

INFORME DE LENGUAJES, PARADIGMAS Y ESTÁNDARES DE PROGRAMACIÓN

ÍNDICE

1. Introducción
2. Tipos de lenguaje de programación
 - 2.1 Descripción
 - 2.2 Ejemplos y usos
3. Paradigmas de programación
 - 3.1 Principales paradigmas y lenguajes representativos
4. Estándares de programación
 - 4.1 Importancia
 - 4.2 Estándares más populares
 - 4.3 Beneficios de usarlos y consecuencias de no hacerlo
5. Conclusiones
6. Referencias

1.- INTRODUCCIÓN

En programación, tanto los lenguajes como los paradigmas y los estándares, tienen mucha importancia porque la definen y muestran cómo funciona esa programación.

Un lenguaje es como un idioma que ayuda a las personas a hablar con las computadoras. es la herramienta informática que hace que se puedan desarrollar los programas. Hoy en día es imprescindible que aprendamos el lenguaje de la programación porque avanzamos cada vez más hacia un mundo tecnológico.

A su vez, los paradigmas de la programación están enormemente relacionados con los lenguajes porque los clasifican según sus características lo que permite organizar códigos, agruparlos, dar estilos diferentes, etc. Los paradigmas resuelven los problemas del lenguaje.

Además, para que todos sigamos las mismas reglas, son necesarios unos estándares que garanticen que todos sigan las mismas pautas para que los programas sean consistentes y funcionen juntos adecuadamente.

La selección adecuada de todos estos elementos es la clave para crear un programa duradero y que funcione bien.

2.- TIPOS DE LENGUAJE DE PROGRAMACIÓN

2.1 Descripción de los lenguajes.

Existen tres niveles de lenguaje en programación:

Los lenguajes de alto nivel son más fáciles de entender y usar, porque se parece más al lenguaje humano. Su principal ventaja es que una vez desarrollado se puede utilizar en diferentes máquinas.

Los de medio nivel son un punto intermedio entre ser fácil y tener el control sobre la computadora. Se utilizan en aplicaciones como los sistemas operativos. Un ejemplo es el lenguaje C

Los de bajo nivel, también llamado de máquina o ensambladores, son un conjunto de códigos binarios que la computadora interpreta, o abreviaturas en inglés o palabras mnemotécnicas como DIV, SUB, ADD... son los que se acercan más a el ordenador y son más difíciles, pero te dan un gran control específico.

2.2 Ejemplos y usos.

Hay muchos lenguajes de programación en la actualidad, los más utilizados según una encuesta de Stack Overflow, 2021 son los siguientes:

- JavaScript que es el más utilizado en todo el mundo para el desarrollo de páginas web, videojuegos, o aplicaciones móviles.
- HTML Es también utilizado en la creación de páginas web, normalmente en unión del JavaScript, crea estructuras, párrafos, encabezados...
- Python, Se usa para desarrollo de aplicaciones y ciencia de datos. Con este lenguaje se han creado plataformas muy importantes como YouTube, o Spotify o Netflix.
- SQL Este lenguaje sirve para consultar y manipular las bases de datos.
- JAVA Es uno de los más antiguos, es el lenguaje que da la estructura a las aplicaciones y a los programas y puede almacenar cantidades de datos muy grandes.
- C Sharp. Es de los más modernos. Se utiliza mucho por los programadores de videojuegos
- NoSQL Se creó para mejorar SQL que es otro lenguaje muy utilizado

3.- **PARADIGMAS DE PROGRAMACIÓN**

3.1 Descripción y lenguajes representativos

Los principales paradigmas de programación son:

Imperativo, Declarativo, Orientado a Objetos, Funcional, Lógico, entre otros.

-Imperativo:

Con este paradigma se dan instrucciones paso a paso de cómo queremos que las cosas ocurran. Es como si estuviéramos dando unas pautas detalladas de lo que queremos que nuestro ordenador haga.

El desarrollador describe paso por paso en el código todo lo que hará el programa. Lenguajes famosos son C, C++, y Java.

Con el paradigma imperativo la programación puede tener un enfoque estructurado, procedimental y modular

-Declarativo:

Sirve para describir el problema, pero no encuentra la solución. Utiliza el razonamiento lógico para dar respuestas a consultas que se le pueden hacer.

Se divide en dos tipos de programación lógica y funcional

SQL y HTML son algunos ejemplos de lenguajes representativos de estos.

-Orientado a Objetos:

Este tipo de paradigma organiza nuestro código alrededor elementos (objetos) del problema a resolver.

Tiene la función de que puede separar los componentes del programa y así es más fácil hacer mejoras. Java y Python son los ejemplos aquí.

-Funcional:

Es un paradigma con funciones matemáticas.

Se basa en componer funciones puras para dar soluciones al software. Es como decirle al ordenador "¡Ejecuta esto!" y no preocuparnos por lo que pasa en el backstage.

Haskell y Lisp son algunos ejemplos.

-Lógico:

Utiliza también la lógica matemática, donde el programa es un conjunto de hechos y reglas. No es tan común, pero Prolog es el paradigma más usado.

Lo normal es que se mezclen todos o varios de los paradigmas a la hora de hacer la programación ya que cada uno crea unas estructuras diferentes que son necesarias en la creación del programa.

4.- ESTÁNDARES DE PROGRAMACIÓN

4.1 Importancia de seguir estándares.

En el mundo de la programación, seguir las reglas es como tener un diccionario universal para escribir el código que necesitamos. Nos permiten que los programas sean legibles o que podamos realizar modificaciones si lo necesitamos. Esto no solo hace que trabajar en equipo sea más fácil, sino que también mejora la calidad y la capacidad de mantener el código, además de que se producen menos errores de seguridad.

La estandarización en la programación, al igual que en otros ámbitos de la vida cotidiana, permite establecer unas características y normas comunes que pueden ser respetadas y utilizadas en todas las partes del mundo.

4.2 Estándares de programación más populares.

Cada lenguaje tiene sus propios estándares de programación, pero hay una serie de "reglas" que son genéricos en la codificación de todos los programas algunos de ellos son:

- La estructura de los directorios y las convenciones de la nomenclatura de los archivos.

Esto facilita que se puedan buscar rápida y fácilmente los archivos que se necesitan para trabajar

- Las sangrías y los formatos

Que facilitan la lectura

- Las convenciones de la nomenclatura de variables

Facilita la comprensión del código

- Los estándares de prueba.

Garantizan una corrección eficaz y eficiente de errores y vulnerabilidades.

- No repetir

Hay que evitar un código repetitivo

- Evitar el uso de variables globales.

Se debe evitar en lo posible su uso porque pueden cambiarse durante toda la vida en la base del código, por tanto, hará que los programas sean más difíciles de reparar.

- Documentación adecuada

Esto ayudará a ver la información importante más rápidamente

4.3 Beneficios de adherirse a los estándares y consecuencias de no hacerlo:

El que un código no esté realizado bajo los estándares de programación, no significa que no vaya a funcionar o que funcione mal, pero utilizarlos es una forma de asegurarse que en el caso de que esto ocurra, va a poder modificarse y si hay que actualizarlo o mejorarlo se va a poder hacer mucho más rápido que si no los tiene.

El adherirse a estos estándares tienen muchos beneficios

1. Consistencia:

Siguiendo las reglas, el código se vuelve más uniforme. Todos usamos la misma estructura y nombres, lo que facilita entender el código.

2. Colaboración:

Al seguir las reglas, trabajar en equipo es más fácil. Diferentes programadores pueden trabajar en el mismo proyecto sin confusiones constantes por estilos de codificación diferentes.

3. Mantenibilidad:

Es más fácil conservar el código cuando se siguen las reglas. Si alguien más tiene que tomar el proyecto, o si vuelves a tu propio código después de un tiempo, seguir las normas hace que la tarea sea mucho más sencilla.

En definitiva, seguir los estándares nos ayudará a construir un software más robusto, fiable y no ayudará a prevenir problemas futuros

El no seguir los estándares tiene muchos inconvenientes

1. Confusión:

Sin reglas, el código puede volverse un caos. Diferentes estilos y prácticas pueden generar confusión, haciendo que el proceso de desarrollo sea más lento.

2. Errores y Bugs:

La falta de consistencia puede dar lugar a errores difíciles de encontrar. Un pequeño malentendido puede llevar a errores que son difíciles de corregir.

3. Problemas en la Colaboración:

No seguir reglas hace que trabajar en equipo sea complicado. Integrar el trabajo de diferentes programadores se convierte en un desafío cuando no hay reglas comunes.

Todos estos problemas crean falta de productividad y sobrecostos no deseados porque se perderá tiempo en repararlo si da problemas, se tardará en poder utilizarlo, será más difícil modificarlo si se queda anticuado.

7 CONCLUSIONES:

Para que un software funcione correctamente, es importante entender bien los lenguajes, paradigmas y estándares. Es como aprender un idioma que le hable tanto a las personas como a las computadoras, así se pueden hacer programas que tengan utilidad. Los lenguajes de programación son como herramientas que ayudan a escribir un código claro y eficiente, y conocer varios lenguajes te da más opciones para resolver distintos problemas. Los paradigmas son como enfoques para resolver cosas, haciéndolo más fácil y mantenible. Y seguir estándares es tener reglas comunes que ayudan a trabajar en equipo y sin problemas y a que el código se mantenga bien con el paso del tiempo. En resumen, entender y usar todo esto es muy importante para hacer programas

que sean buenos y que se adapten a los cambios tecnológicos que están en constante evolución.

8- REFERENCIAS:

Fuentes utilizadas en la elaboración del informe. (Webgrafía)

<https://platzi.com/blog/paradigmas-programacion/>

<https://www.diarlu.com/lenguajes-de-programacion/>

<https://www.linkedin.com/pulse/paradigmas-de-programaci%C3%B3n->

<https://www.inesem.es/revistadigital/informatica-y-tics/estandares-web/>

<https://mosaic.uoc.edu/ac/le/es/m1/ud4/index.html>

<https://www.genbeta.com/desarrollo/diferencias-entre-paradigmas-de-programación>

<https://www.bing.com/search?q=Bing+AI&showconv=1&FORM=hpcodx>

<https://www.stackscale.com/es/blog/lenguajes-programacion-mas-populares/>

<https://web.rramirez.com/estandares-basicos-de-programacion/>