

Clase 10 “Lenguajes y Paradigmas de Programación”

Tipados de lenguajes

- ✓ Se distinguen según si permiten o no violaciones de los tipos de datos una vez declarados. Pueden ser:
- **Tipado débil:** En estos lenguajes no indicamos siempre el tipo de variable y podemos operar con variables de distintos tipos. Su principal ventaja es que son más rápido de desarrollar, pero una clara desventaja es que podemos cometer muchos más errores si no tenemos cuidado.
- **Tipado fuerte:** En estos se nos obliga a indicar el tipo de dato al declarar la variable y no puede ser cambiado una vez definida la variable. La ventaja es que, al ser código más expresivo, cometemos menos errores. La desventaja es que son mucho más estrictos a la hora de programar y que hay que escribir mucho más código
- **Tipado estático:** la comprobación de tipificación se realiza durante la compilación y no durante la ejecución. Permite que los errores de tipificación sean detectados antes y que la ejecución del programa sea más eficiente y segura.
- **Tipado dinámico:** La comprobación de tipificación se realiza durante su ejecución en vez de durante la compilación. Comparado con el tipado estático, este es más flexible, a pesar de ejecutarse más lentamente y ser más propenso a contener errores de programación.

Frameworks

- ✓ Es una estructura previa / esqueleto que se puede aprovechar para desarrollar un proyecto.
- ✓ Es una especie de plantilla, un esquema conceptual, que simplifica la elaboración de una tarea, ya que solo es necesario complementarlo de acuerdo a lo que se quiere realizar.

¿Qué es un paradigma?

- ✓ Un paradigma es una forma de pensar bajo un modelo preestablecido.



Del código al ejecutable

- **Código Fuente:** El código fuente es una colección de instrucciones de computadora escritas usando un lenguaje de programación legible por humanos.
- **Código de máquina:** El código de máquina es una secuencia de sentencias en lenguaje de máquina o binario. Es el resultado obtenido después de que el compilador convierta el código fuente en un lenguaje que pueda ser comprendido por el procesador.
- **Compilador:** Es una aplicación traduce (compila) el código fuente en un código que el procesador puede comprender y ejecutar. Este código de máquina se almacena en forma de archivo ejecutable.
- **Intérprete:** Traduce el código fuente línea a línea y lo ejecuta directamente. El proceso de traducción funciona mucho más rápido que en un compilador, pero la ejecución es más lenta y se necesita una gran cantidad de memoria.