

## UML Y DIGRAMAS DE CLASE.

Para que nosotros podamos comprender de una mejor manera los patrones de diseño, es necesario que contemos con los conocimientos básico de UML y diagramas de clase, únicamente lo básico.

En este post, aprenderemos lo necesario para que podamos continuar con el curso.

### ¿QUÉ ES UML?

UML por sus siglas al español, **Lenguaje unificado de modelado**, es el lenguaje que nos permitirá modelar nuestros sistemas de Software.

UML es un lenguaje gráfico, si, puede sonar extraño, cuando escuchamos lenguaje, es probable que lo relacionemos con código de programación, en este caso, no es así.

UML cuenta con varios tipos de diagramas, los cuales, muestran diferentes aspectos de las entidades a representar. Aquí un listado de las que considero, a título personal, las más interesantes/relevantes.

- Diagrama de clases.
- Diagrama de Objetos.
- Diagrama de casos de uso.
- Diagrama de estados.
- Diagrama de secuencias.
- Diagrama de actividades.

## DIAGRAMA DE CLASES

Los diagramas de clase, son los diagramas que más estaremos utilizando en UML; Estos diagramas nos permiten representar un conjunto de clases, interfaces y su relación entre ellos.

Veamos un ejemplo. Representemos nuestra clase **Jaguar**.

# Jaguar

+ edad

+ peso

+ getEdad()

+ getPeso()

+ setEdad()

+ setPeso()

En la parte superior encontraremos el nombre de nuestra Clase. En la parte del medio encontraremos todos los atributos de nuestra clase. En la parte inferior encontraremos todos los métodos de la clase.

Marcaremos con el caractere más (+) a todos aquellos atributos y/o métodos que sean publicos, con el caracter menos (-) a los privados.

Recordemos que las interfaces tambien son clases; Su representación es casi la misma, unicamente colocamos **<< interface >>** en la parte superior.

## << interface >> Felino

- + Rugir ()
- + Maullar ()
- + Cazar ()

### HERENCIA

La relación entre clases es parte integral de la programación orientada a objetos. Si nosotros queremos representar la herencia entre dos clases lo haremos de la siguiente manera.

En este caso estamos indicando que la clase **Jaguar** hereda de la clase **Animal**.

Es de suma importancia que nosotros representemos la herencia con una línea sólida con una flecha cerrada y hueca. La flecha debe de apuntar siempre a la clase padre.

Ahora indicamos que nuestra clase **Jaguar** se encuentra heredando de la clase **Animal** e implementando la interface **Felino**.

