

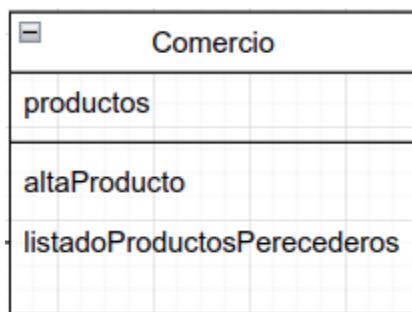
Diagrama de clases

Un diagrama de clases es una representación gráfica de la estructura estática y las relaciones entre las clases que componen un sistema de software.

Se utilizará la notación UML para representar el diagrama de clases.

Clase

Una clase se diagrama mediante un rectángulo de tres filas. La fila superior contiene el nombre de la clase, la fila del centro contiene los atributos de la clase y la última expresa los métodos o las operaciones que la clase implementa.



Modificadores

En UML existen varios modificadores, que son detalles extras que se dan sobre los atributos y los métodos. Si bien la lista es más amplia, nosotros utilizaremos los siguientes modificadores:

- Derivado (/)
- Estático (subrayado) Para las variables y métodos de clase.

El modificador Derivado, se refiere a los atributos que se pueden calcular a partir de otros atributos, no son necesarios de almacenar ya que pueden ser calculados, pero según la naturaleza del problema a resolver podemos necesitar almacenarlos. También puede ser llamado atributos calculados.

El modificador Estático se utiliza para identificar variables y métodos de clase. Las variables de clase se escriben siempre iniciando en mayúscula, mientras que los métodos de clase se escriben iniciando en minúscula.

En UML existen otros modificadores que no utilizaremos, tales como los modificadores de público (+), privado (-) ya que todos los atributos en Smalltalk se comportan como privados y todos los métodos como públicos. Esta es una característica de los lenguajes orientados a

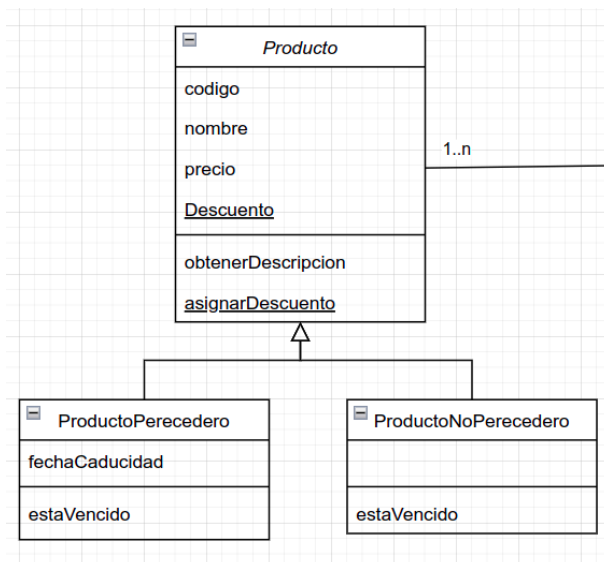
objetos. El hecho de que los atributos sean privados permite la aplicación del concepto de encapsulamiento.

Tipos de datos

Ya que en Smalltalk no hay tipos de datos, no utilizaremos la notación de tipos de datos en los atributos o variables de clase, tampoco en los métodos para representar lo que retornan.

Herencia

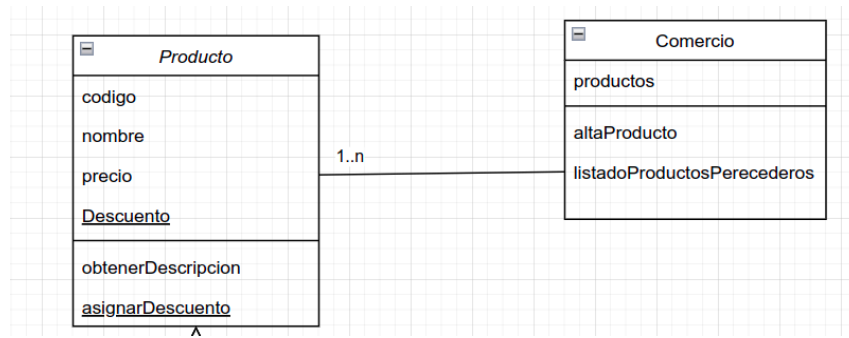
En cuanto al diagrama, la herencia es una forma de modelar la relación entre clases donde una clase (la subclase) hereda características y comportamientos de otra clase (la superclase) representado con una línea recta con una punta de flecha cerrada que señala a la superclase. Esto ayuda a visualizar la estructura jerárquica de clases en un sistema orientado a objetos.



Ensamble

Esta relación indica que la clase "todo" está compuesta por instancias de la clase "parte", y que estas instancias están fuertemente acopladas con la clase "todo". En otras palabras, la clase "todo" contiene o está formada por instancias de la clase "parte".

Se representa con una línea entre las dos clases, la cardinalidad se pone del lado donde está la clase "parte" y es quien también tendrá el atributo que alojara a la clase "parte". La cardinalidad indica el número de instancias de la clase "parte" que están relacionadas con una instancia de la clase "todo". Esto se simboliza con el formato (n...m) en el lado de la parte, donde "n" es el número mínimo de instancias de la clase "parte" y "m" es el número máximo de instancias de la clase "parte" que pueden estar asociadas con una instancia de la clase "todo".



Clases abstractas

Son clases de las cuales no se crean instancias, contienen aspectos compartidos (variables y métodos de instancia) que luego serán heredados por sus subclases. Por otro lado, sus subclases heredan los atributos y comportamientos de la clase abstracta, incorporando atributos y comportamientos específicos de acuerdo al objetivo con el que se haya creado la subclase.

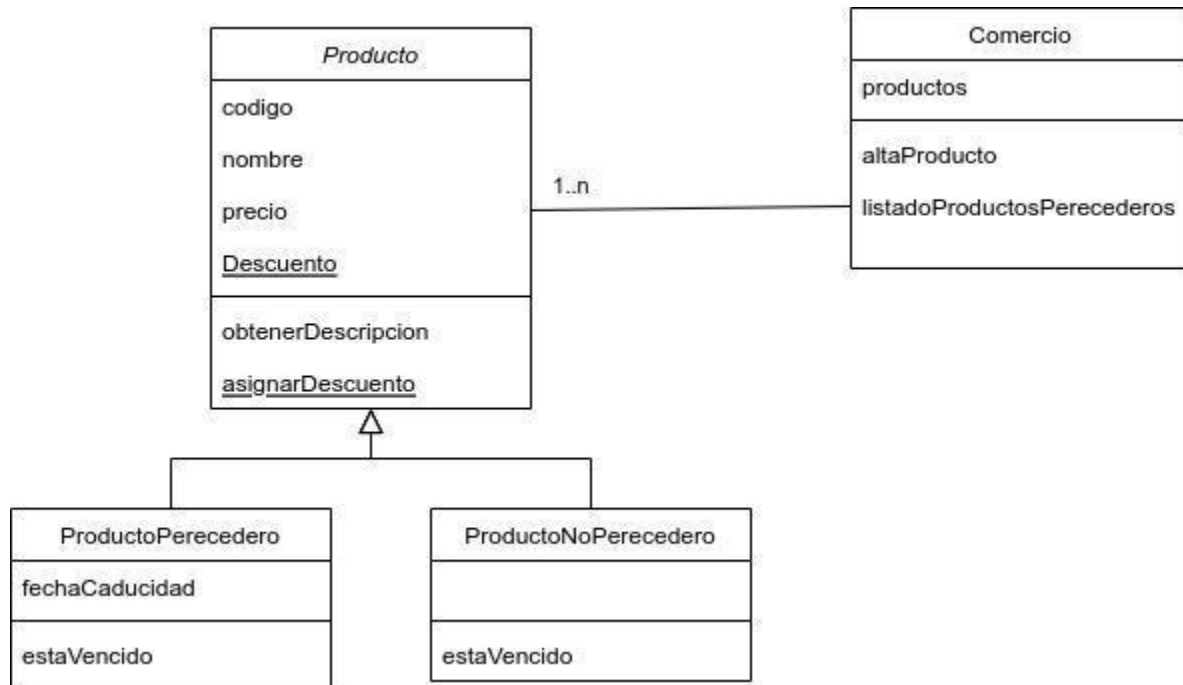
En cuanto al diagrama de clases, si bien no es obligatorio marcar cuando una clase es abstracta o no, se puede diferenciar por medio de escribir el nombre de la clase en cursiva, o bien poner el identificador “abstracta”.

Resumen

En síntesis, los elementos que representaremos en un diagrama de clases son:

- Clases
- Atributos de instancia una clase
- Atributos de clase de una clase
- Métodos de instancia de una clase
- Métodos de clase de una clase
- Relación de Herencia
- Relación de Ensamble
- Atributos derivados
- Clases Abstractas

Ejemplo de diagrama de clase



- Los elementos mostrados tenemos las clases **Producto**, **ProductoPerecedero**, **ProductoNoPerecedero** y **Comercio**.
- Como atributos de instancia podemos nombrar a los atributos código, nombre y precio de la clase **Producto**, fechaCaducidad de la clase **Producto Perecedero** y productos de la clase **Comercio**.
- Como atributos de clase tenemos a la variable *Descuento* en la clase **Producto**.
- Como método de instancia tenemos a *#obtenerDescripcion* en la clase **Producto**, *#estaVencido* en la clase **ProductoPerecedero** y **ProductoNoPerecedero** y los métodos *#altaProducto* y *#listadoProductosPrecederos* en la clase **Comercio**.
- Cómo método de clase tenemos *#obtenerDescuento* en la clase **Producto**.
- Hay una relación de herencia entre **ProductoPerecedero**, **ProductoNoPerecedero** con **Producto**, dónde **Producto** es la superclase y las otras las subclases.
- Existe una relación de ensamble entre producto y comercio, donde el **Comercio** es la clase “todo” y **Producto** es la clase “parte”, el comercio tiene de 1 a n productos.
- Por último, la clase **Producto** es abstracta y se lo marca con letra cursiva.

Cabe destacar que en la variable “productos” de la clase **Comercio** se alojarán los objetos **Producto**, y los productos alojados serán perecederos o no perecederos indistintamente, ya que la relación se da con la clase abstracta **Producto**.