

Entrega: Informe de lenguajes, paradigmas y estándares de programación

Los lenguajes de programación, los paradigmas que los caracterizan y los estándares de programación son elementos fundamentales en el mundo de la informática y la tecnología. Estos aspectos son cruciales para el desarrollo de software, ya que proporcionan las herramientas y guías necesarias para escribir, entender y mantener código de manera eficiente y efectiva.

El lenguaje ha sido el principal medio de comunicación y nos ha permitido la interacción humana y el avance como sociedad. Con el avance de la tecnología surgió la necesidad de comunicarse con las máquinas y surgen los lenguajes de programación para facilitar la comunicación tanto entre el hardware como el humano con la máquina. Pero ¿qué es un lenguaje de programación?

Los lenguajes de programación son conjuntos de reglas y sintaxis que permiten a los programadores comunicarse con las computadoras y darles instrucciones para llevar a cabo tareas específicas. Cada lenguaje tiene sus propias características y propósitos, lo que los hace adecuados para diferentes tipos de aplicaciones y entornos. Existen docenas de lenguajes de programación utilizados a día de hoy como por ejemplo: C++, C#, JavaScript, Java o Python

Los paradigmas de programación, por su parte, son enfoques o estilos de programación que definen la forma en que se estructura y organiza el código. Cada paradigma tiene sus propias ventajas y desventajas, y es importante elegir el adecuado según los requisitos del proyecto.

Por último, los estándares de programación son conjuntos de reglas y buenas prácticas que guían a los programadores en la escritura de código limpio, legible y mantenible. Estos estándares son esenciales para facilitar la colaboración entre equipos de desarrollo y para garantizar que el código pueda ser comprendido y modificado en el futuro.

TIPOS DE LENGUAJE DE PROGRAMACIÓN.

Los lenguajes de programación los podemos dividir en tres categorías principales según su nivel de abstracción y cercanía al hardware del ordenador estos son 1.

lenguajes de alto nivel

2. lenguajes de media nivel

3. lenguajes de bajo nivel

Lenguajes de alto nivel:

Están diseñados para ser más cercanos al lenguaje humano lo cual facilita el trabajo, escritura y comprensión del código. Estos lenguajes poseen un alto nivel de abstracción, permitiendo a los programadores o gente que trabaje con él centrarse en la lógica del programa sin preocuparse por el hardware.

los ejemplos más representativos de alto nivel son:

1. **python:** conocido por su simplicidad y legibilidad. ampliamente utilizado en desarrollo web, ciencia de datos, inteligencia artificial y automatización de tareas.
2. **Java:** Es un lenguaje de programación orientado a objetos que se ejecuta en una máquina virtual. popular en desarrollo de aplicaciones empresariales, aplicaciones Android y sistemas embebidos.
3. **JavaScript:** Es un lenguaje de programación utilizado principalmente para programación en el lado del cliente en páginas web.

Los usos más comunes de estos lenguajes de alto nivel son el desarrollo de software de aplicaciones, scripts, sitios webs dinámicos y aplicaciones móvil.

Lenguajes de medio nivel:

Los lenguajes de medio nivel son aquellas que se encuentran en un punto medio entre el bajo y el alto, y muchas veces se pueden confundir los lenguajes medios como altos. Esta confusión se suele dar debido a que se comportan como lenguajes de alto nivel, ya que, proporcionan la abstracción que facilita la abstracción, pero permiten manejos de bajo nivel.

Los ejemplos más representativos son:

1. **C:** Es conocido por su eficiencia y portabilidad. Ampliamente utilizado en el desarrollo de sistemas operativos, controladores de dispositivos y aplicaciones de sistemas embebidos.
2. **C + +:** Es una extensión de C que introduce características de programación orientada a objetos. comúnmente utilizada en el desarrollo de juegos, software de sistemas y aplicaciones de alto rendimiento.

Los usos comunes de los lenguajes de medio nivel son adecuadas para el desarrollo de sistemas operativos, controladores de hardware, software de sistemas embebidos y aplicaciones que requieren un alto grado de optimización.

Lenguajes de bajo nivel:

Este nivel también es llamado lenguaje ensamblador, los lenguajes de este nivel son aquellos en el que sus instrucciones ejercen un control directo sobre el hardware y por lo tanto están condicionados por la estructura física del ordenador que lo soportan.

el bajo nivel no hace referencia a la potencia, este se refiere a la reducida abstracción que existe entre el lenguaje y el hardware.

Los Ejemplos más representativos:

1. **Lenguaje ensamblador:** El lenguaje ensamblador es un conjunto de instrucciones directamente comprensibles por la CPU. Se utiliza para programación de sistemas y para tareas que requieren un control preciso del hardware.

Usos comunes: Los lenguajes de bajo nivel son esenciales para el desarrollo de sistemas operativos, controladores de hardware y software de tiempo real.

Lo más importante de los lenguajes de programación es saber elegir el más adecuado según los requisitos y las restricciones del proyecto.

Paradigma de programación

Es un enfoque o estilo fundamental de desarrollo de software. Define cómo se estructuran y organizan las tareas para construir un programa. Es una forma de abordar los problemas y solucionarlos mediante la programación. Cada paradigma tiene su propia forma de representar la información y las operaciones que se pueden realizar con ella.

Existen distintos tipos de paradigmas:

1. Programación Imperativa (o Procedimental):

Se basa en dar instrucciones al ordenador sobre cómo realizar una tarea paso a paso, es decir se centra en describir “cómo” debe hacer las cosas el

ordenador.

- Los lenguajes que usan estos paradigmas son: C, Pascal, Fortran.

2. Programación Orientada a Objetos (OOP):

Es una forma de organizar el software como una colección de objetos que contienen tantos datos como comportamientos. Este paradigma introduce conceptos como herencia, encapsulamiento y polimorfismo.

- Los lenguajes que usan este paradigma son: Java, C + +, Python, Ruby.

3. Programación Funcional:

Evalúan funciones matemáticas y enfatizan la inmutabilidad, los programas son tratados como una evaluación de las funciones compuestas. - los lenguajes que usan este paradigma son: Haskell, Lisp, Erlang, Scala

4. Programación Lógica:

En este paradigma los programas son vistos como un conjunto de afirmaciones y reglas lógicas, es usado sobre todo en inferencia y consultas sobre datos.

- los lenguajes que usan este paradigma son: Prolog

5. Programación Declarativa:

Describe “que” se desea lograr

- los lenguajes que usan este paradigma son: SQL

6. Programación Evento-Dirigida (o Basada en Eventos):

Se centra en la respuesta a eventos, como entradas del usuario o señales del sistema. Común en desarrollo de interfaces gráficas y aplicaciones web. - los

lenguajes que usan este paradigma son JavaScript

Ejemplo de lenguaje: JavaScript.

7. Programación Concurrente y Paralela:

Se centra en el diseño para ejecución simultánea de tareas, centrada en la construcción de software que puede realizar múltiples operaciones al mismo tiempo.

- Los lenguajes que usan este paradigma son Go, Erlang.

Algunos lenguajes solo usan un paradigma, los lenguajes modernos suelen ser multiparadigmas, es decir, algunos lenguajes de programación usan más de uno para su funcionamiento.

Estándares de programación:

Los estándares de programación son conjuntos de reglas y guías que establecen la forma en que el código debe ser escrito y organizado.

Legibilidad y Mantenibilidad: Los estándares promueven la escritura de código limpio y legible, lo que facilita su comprensión y mantenimiento. Esto es crucial para equipos de desarrollo que trabajan en conjunto o para programadores que regresan a su propio código después de un tiempo.

Consistencia: Los estándares ayudan a mantener una estructura y estilo uniforme a lo largo de un proyecto o entre diferentes proyectos en una organización. Esto facilita la colaboración y la comprensión del código por parte de otros desarrolladores.

Compatibilidad y Portabilidad: Al adherirse a estándares ampliamente aceptados, el código es más compatible con diferentes entornos y plataformas, lo que facilita su portabilidad.

Mejora la Colaboración: Facilita la colaboración entre equipos de desarrollo al establecer una base común de entendimiento.

Aumenta la Legibilidad y Mantenibilidad: Código más limpio y organizado es más fácil de leer y mantener, lo que ahorra tiempo y esfuerzo a lo largo del ciclo de vida del proyecto.

Algunos ejemplos:

PEP (Python Enhancement Proposals): Los PEP son propuestas de mejora para Python, un lenguaje de programación de alto nivel. Establecen estándares y directrices para aspectos como el estilo de código, la organización de módulos y la documentación.

Google Java Style Guide: Este es un conjunto de directrices establecidas por Google para el desarrollo en Java

En resumen, seguir estándares de programación es esencial para mantener un código limpio, legible y fácilmente mantenible. Esto no solo beneficia a los desarrolladores individuales, sino que también mejora la eficiencia y la calidad de los proyectos de software en su conjunto.

Para concluir:

Los lenguajes de programación, los paradigmas y los estándares de programación son pilares fundamentales en el mundo de la informática y la tecnología. Han permitido a la humanidad comunicarse tanto con las máquinas como entre sí,

facilitando el desarrollo de software y la creación de tecnologías avanzadas que han transformado nuestra sociedad.

La elección del lenguaje y paradigma de programación adecuados para un proyecto es crucial, ya que influye en la eficiencia, legibilidad y mantenibilidad del código. Los lenguajes de alto, medio y bajo nivel ofrecen diferentes niveles de abstracción y control sobre el hardware, lo que los hace apropiados para distintos tipos de aplicaciones.

Los paradigmas de programación proporcionan enfoques específicos para abordar problemas y organizar el código, cada uno con sus ventajas y desventajas. La elección del paradigma correcto depende de las necesidades y requisitos del proyecto.

Además, adherirse a estándares de programación es esencial para mantener un código limpio, legible y fácilmente mantenible. Esto facilita la colaboración entre equipos de desarrollo, reduce errores y fallos, y mejora la eficiencia en el ciclo de vida del proyecto.

En resumen, el conocimiento y la aplicación adecuada de lenguajes de programación, paradigmas y estándares de programación son esenciales para el éxito en el desarrollo de software y la creación de tecnologías innovadoras que continúan impulsando el avance de la sociedad moderna.

Webgrafía:

Epitech Spain. (2021, Julio 12).

<https://www.epitech-it.es/lenguaje-bajo-nivel/> Profile. (2020, Mayo 09)

<https://profile.es/blog/que-son-los-paradigmas-de-programacion>

[/](#) Assembler institute of technology. (2022, Enero 26).

<https://assemblerinstitute.com/blog/tipos-lenguaje-programacion>

[/](#) OpenWebinars. (2020, julio 16)

<https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/> OpenAI. (2023, Noviembre 7). Respuesta generada por ChatGPT en [2023, noviembre 7].