

Informe de: Lenguajes, paradigmas y estándares de programación

- Introducción:

Los lenguajes en programación son esenciales ya que es una herramienta que a la hora de crear software, aplicaciones, sistemas informáticos... permite a los programadores expresar sus ideas ; los paradigmas son clave también porque estos definen un lenguaje de programación y cómo funciona (dan un conjunto de ideas que pueden ser utilizadas para realizar tareas en término de código ; los estándares simplemente recomiendan acciones y dan guías para la valoración de un producto software.

- Tipos de lenguaje de programación:

1. Lenguajes de alto nivel: Estos lenguajes están diseñados para que los humanos (programadores) puedan comprenderlos y leerlos con facilidad, también ofrecen una abstracción del hardware más alejada para tu ordenador y dan un entorno más cómodo para los programadores.

Ejemplos:

- Python, utilizado en aplicaciones Web, desarrollo del software y Machine Learning.
- Java, utilizado en desarrollo de videojuegos, inteligencia artificial, Macrodatos e internet de las cosas.
- JavaScript, utilizado en creación de páginas Web interactivas, juego y desarrollo front-end (HTML, CSS Y JavaScript).

2. Lenguajes de medio nivel: Este es utilizado cuando un lenguaje tiene características de alto y bajo nivel; también tienen acceso a bajo nivel, permiten manipular un grado del hardware ofreciendo más control.

Ejemplos:

- C ++, utilizado para crear datos complejos, definir operaciones sobre los datos complejos y relacionar los datos complejos entre ellos.
- C, utilizado para resolver tareas complejas y para optimizar un código de base C ++.

3. Lenguajes de bajo nivel: Este tipo de lenguajes están más en relación con el hardware del ordenador; requieren un mayor nivel de detalle , tienen como característica que pueden tener acceso directo a la

máquina. Permitiendo a los programadores una facilidad de actuar ya que es más directa con el hardware del ordenador.

Ejemplos:

- Ensamblador, utilizado para analizar el hardware.
- Lenguaje de máquina, utilizado para juntar las instrucciones que recibe una máquina a la hora de llevar a cabo los procesos para los que haya sido programada.
- C, utilizado para crear sistemas operativos.

- Paradigmas de programación:

1. Paradigma imperativo; es un método que permite desarrollar programas a través de procedimientos, con una serie de instrucciones , esta explica paso a paso cómo funciona el código para que el proceso sea lo más claro posible. (lenguaje C, C++)
2. Paradigma declarativo; se define el objetivo que debe lograr el programa, es decir que el enfoque está centrado en lo que se necesita lograr. (lenguaje PROLOG)
3. Paradigma orientado a objetos; este utiliza objetos como elementos necesarios a la hora de construir una solución, objetos (abstracción de algo del mundo real). (lenguaje smaltalk, c++)
4. Paradigma funcional; este está basado en un conjunto de funciones que pueden ser evaluadas para obtener un resultado, este paradigma está basado en conceptos que vienen de las matemáticas. (lenguaje HASKELL Y JAVA)
5. Paradigma lógico; este se basa en una lógica matemática , predomina un software programado según este principio contiene un conjunto de principios como una recopilación de hechos. (lenguaje PROLOG)

- Estándares de programación: La importancia de seguir los estándares y trabajandolos tenemos una base que nos sirve en el camino de mejora.

1. PEP8, aclara el pacto de estilo para la escritura de código python, ofrece pautas para el espaciado de las líneas de código, favorece la comprensión en el código Python.

2. JSR 316, detalla tipos para java Platform, define las tecnologías que se llevan a cabo en la plataforma java EE 6.
3. PSR, implanta tipos para la escritura de código en PHP, recomiendan cosas para la estructura de directorios y mejora la representación y orden del código PHP.

- Beneficios de adherirse a estándares:

1. Legibilidad y Consistencia
2. Facilita el mantenimiento
3. Colaboración efectiva
4. Mejora la calidad del código
5. Facilita la incorporación continua

- Consecuencias de no hacerlo:

1. El código complica el mantenimiento
2. Las colaboraciones se complican
3. Los errores aumentan
4. Mayor esfuerzo en mantenimiento

- Conclusión: Como ya dije antes es importante entender los paradigmas porque podrás saberlos utilizar , ya que su función es definir el lenguaje de programa; los estándares nos permiten arreglar un ciclo de entrega con alta calidad; y los lenguajes nos permite simplemente expresar tus ideas que serán ejecutadas; estos permiten que los dispositivos electrónicos interpreten instrucciones.

