

10 - Lenguajes y Paradigmas de Programación

martes, 13 de abril de 2021 17:05

LENGUAJES DE PROGRAMACIÓN:

- Específicos: resuelven problemas puntuales.
- Generales: permiten desarrollar aplicaciones distintas.
- Alto Nivel: más cercanos al lenguaje natural, se puede escribir código rápido, abstractos de la funcionalidad interna de la computadora.
- Bajo Nivel: dan instrucciones muy específicas y son eficientes con el uso de recursos (es importante conocer en qué hardware se va a usar).
- Tipado Débil: no se suele indicar el tipo de variable, y podemos asignar valores de distintos tipos a la misma variable. Tienen un desarrollo más rápido, pero son más propensos a errores.
- Tipado Fuerte: se nos obliga a indicar el tipo de dato al declarar la variable, y no puede ser cambiado. Menos errores pero más estricto.
- Tipado Estático: la comprobación de tipificación se realiza durante la compilación y no la ejecución (los errores son detectados antes y la ejecución es más segura).
- Tipado Dinámico: la comprobación de tipificación se realiza durante la ejecución, siendo más flexible pero ejecutándose lentamente y con propensidad a errores.

COMPILACIÓN: Toma todo el código fuente (escrito en un lenguaje de programación legible por humanos) y lo transforma en el programa ejecutable escrito en código máquina (sentencias en binario), ejecutable en toda computadora con arquitectura y SO similar a la computadora que lo compiló. (Tiene un gran rendimiento).

MÁQUINAS VIRTUALES (VM): El código fuente va a ser compilado al código máquina pero no nuestro, sino el que entienda la máquina virtual a la que estamos asociados (buena portabilidad).

INTÉRPRETE: hace un análisis línea por línea y traduce en el momento al código máquina, de forma independiente de la arquitectura (más ralentizado).

PARADIGMAS DE PROGRAMACIÓN:

La forma de pensar bajo un modelo se le llama paradigma.

- Programación Imperativa:
 - Estructurado: sigue una línea de pensamiento donde se suele ejecutar una instrucción a la vez y uno se rige en un acotado set de instrucciones (para desarrollo de sistemas).
 - Orientado a Objetos: el código puede agruparse de tal forma que represente a una entidad e interprete mensajes (para abstracciones y crear entidades).
- Programación Declarativa:
 - Funcional: se basa en el concepto de funciones matemáticas (siempre que a la función X se le pase el valor A, va a devolver el valor B. Se lo conoce como inmutabilidad, y es característico de este paradigma).
 - Lógico: en lugar de desarrollar instrucciones, utiliza reglas lógicas para consultar al sistema, quien infiere que hacer en base a dichas reglas.
- Lenguaje específico de dominio: tratan de resolver problemáticas superespecíficas.
- Multiparadigma: han habido lenguajes que han modificado su estructura para permitir soluciones en distinto paradigmas.

MARCOS DE TRABAJO (FRAMEWORKS):

Es una estructura previa (como una plantilla o esquema conceptual) que se puede aprovechar para desarrollar un proyecto de forma más simple con una metodología dada.