UTEC
Universidad Tecnológica

Conceptos de la Programación Orientada a Objetos

1. Introducción

La Programación Orientada a Objetos (POO) es un enfoque de programación que se fundamenta en el uso de objetos como los componentes esenciales para la construcción de software. En este paradigma, los objetos son representaciones virtuales de entidades del mundo real, lo que facilita la modelización y resolución de problemas de manera más intuitiva y estructurada.

En los lenguajes de programación orientados a objetos, como Java, Python o C++, los programas se construyen alrededor de los objetos, que son instancias específicas de clases. Una clase define la estructura y el comportamiento de un tipo de objeto, mientras que los objetos son las instancias individuales de estas clases, cada uno con su propio estado y comportamiento.

Un objeto en POO está compuesto por tres elementos principales: estado, comportamiento e identidad. El estado se refiere a los atributos o propiedades que caracterizan al objeto y que pueden cambiar a lo largo del tiempo. El comportamiento está definido por los métodos o funciones asociadas al objeto, que determinan las acciones que puede realizar y cómo interactúa con otros objetos. La identidad, por su parte, corresponde al nombre único del objeto que lo distingue de otros objetos similares.

De esta manera, la POO proporciona un marco de trabajo flexible y modular para el desarrollo de software, donde los objetos pueden interactuar entre sí mediante la invocación de métodos y el intercambio de mensajes. Esta abstracción del mundo real en términos de objetos y su interacción permite una mayor reutilización de código, facilidad para el mantenimiento y una mejor organización de los programas, lo que resulta en un código más legible, mantenible y escalable.

UTEC
Universidad Tecnológica

2. Conceptos Fundamentales

Abstracción

La abstracción se manifiesta al identificar y priorizar únicamente las características esenciales relacionadas con una tarea o problema particular, mientras se prescinde de detalles superfluos que no contribuyen directamente a la comprensión o solución del problema. En este proceso, se simplifica la complejidad del problema al enfocarse en un conjunto limitado de datos o elementos que son considerados relevantes y necesarios para representar de manera efectiva el contexto en cuestión.

Concentrarse en las características esenciales ignorando detalles específicos de las mismas y dejando de lado otras características que no interesen.

Por ejemplo, supongamos una realidad referente a dictado de un curso de una carrera, nos concentraremos en contenido del curso, los prerrequisitos que deben cumplir los alumnos inscriptos y perfil del docente que lo dicta, dejaremos de lado todos los demás elementos que existan en la carrera. Sólo nos concentraremos en los datos que nos interesa representar, por ejemplo, del docente nos interesa, su nombre, que tenga experiencia en el dictado del curso y certificación de los conocimientos del mismo, no nos interesa ni la edad ni donde vive el docente.

Encapsulamiento

El concepto de Encapsulamiento establece que debemos pensar los datos del objeto como si estuvieran escondidos (encapsulados) dentro del mismo. Permite establecer dos visiones para el objeto: Interna y Externa. La interna se concentra en los atributos del objeto y la externa en sus métodos. La idea es pensar que los atributos están protegidos dentro del objeto y queremos evitar que cualquiera pueda venir y modificarlos. El acceso a esos datos podrá ser realizado únicamente a través de los métodos definidos para el objeto.

Pensemos en una cuenta bancaria con atributos número de cuenta, cédula del destinatario de la cuenta y el monto almacenado. Si estos datos estuvieran a la vista de todo el mundo, cualquier persona podría modificarlos, en cambio lo que se hace es encapsular esos datos y sólo se acceden mediante los métodos que definamos para ello, por ejemplo, el método “acceder a la cuenta”.

Instanciación

El proceso de instanciación implica crear objetos a partir de clases. Para instanciar objetos es necesario contar con clases. Los objetos son instancias (copia exacta) de clases, una vez que el objeto está creado, éste posee identidad, estado (datos con valores o atributos) y comportamiento (métodos que actúan sobre los propios datos).

Herencia

La Herencia es un mecanismo que permite crear una nueva clase de objetos a partir de otra clase ya existente. La clase original suele denominarse clase base o superclase, mientras que la nueva clase suele denominarse clase derivada o subclase. La clase derivada hereda todos los atributos y métodos que posea su clase base correspondiente y además incorpora otros nuevos, propios de la clase derivada. Por esta razón es que se dice que la herencia es una forma de reutilización de código.

Polimorfismo

Polimorfismo significa que un mismo método se puede definir en varias clases, pero comportarse de diferente forma dependiendo de la clase en la cual esté. Dichas clases deben estar relacionadas entre sí mediante herencia. Supongamos que tenemos una clase llamada Figura. Dicha clase posee como clases derivadas a las clases Triángulo y Cuadrado. Un método (comportamiento) que interesa para todas las figuras es la operación que calcula su área.

Las clases poseen una operación que calcula su área, pero la forma en la cual se comporta dicha operación varía de una clase a otra. En la clase Triángulo es (Base * Altura) / 2 y en la clase Cuadrado es (lado * lado). El método que calcula el área es un método polimórfico. Está en todas las clases cuyos objetos son figuras, pero se comporta de forma diferente en cada una de ellas.

Etapas de la Programación Orientada a Objetos

La metodología para resolver problemas desde la Programación Orientada a Objetos se divide en tres etapas, Análisis del sistema, Diseño del sistema e Implementación del sistema en un lenguaje Orientado a Objetos.

Etapa 1 - Análisis del Sistema

Consiste en determinar cuál es el propósito general del Problema a resolver o Sistema, qué aspectos abarcará y qué aspectos no. Partiendo de los requerimientos que tenga la realidad a resolver, se estudian aquellos aspectos que sean relevantes para la resolución de los requerimientos y se construye un Modelo de Análisis que los represente. La construcción del modelo de análisis se hace identificando los Objetos presentes en la realidad que son relevantes y cómo se relacionan entre sí, dejando de lado a todos aquellos que no son de importancia.

Etapa 2 - Diseño del Sistema

Consiste en traducir el modelo generado durante el análisis en un conjunto de clases y métodos, generando una documentación lo más clara posible de las clases que hay que programar y de los métodos que hay que implementar en ellas.

UTEC
Universidad Tecnológica

Etapa 3 - Implementación del Sistema

A partir de la documentación generada en la etapa del diseño se implementan en un lenguaje de Programación Orientado a Objetos cada una de las Clases especificadas durante el diseño, definiendo estructuras de datos apropiadas para cada una de ellas, así como los métodos de cada clase, también especificados durante el diseño.

Modelado

¿Qué es un Modelo?

Lo primero que hacemos es guiarnos por la definición textual del diccionario. Según la RAE (Real academia española), la palabra modelo significa:

- Arquetipo o punto de referencia para imitarlo o reproducirlo.
- Representación en pequeño de alguna cosa.

Como la propia definición nos lo indica, el modelo es una representación o imitación de una cosa. Esa cosa, por lo general es un aspecto de la realidad (por ejemplo, una mesa, una persona).

Pasando en limpio, podemos definir al Modelo como: “El modelo es una herramienta que nos permite representar un cierto aspecto de la realidad, con cierta precisión y de la manera más completa posible”

Formatos del modelo

El modelo, no tiene ningún formato definido, ya que depende totalmente de la realidad que queramos modelar y como queramos hacerlo. Un modelo puede ser un dibujo, una foto, una

descripción verbal o textual, una escultura, etc. El formato del modelo, es generalmente elegido en función lo que queremos representar. Puede ser mucho más fácil modelar una persona con una foto, que haciendo la descripción, o mejor dicho, puede ser más preciso en ese caso, modelar a través de la foto que con la descripción de la persona.

Ejemplos

El mapa

Un ejemplo típico de modelo es el mapa. En él puede verse reflejado las formas de los distintos continentes, pero el mapa en si no es el territorio. La representación en forma de dibujo del territorio, nos muestra nociones de distancias por ejemplo o si es un mapa de población, la distribución de población. Como puede verse en las figuras, ambos mapas representan el territorio de Uruguay, pero en contextos distintos. En uno de ellos solamente vemos su límite con el resto de los países vecinos, mientras el otro tiene una granularidad un poco más detallada y muestra subdivisiones de los departamentos de nuestro país. De esto hablaremos más adelante, acerca del contexto de la realidad que vamos a modelar.

UTEC
Universidad Tecnológica

Caída libre de objetos

Si en tengo una piedra en mi mano, y la suelto, está, inevitablemente caerá hasta el suelo. Para este caso, tenemos un modelo Físico y Matemático por detrás de este acto que describe y fundamenta el motivo de ese comportamiento. En este caso, el modelo se representa mediante cálculos y ecuaciones matemáticas.

Modelado

Teniendo ahora el concepto de modelo, es sencillo poder comprender lo que significa el modelado. El modelado, es el proceso mediante el cual pasamos de la realidad a nuestro modelo elegido. En este momento, siempre elegiremos los factores determinantes para lo que queremos modelar. En el ejemplo del mapa, el modelado sería conocer y medir las distancias del territorio, y pasar al trazado del mismo de manera que se cumplan las relaciones de distancia. Como mencionamos antes, cuando trazamos el mapa de América Latina, tomamos la decisión de modelar únicamente los países, puesto que en dicho contexto no nos interesan más divisiones que estas. Lo mismo ocurre cuando trazamos el mapa de Uruguay, hemos decidido dejar fuera al resto del mundo, ya que nos interesa modela el territorio uruguayo.

Conceptos, Objetos y Modelado

Concepto

Es una entidad del dominio del problema que estamos representando que tiene una estructura. Esto implica que determina un estado (atributos que lo caracterizan) y un comportamiento. Usualmente en una realidad planteada los conceptos que identificamos para modelar son los relevantes al problema o situación que queremos resolver o aplicar.

Puede haber otros conceptos que identifiquemos que también tengan una estructura, pero no son relevantes al dominio del problema, esos conceptos no serán modelados. Por ejemplo: “En un supermercado se cuenta con la información de los productos que tienen una descripción y un precio asociado. A su vez cada producto es comprado a un proveedor.” Si lo que queremos modelar es la realidad del supermercado sin importar quién nos vende los productos el concepto relevante que modelamos es el Producto. Si el caso fuera que también queremos tener una trazabilidad de quien nos vendió cada producto, modelamos los 2 conceptos: Producto y Proveedor.

Concepto de Objeto

En esta sección, conoceremos un nuevo concepto que manejaremos a lo largo del curso. El concepto de “Objeto”. Si bien todos nosotros conocemos lo que significa la palabra objeto, siempre la asociamos a elementos físicos como ser un auto, una mesa, una casa, etc. Pero, los objetos pueden también ser cosas intangibles como los sentimientos, o algo más básico como "un programa de tele" ó "un curso de programación".

Los objetos son instancias particulares de cada concepto, por ejemplo, un objeto del concepto Empleado es: “el empleado Juan”, “el empleado Enrique”, etc. Entonces, vamos a definir a los objetos como:

“Un objeto es, cualquier elemento de la realidad, tangible o intangible.”

Como ejemplos de esto, podemos decir que un objeto es una planta, un auto, el supermercado, una canción, la amistad. Ahora, ustedes se preguntarán, por qué hemos definido este concepto de objeto. Como ya vimos en la sección anterior, el proceso de modelado, es el paso de la realidad al modelo entonces, trataremos en juntar ambos conceptos y “modelar un objeto”.

Primero que nada, vamos a definir algunos términos que vamos a utilizar. El primero de estos es el de “atributo”. Cada objeto de la realidad, tiene asociado ciertas características, veamos algunos ejemplos:

- Mesa: es marrón, redonda, baja.
- Canción: es de “No te va Gustar”, se llama “Clara”, es de género Rock.
- Casa: tiene 2 puertas, 4 habitaciones, 1 baño, 1 cocina.

Llamaremos entonces atributos a las características que puedo reconocer de cierto objeto, como, por ejemplo, su tamaño, su color, su estatura, su nombre, su género musical, etc. Es importante, distinguir entre el atributo en sí, y el valor que el mismo pueda tomar. En caso de los ejemplos anteriormente citados, decimos que: Marrón es el color de la mesa, redonda es su forma, baja es su estatura. La canción tiene nombre Clara es del artista No te va gustar y es de género Rock.

Ejemplos de Modelado

A continuación, veremos ejemplos de cómo modelar cierto objeto en una realidad dada. Lo que haremos será plantear cuál es la realidad que nos interesa conocer, para identificar los atributos de importancia del objeto en cuestión.

Ejemplo 1

Supongamos que tenemos una empresa de vestimenta, y queremos hacer un estudio de mercado acerca de ciertas prendas. Para esto, saldremos a hacer una encuesta a la calle consultando a cada persona que marca de pantalón y remera utilizan. Por último, vamos a separar por edades las respuestas, para conocer por edad, que tipo de pantalón y remera se utiliza más.

Nuestro principal objeto de trabajo, será la persona, ya que queremos modelar datos de ella.

¿Qué datos de la misma nos interesa conocer?

¿Su nombre?

¿Con quién vive?

¿La dirección?

¿Cuántas veces por semana va al gimnasio?

La respuesta es NO. Si bien estas características son de la persona, y seguramente las tiene, no es lo que nos interesa en el contexto que queremos modelar. No nos cambiaría nada saber dónde vive ni como se llama, para saber qué tipo de ropa usa. Entonces ¿Cuáles serían los datos que necesitaríamos?

1. Marca pantalón que utiliza habitualmente
2. Marca camiseta que utiliza habitualmente
3. Edad

Por lo tanto, estos son los tres atributos que nos interesan del objeto persona para la realidad dada. Si bien estos son los que nos interesan, la persona, obviamente sigue teniendo un Nombre, viviendo en cierto lugar y con ciertas personas, pero no es de nuestro interés para lo que queremos modelar. Por esto último es importante entender el contexto del objeto que estamos modelando.

Ejemplo 2

Supongamos ahora, que ya tenemos el resultado de la encuesta del ejemplo anterior y hemos tomado la decisión de fabricar cierto modelo de camiseta y cierto modelo del pantalón, pero necesitamos ahora, saber un promedio de estatura y peso de la población, para conocer los talles que fabricaremos.

Teniendo esto en cuenta, ¿Cuales son los atributos que nos interesan en este caso?

¿Marca de pantalón que utiliza?
¿Marca de camiseta que utiliza?
Edad

Nuevamente la respuesta es NO. En este caso esta información no nos interesa de la persona, sino que los atributos que necesitamos conocer serían:

1. Peso
2. Altura

Si bien, el objeto que estamos intentando modelar (persona), es el mismo que en el ejemplo anterior, al encontrarnos en otra realidad distinta a la anterior, los atributos que necesitamos conocer, no necesariamente tienen porqué ser los mismos.

Por lo tanto, ***dependiendo del CONTEXTO en que nos situamos son los atributos que nos interesan de cada objeto.*** Viendo los ejemplos anteriores esto queda claro: en una realidad a modelar nos interesaba las marcas de la ropa utilizada y en otro no, más allá de que en ambos estamos modelando el objeto persona.

UML

Ahora que tenemos las nociones que necesitamos acerca del modelado de la realidad, veremos un lenguaje gráfico que nos permite modelar todo lo que hemos mencionado. El lenguaje es denominado Unified Modeling Language (Lenguaje Unificado de modelado), o como es conocido habitualmente, por su sigla UML. UML es utilizado mundialmente para la documentación en las distintas etapas del desarrollo de software.

UTEC
Universidad Tecnológica

¿Por qué son útiles estos diagramas?

Dentro del proceso de construcción de una aplicación de software, es importante dejar documentado información de la composición del mismo. Si bien, UML, define muchos tipos de diagramas para distintos enfoques del modelado del software, nos centraremos en nuestro estudio únicamente en uno de ellos que es el **Diagrama de Clases**. Aunque también vimos el **Modelo Conceptual**.

¿En qué se diferencian el diagrama de clases con el modelo conceptual?

- El modelo conceptual es realizado en la etapa de análisis.
- El diagrama de clases es realizado en la etapa de diseño del sistema (luego del análisis).
- En el modelo conceptual NO se modelan operaciones (los conceptos no tienen operaciones).
- En el diagrama de clases SI se modelan operaciones (las clases sí tienen operaciones).
- Frecuentemente un concepto modelado en la fase de análisis, pasa luego a ser una clase en la fase de diseño.

Diagrama de Clases

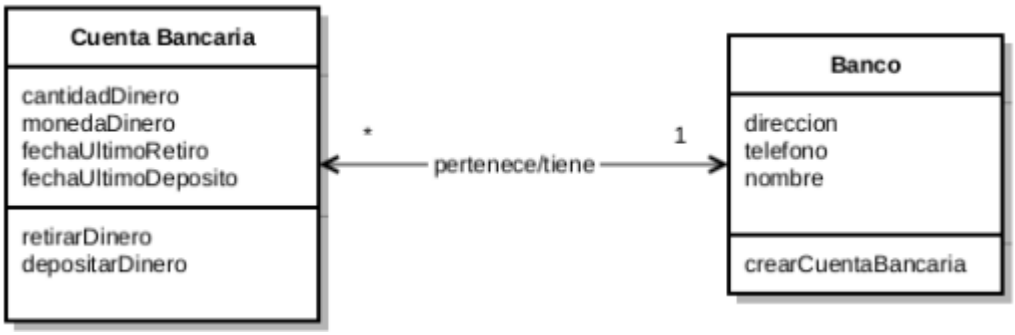
Veremos algunos de los componentes importantes del “Diagrama de Clases”. Dentro de este existen dos muy importantes, las clases y las relaciones.

Clases:

- La representación de una clase, es como muestra la figura.
- En la parte superior, se indica el nombre de la Clase (en singular).
- En la parte del medio, se indica el nombre de cada uno de los atributos.

● Por último en el sector inferior, se indican las operaciones.

Relación de asociación:



●Para modelar las relaciones se utilizan líneas que unen las diferentes clases que se relacionan.

● Sobre la línea, se le asigna un nombre a la relación. En el ejemplo anterior, depende de donde se lea, la relación tiene dos nombres: “El banco tiene cuentas bancarias” ó “La cuenta bancaria pertenece a un banco”.

● Con respecto al * y el 1 que se encuentran en los extremos de las líneas, indican las cardinalidades:

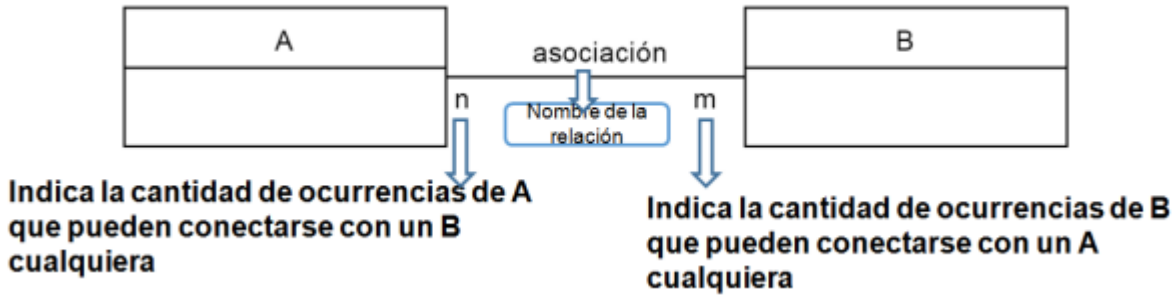
  ○ El 1 indica que toda cuenta bancaria tiene asociado uno y solo un banco, mientras que el banco tiene asociadas muchas cuentas (sin límite). El asterisco, tiene un significado de 0 a “infinito”, para modelar relaciones sin un límite específico.

● Para modelar cardinalidades o multiplicidades, lo expresamos de la siguiente manera:

  ○ Min..Max, cuando la cardinalidad está en un rango, por ejemplo: 2..5 (2 a 5), 1..* (al menos uno), 0, 3, 5 (0, 3 o 5 exacto).

¿Cómo obtenemos las cardinalidades?

Una relación determina 2 cardinalidades o multiplicidades: 1. Nos tenemos que “parar” sobre el concepto A y preguntar, ¿con cuántos B se puede relacionar UN A? 2. Nos tenemos que “parar” sobre el concepto B y preguntar, ¿con cuántos A se puede relacionar UN B?

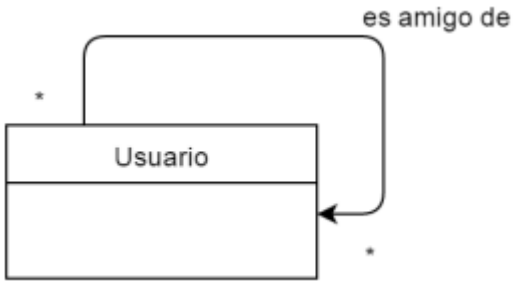


Posibles multiplicidades o cardinalidades

1. * - Cualquier cantidad (cero o más)
2. 1..* - Al menos uno (uno o más)
3. 0..1 – Rango: De cero a uno
4. 4 - Exactamente cuatro
5. 1,2,5 - Exactamente uno, dos o cinco

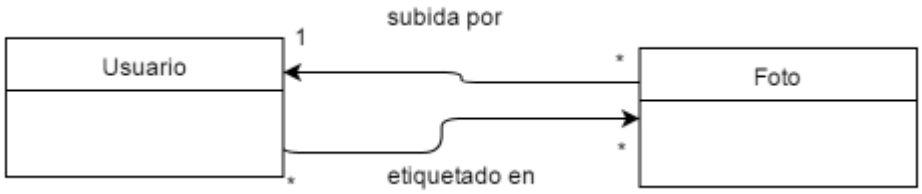
Una clase puede estar relacionada consigo misma

Por ejemplo, en Facebook los usuarios pueden ser amigos. Esto lo representamos con una relación de la clase usuario con usuario de nombre “es amigo de”.



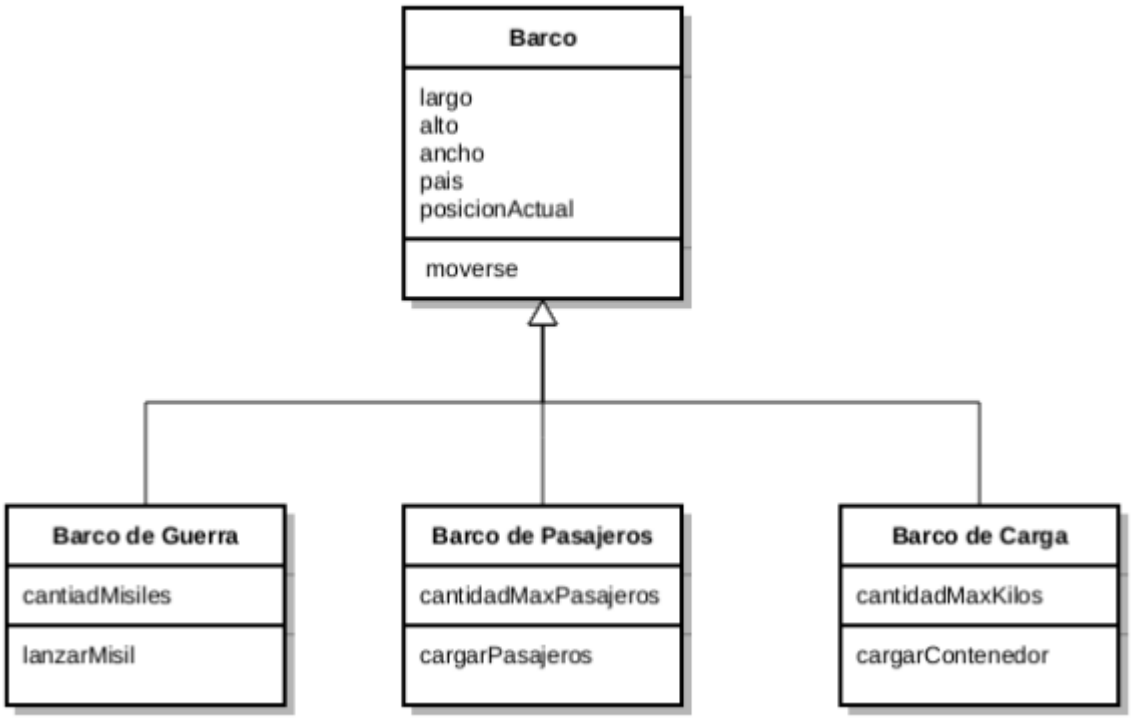
Dos clases con más de una relación definida entre ellas

Por ejemplo, en una foto hay muchos usuarios etiquetados, pero la foto es subida por un solo usuario.



UTEC
Universidad Tecnológica

Relación de Herencia:



● Para modelar las herencias, lo hacemos con una línea, que el extremo de la flecha es "vacía" como luce en la figura.

● Una clase Barco de Guerra tiene como atributos, los de la Clase Barco más los propios, en este caso: cantidad de misiles. Además, como podemos ver, cada subclase tiene operaciones específicas, además de la operación "moverse" que es heredado a todas.

¿Por qué utilizamos la herencia en la programación?

La utilizamos para no duplicar código en nuestro programa. Es decir, si no hubiera herencia, tendríamos que repetir la implementación de cada atributo y cada operación por cada clase "hija". En el ejemplo anterior, para representar Barco de Guerra, Barco de Pasajeros y Barco de Carga, utilizamos una clase padre "Barco" donde implementamos todo lo que comparten

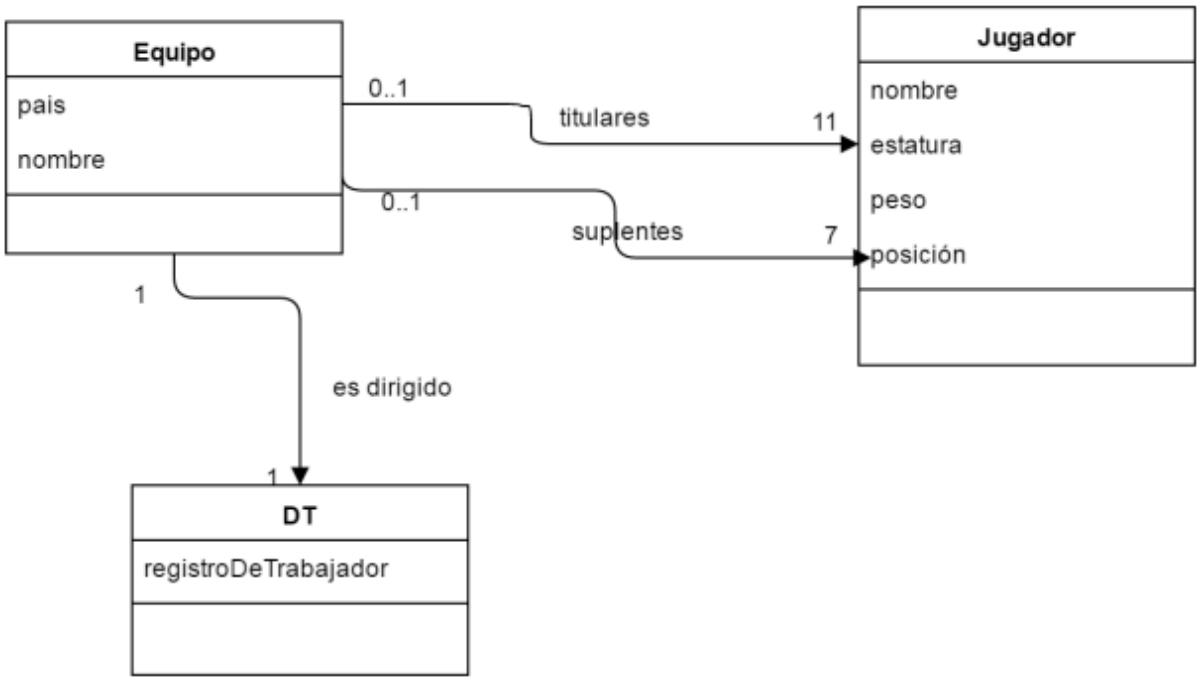
las tres clases anteriores. Sin herencia, tendríamos que repetir, la implementación de la operación “moverse” en cada clase hija.

Por lo tanto, con la herencia estamos ahorrando código duplicado. Que es una buena práctica a la hora de programar.

Ejemplo completo de modelado UML

Queremos modelar a un equipo de fútbol. Se sabe que, en el equipo, juegan 11 jugadores de titulares y se tienen 7 suplentes. De los jugadores, conocemos su nombre, estatura, peso, posición, y sabemos que solo juegan en un club. Del equipo, conocemos el país al cual pertenece y el nombre del mismo. Por último, sabemos también que el mismo es dirigido por un DT (que solo dirige a dicho equipo), que necesitamos conocer su número de registro de trabajador.

¿Como modelaríamos en UML las clases y relaciones de esta realidad?



Universidad Tecnológica

Observaciones:

● Notemos que tenemos dos relaciones posibles entre equipo y jugador, dependiendo el rol que esta toma. La cardinalidad del jugador respecto al equipo, es 0..1, porque bien está relacionado por la relación titulares o bien está relacionado por la relación suplentes.

● Cuando nos planteamos un problema, no siempre tenemos todos los datos necesarios y debemos asumir algunas cosas según el sentido común. Un ejemplo de esto es lo que asumimos para modelar las cardinalidades de las relaciones con la clase Jugador. No se nos dice: un jugador no puede ser titular y suplente al mismo tiempo, pero sabemos que esto no es posible en la realidad.