Programming Course

Student: Diego Peña  
Date: 3/1/22.

## Links

<https://campus.open-bootcamp.com/cursos/79/leccion/2218>

# Concepts:

### ¿What is a Programming Language?

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Sitio web

Descripción generada automáticamente

### Types of Programming Paradigms

#### Imperativos vs Declarativos

##### Imperativos

* Paradigma más antiguo.
* Secuencia de instrucciones.
* Define el paso a paso.

##### Declarativo

Se centra en el qué, en lugar del cómo

Este se centra en cual va a ser el “resultado final”

Ej imperativo: los pasos a seguir de una receta de cocina.

Ej declarativo: la foto o resultado finales de la receta al momento de ya estar cocinado.

##### Ventajas y Desventajas

* Legibilidad.
* Facilidad de Aprendizaje.
* Escalabilidad del Código.

Nota:

Legibilidad: Cualidad de lo que es legible.

Escalabilidad: Es un anglicismo que describe la capacidad de un negocio o sistema de crecer en magnitud.​ Aunque la palabra escalabilidad no existe en el diccionario de la RAE el adjetivo más cercano ampliable es de poco uso en telecomunicaciones y en ingeniería informática.

Es la capacidad de ampliación de un sistema para satisfacer las necesidades empresariales. Para escalar un sistema, debe agregar hardware adicional o actualizar el hardware existente sin modificar mucho la aplicación.

#### Funcionales vs Procedimentales

##### Procedimentales

son similares a los imperativos

##### Funcionales

son bloques de código que ya están definidos, creando así nuestro programa.

Los paradigmas funcionales son reutilizables para cualquier programa que necesitemos en el futuro.

#### División por Niveles

Los lenguajes de programación se clasifican o se dividen por el nivel de especificidad que tiene cada uno.

*Esto quiere decir mientras más bajo sea el nivel mayor características específicas se podrán acceder al hardware del computador.*

##### 0 – Lenguaje de Máquina

El lenguaje en base a 1 y 0, usando el sistema binario.

##### 1 – Lenguaje Ensamblador

Es un lenguaje intermedio el cual logra comunicarse las personas con la máquina, son instrucciones específicas, además de ser una abstracción del lenguaje de maquina.

##### 2 – Bajo Nivel

Nota: Los lenguajes de maquina y ensamblador son lenguaje de bajo nivel, porque están relacionados al comportamiento del hardware y la arquitectura del sistema que estemos utilizando (arquitecturas de hardware).

##### 3 – Medio-Bajo Nivel

Los más conocidos son C y C++, lenguaje más reconocido por sus múltiples capacidades para la programación, capacidades de alto nivel como también de bajo nivel, accediendo a registros de memoria. Este lenguaje tiene una sintaxis muy legible.

##### 4 – Medio-Alto Nivel

Lenguajes de programación moderno, EJ: Java, PHP, JavaScript, Python…

##### 5 – Alto Nivel

Para este nivel existen lenguajes de programación o Frameworks que son de alto nivel.

Los lenguajes de alto nivel están basados en lenguajes de medio nivel.

Frameworks: es una serie de funcionalidades que son añadidas en base a librerías, para que al momento de programar se haga más fácil el desarrollo de códigos, estas librerías son añadidas a los lenguajes de medio nivel volviéndolos lenguajes de alto nivel.

### Proceso de Conversión

Para hacer una conversión de nuestro programa a la computadora se necesita un traductor.

Existen 2 formas de convertir una aplicación creada por nosotros.

#### Lenguajes de Programación Compilados

Ej. C++, hace falta compilarlo, construir un archivo binario para que una maquina sea capaz de comprender lo

#### Lenguajes de Programación interpretados

Ej. JavaScript o Python.

### Entorno de desarrollo integrado IDE

Son programas con características para apoyar en el desarrollo de la programación.

Ej:

* Visual Studio Code (el más utilizado en el mundo del desarrollo)
* Atom
* Sublime Text
* PyCharm
* Notepad++

### Control de Versiones

Ejemplos de sistemas de control de versiones:

* CVS (Cooncurrent Versión System)
* SVN (Subversión)
* Git
* GitHub & GitLab

La diferencia entre GitHub & GitLab:

GitHub se centra el repositorios **Públicos**.

GitLab se centra en repositorios **Privados**.

#### Ejemplo de como funciona este control de versiones

Diagrama

Descripción generada automáticamente

### Introducción a los Lenguajes de Programación