Flujo alternativo:	
Poscondiciones:	El pedido ingresado será almacenado con estado Ingresado.



Nótese que el caso de uso Configurar platillo es un caso de uso extendido de Seleccionar platillo, por lo que al finalizar el caso de uso (el usuario hace clic en el botón Retornar), el sistema retornará a la ejecución del caso de uso Seleccionar platillo.

2.2 Modelo conceptual

El modelo conceptual es una primera aproximación al diagrama de clases del sistema. En esta etapa lo que se busca es identificar los conceptos más relevantes del problema, sus atributos y relaciones. El modelo conceptual no es una descripción de los componentes del software; sino más bien representa los conceptos en el dominio del problema en el mundo real.

2.2.1 Simbología

Clase

Es una definición de una categoría de objetos. Es representada por un rectángulo dividido en tres partes: Nombre de las clases; Atributos y sus tipos; Prototipos de las operaciones (sus nombres

tipos) y el tipo de retorno.

Cuando se está elaborando el modelo conceptual lo más relevante es el nombre de la clase y de ser posible, sus atributos.

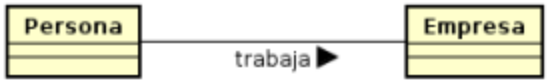
Relaciones:

En el modelo de clases, existen cinco tipos de relaciones:

1. Asociación.
2. Clases de asociación.
3. Composición.
4. Agregación.
5. Generalización

Asociación

Una asociación es un tipo de relación entre clases que describe una conexión significativa entre

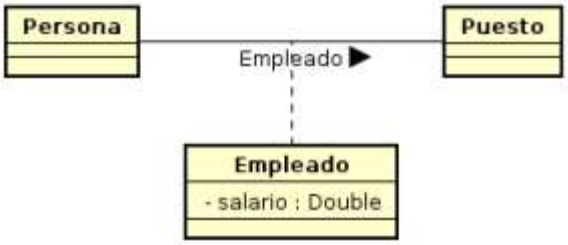


dos clases. Se representa por una línea continua entre los conceptos y el nombre de la relación.

La asociación sirve para dar un significado al modelo. En el ejemplo que se muestra, la asociación describe el hecho de que hay personas que trabajan en una empresa.

Clases de asociación

Una asociación entre dos clases define únicamente que las clases tienen una relación entre ellas. Sin embargo, cuando producto de esa relación aparecen nuevos atributos, a dicha asociación se agrega una clase de asociación en la que será posible definir esos atributos.



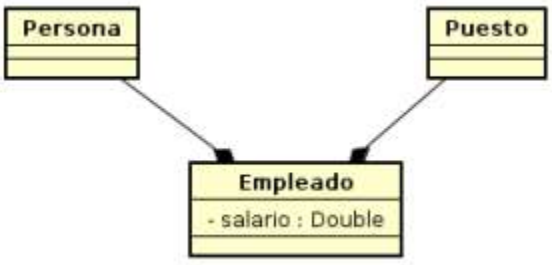
En el ejemplo que se muestra, una persona trabaja en una empresa desempañando un puesto, eso lo convierte en un empleado en el que será necesario especificar un salario.

Composición

Una composición es un tipo de relación entre clases que indica que una clase contiene (o está

compuesta por) objetos de otras clases.

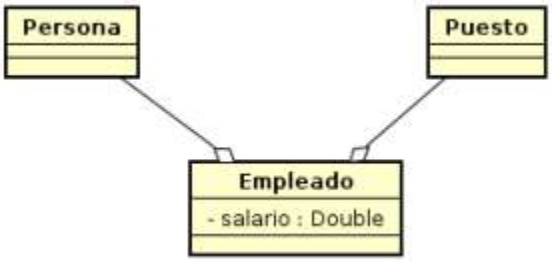
A diferencia de la agregación, la composición indica que la relación entre los objetos es de tipo



"parte/todo".

En el ejemplo que se muestra, cuando el objeto de la clase Empleado, que incluye dos objetos (uno de la clase Persona y otro de la clase Puesto), será destruido cuando el objeto que los contiene sea destruido.

Agregación

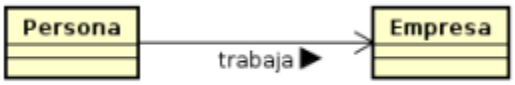


Al igual que la composición, la agregación indica que una clase está compuesta por objetos de otras clases. Sin embargo, a diferencia de la composición, la agregación indica que los objetos incluidos en el objeto que los contiene son incluidos por referencia. Esto significa que los objetos incluidos en el objeto que los contiene, seguirán existiendo cuando el objeto que los contiene sea destruido.

En este ejemplo, los objetos de Persona y Puesto que se incluyen en un objeto Empleado, existen indistintamente de que el objeto de la clase Empleado exista.

Navegabilidad

Es posible establecer la forma en que los conceptos asociados conocerán la relación. La



navegabilidad indica en qué sentido puede darse la relación y como los conceptos involucrados serán "conscientes" de dicha relación. La navegabilidad se representa a través de una flecha continua que va desde la clase que puede "ver" la relación hacia la clase que no puede "verla".

En el ejemplo que se muestra, una instancia de la clase Persona podrá conocer la empresa para

la que trabaja, puesto que la navegabilidad indica que la clase Empresa es visible para ella. Sin embargo, una instancia de la clase Empresa, no podrá conocer las personas que trabajan en ella.

Multiplicidad

La multiplicidad describe la cantidad de instancias de una clase que participan en una relación. Por defecto las relaciones no requieren la especificación explícita de la multiplicidad cuando esta es de 1.

Cuando esta cantidad de objetos no es conocida, se puede indicar un rango estimado. Las multiplicidades más utilizadas son: 1, 1..*, 0..1 y *.

El ejemplo que se muestra indica que de un conjunto de instancias de la clase Persona (muchas) podrán tener solamente una instancia de la clase Empresa asociada. El código Java para dichas clases sería como se muestra en la lista de código siguiente.

class Persona{	*class Empresa{*
Empresa empresa;	*List<Persona>personas;*
String nombre;	*String nombreComercial;*
String apellidos;	*}*
}	

Lista de código 1. Implementación del modelo de clases Persona Empresa.

El segundo ejemplo indica que de un conjunto de instancias de la clase Persona (muchas) podrán tener varias (muchas) instancias de la clase Empresa asociada. El código Java para dichas clases sería como se muestra en la lista de código siguiente.

class Persona{ *List<Empresa> empresas;* *String nombre;* *String apellidos;* *}*	*class Empresa{* *List<Persona>personas;* *String nombreComercial;* *}*

Lista de código 2. Implementación del modelo de clases Persona Empresa muchos a muchos.

El tercer ejemplo muestra una relación de 1 a muchos entre Persona y Empresa, pero especificando navegabilidad en la relación.

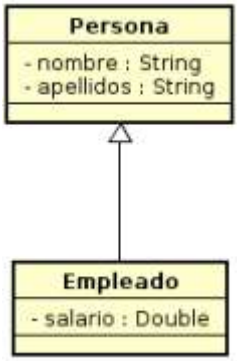
class Persona{ *String nombre;* *String apellidos;* *}*	*class Empresa{* *List<Empresa> empresas;* *String nombreComercial;* *}*

Lista de código 3. Implementación del modelo de clases Persona Empresa con navegación.

Generalización

A la clase base se le llama superclase o clase “madre”, a la clase que hereda se le llama subclase o clase “hija”. La clase hija heredará todos los atributos y operaciones de la clase madre que no sean privados.

Al igual que con los casos de uso, determinar qué clases pueden ser subclases o superclases de otras es un proceso de abstracción en el que se deducen o inducen dichas relaciones a lo largo del proceso. Identificar dichas relaciones está determinado por la necesidad de reutilizar las características (atributos y operaciones) de una clase, e incorporar nuevas características. O bien, de advertir que varias clases tienen un comportamiento común, que puede ser generalizado.



En el ejemplo, Persona es una superclase de Empleado o Empleado es una subclase de Persona. La clase Empleado hereda de la clase Persona y la extiende.

Proceso

Existen varios métodos para identificar los conceptos más relevantes del negocio. El diccionario de casos de uso son una ayuda muy útil, no solamente para reconocer los requerimientos funcionales del sistema, sino además para familiarizarnos con estos conceptos más relevantes.

El modelado de los procesos del negocio, a través de diagramas de actividades también son una buena referencia para reconocer dichos conceptos.

Independientemente del método que se adopte, lo fundamental es reconocer estos conceptos, aislarlos y establecer sus características y relaciones.

Esa identificación puede realizarse con una lista de conceptos más relevantes (generalmente los sustantivos) encontrados tanto en la descripción del problema como en los casos de uso. Esta lista es llamada por algunos autores como Clases candidatas. El mismo proceso de elaboración del diagrama de clases permitirá ir depurando la lista elaborada bajo los criterios de pertinencia, precisión (no redundancia) y relevancia.

Ejemplo

Retomaremos el problema de los pedidos en línea en un restaurante.

Las clases candidatas que podemos identificar son:

1. Cliente
2. Platillo
3. Pedido
4. Despachador
5. Ingredientes
6. Menú

Los conceptos Cliente y Despachador se refieren a actores, por lo que no deberán ser incluidos en el modelo conceptual.

Aunque el cliente no deba ser modelado como clase, será necesario contar con una entidad que permita validar las credenciales de usuario del cliente durante el proceso de inicio de sesión. Por ello requeriremos una entidad Usuario.

También se debe tomar en cuenta que, un pedido consta de varios platillos. Cuando un usuario seleccione un platillo deberá indicar la cantidad que desea y en caso de desearlo, los ingredientes adicionales que desea agregar a ese platillo. Esto representa la necesidad de definir una clase de asociación entre Pedido y Platillo, que permita incorporar los atributos producto de esa relación. Además, esta nueva clase de asociación estará relacionada con los ingredientes.

Nótese que el concepto de menú se refiere a los platillos disponibles, por lo que no será necesario agregarlo al modelo.

Figura 3: Modelo de casos de uso para el sistema de pedidos de comida en línea

Para verificar la validez del modelo puede comprobarse la semántica del modelo "leyendo" su significado.

1. Un usuario realizar uno o varios pedidos.
2. Un pedido incluye uno o varios platillos como detalle.
3. Cada platillo que es seleccionado e ingresado a la orden (DetallePedido) puede ser complementado con uno o varios ingredientes adicionales.