

Diagrama de clases (Parte 2)

Sitio: Agencia de Habilidades para el Futuro
Curso: Modelado y Diseño de Software 1º D
Libro: Diagrama de clases (Parte 2)

Imprimido por: Eduardo Moreno
Día: lunes, 31 de marzo de 2025, 10:06

Tabla de contenidos

1. Introducción

1.1. Vinculación con la historia de taxi

2. Relación entre modelado y programación

2.1. Ejemplo

3. Visibilidad de las clases

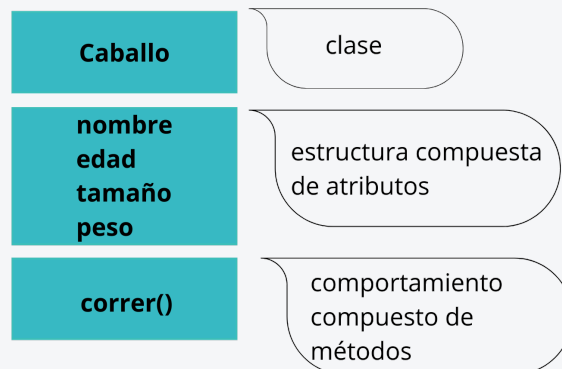
3.1. Ejemplo de aplicación



En la semana anterior vimos cómo se representaba una clase y qué elementos la componen. También qué son los objetos y las clases.

Cuando vas a desarrollar un programa informático (por ejemplo, el que necesita la empresa de taxis) debes tener en cuenta que se compone de muchos objetos y clases, algunos de los cuales comparten la misma estructura y comportamiento.

Como ejemplo y para recordar a qué nos referimos, te damos un ejemplo:



- ¿Para qué necesitamos saber esto?
- ¿Cómo relacionamos esto con la programación?
- ¿Cómo lo aplicaríamos al caso de nuestro cliente?

¡Vamos a descubrirlo paso a paso!



Empresa de alquiler de taxis



¿Recordás el caso de la empresa de taxis? Podés revisarlo [acá](#). Este cliente es una empresa nueva, en expansión y necesita organizarse mejor. Por eso, tiene algunos requerimientos para nosotros, ya que quiere informatizar el proceso de administración, para optimizar el funcionamiento de las unidades.

Hay dos procesos para administrar, que involucran a los

choferes y los taxis:

1) Proceso de entrega del taxi al chofer

Cuando es el horario de ingreso del chofer, este solicita el taxi para comenzar su recorrida.

El empleado que lo recibe en la empresa, le solicita su número de chofer y el número de móvil o auto que tiene asignado. Luego, verifica que sea chofer de la empresa y que su móvil esté disponible. Si no es chofer registrado, se rechaza el pedido. Si el chofer está registrado, se procede a ubicar el auto.

- Si no tiene el móvil disponible, el empleado controla si el auto está en reparación.
- Si está en reparación, le entrega nota con fecha probable de entrega del móvil. Si no está en reparación, el empleado entrega un nuevo número de móvil (que lo extrae del almacén de móviles de auxilio), poniendo el mismo como no disponible.
- Si todo es correcto, el empleado le da el comprobante de entrega de móvil, actualizando los registros correspondientes.

2. Proceso de devolución del taxi por parte del chofer



El chofer concurre a devolver su móvil.

El chofer entrega al empleado administrativo el comprobante de entrega de móvil (recibido cuando ingresó a su horario de trabajo). Se verifica que sea un comprobante válido y entregado por la empresa.



Si es válido, el empleado verifica el horario de ingreso del chofer con el del comprobante para calcular el pago del alquiler.

Con el valor del alquiler, genera el comprobante de pago de alquiler para entregar al chofer, almacenando la copia en el lugar correspondiente.



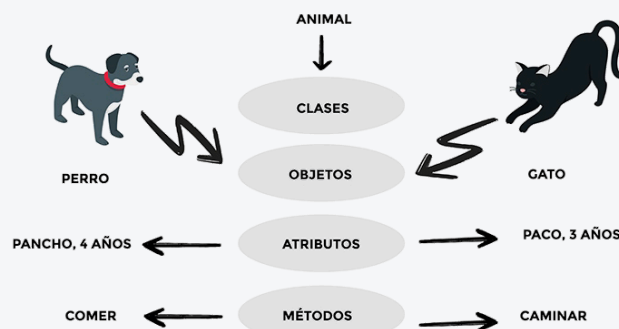
Considerando este proceso (¡bastante engorroso!), la empresa nos pide **desarrollar un software que administre y controle los taxímetros y los choferes al comienzo del turno de trabajo.**



¿Para qué necesitamos seguir aprendiendo sobre clases y objetos?

Porque cuando hacemos un programa o código orientado a objetos, si tuviéramos que definir la estructura y comportamiento del objeto cada vez que queremos crearlo, estaríamos utilizando mucho código redundante o repetido. Volvamos a pensar un minuto en nuestro cliente: necesita un programa para gestionar la entrega de los taxis a los choferes y su devolución. ¿Todos los objetos que vamos a requerir en nuestro desarrollo son de la misma clase?

Por eso, lo que se hace es primero crear una clase, que es una descripción de un conjunto de objetos que comparten una estructura y un comportamiento común. ¿Te acordás de los animales? ¿El gato y el perro de la semana anterior? Vamos a volver a verlo:



Y a partir de la clase, se crean tantas "copias" o "instancias" como necesitemos. Esas copias son los objetos de la clase.

Las clases constan de datos y métodos que resumen las características comunes de un conjunto de objetos.

Un programa informático está compuesto por un conjunto de clases, a partir de las cuales se crean objetos que interactúan entre sí.



En otras palabras, una clase es una plantilla o prototipo donde se especifican:

- Los atributos comunes a todos los objetos de la clase.
- Los métodos que pueden utilizarse para manejar esos objetos.



Ejemplo

Para entender mejor el concepto entre un objeto y su clase, pensá en un molde de galletitas y las galletitas.

El molde sería **la clase**, que define las características del objeto, por ejemplo su forma y tamaño. Las galletitas creadas a partir de ese molde son los **objetos o instancias**.



Otro ejemplo, imagina una **clase Persona** que reúna las **características comunes** de las personas (color de pelo, ojos, peso, altura, etc.) y **las acciones** que pueden realizar (crecer, dormir, comer, etc.).

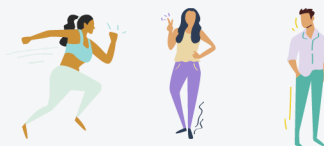
Posteriormente, dentro del programa podremos crear un **objeto Trabajador** que esté basado en esa clase Persona.

Entonces se dice que el **objeto Trabajador es una instancia de la clase Persona**, o que la clase Persona es una abstracción del objeto Trabajador.

Cualquier objeto instanciado de una clase contiene una copia de todos los atributos definidos en la clase. En otras palabras, lo que estamos haciendo es reservar un espacio en la memoria del ordenador para guardar sus atributos y métodos. Por tanto, cada objeto tiene una zona de almacenamiento propia donde se guarda toda su información, que será distinta a la de cualquier otro

Persona
nombre edad sexo
comer () caminar () dormir ()

Una clase



Muchos objetos



Alcance de los atributos y métodos

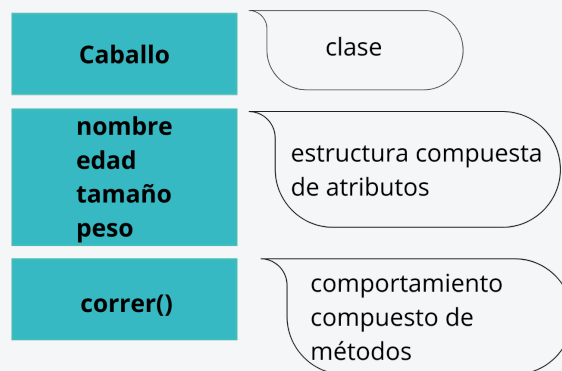
Estamos trabajando siempre con clases, pero estas clases no están aisladas, se relacionan de diferentes maneras. Volvamos a pensar en nuestro ejemplo de la empresa de taxis... ¿acaso el nudo de la cuestión no es el vínculo entre choferes y taxis, que pertenecen a clases diferentes?

Vamos a integrar los conceptos vistos hasta ahora:

En el dominio del problema (y de la solución) las clases no están aisladas, sino que se relacionan de diferentes formas:

- Una persona **VIVE EN** un apartamento (**asocia** persona y apartamento).
- Una persona **TRABAJA EN** una empresa (**asocia** persona y empresa).
- Una rosa **ES UN TIPO DE** flor (una rosa **especialización** o herencia de flor).
- Una rueda **ES UNA PARTE DE** un coche (rueda **constituyente** de coche).

Ahora bien ¿de qué manera se hace efectiva esta relación entre clases? La clave está en los atributos y métodos. Si recordamos lo visto hasta ahora, cuando trabajamos con clases y las representamos con UML, la gráfica es la siguiente:



Podemos observar en la imagen que la clase se representa con un rectángulo dividido en 3 partes:

1.- Nombre de la clase

2.- Atributos o propiedades de la clase

3.- Métodos o comportamiento de la clase

Tanto los atributos como los métodos tienen un alcance, es decir, que son o no visibles por otras clases relacionadas.

La visibilidad se clasifica de la siguiente manera:

(-) Privado

Es el más fuerte. Esta parte es totalmente invisible para otra clase

(#) Protegido

Los atributos/comportamientos protegidos están visibles para las clases amigas y para las clases derivadas de la original

(+) Público

Los atributos/operaciones públicos son visibles a otras clases.

¡Vamos a ver un ejemplo de cómo aplicar estos nuevos conceptos!



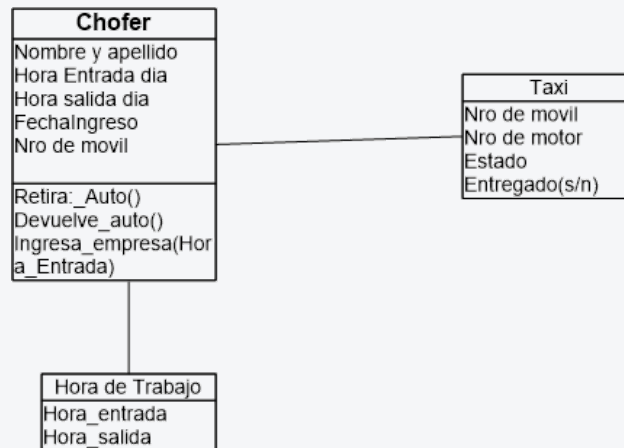
Ejemplo: la empresa de alquiler de taxis

Volvamos al ejemplo donde, según requerimientos del cliente, se nos pedía administrar y controlar los taxímetros y los choferes al comienzo del turno de trabajo. El sistema gestiona la entrega de taxis a los choferes de acuerdo al horario de trabajo de los mismos y la devolución de los coches por el chofer.



A partir de este requerimiento, hicimos el modelo E-R de 2 de los requerimientos

Vamos a transformarlos en clases

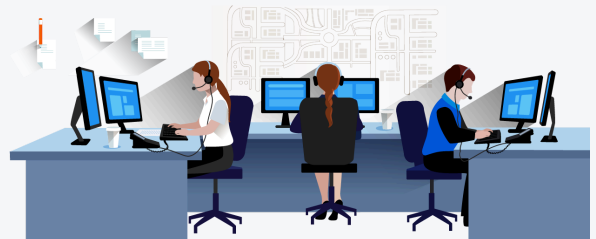


Ya tenemos las clases del trabajo de la semana anterior. Vamos a determinar cuáles son los alcances de los atributos.

Analizamos si son públicos, privados o protegidos.

De la **clase Chofer**, podemos decir que:

- + Nombre y apellido es público, ya que es necesario que las otras clases conozcan al menos los datos del chofer.
- - Hora entrada día y - Hora salida día pueden ser privados ya que son atributos para la clase chofer pero no para las otras clases.
- - Fecha de ingreso es privada porque solo la necesita la clase chofer.
- + Nro. de móvil es público ya que lo necesita la clase taxi.



Te desafiamos a determinar el alcance de los métodos de la clase chofer.