

Informe de Lenguajes, Paradigmas y Estándares de Programación

Introducción

En el mundo de la programación hay varios aspectos a tener en cuenta tales como: los **lenguajes de programación**, los cuales nos permiten comunicarnos con nuestros ordenadores con la finalidad de crear algún software, cabe destacar que cada lenguaje es diferente y poseen diferentes finalidades; los **paradigmas**, los cuales nos indican cómo se deben organizar los programas, con el propósito de ayudarnos con la organización o la resolución de problemas; y los **estándares** traen la importancia de la organización, con el fin de mantener la consistencia en un código y que sea más sencillo de usar.

Tipos de lenguajes de programación

En el mundo de la programación se pueden encontrar diferentes lenguajes de programación que a la vez se organizan en diferentes subtipos. Esta organización se divide en lenguajes de **alto, medio y bajo nivel** de los cuales hablaré ahora.

- **Alto nivel:**

Estos lenguajes se caracterizan por ser los más “sencillos” y accesibles para las personas. Se centran en aproximarse al lenguaje humano, es decir, alejarse del lenguaje máquina, el cual es el que se acerca más a los ordenadores como tal (código binario), y facilitar así la creación de programas. Algunos ejemplos de este tipo de lenguaje son:

- ❖ *Python*: Este lenguaje es bastante conocido debido a la facilidad de su uso, utilizado en diferentes campos como la automatización. Es descrito como un lenguaje adecuado para principiantes.

- ❖ *Java*: Utilizado para el desarrollo de aplicaciones. Este lenguaje se caracteriza por su capacidad de poder ejecutarse desde diferentes plataformas haciéndolo muy accesible.
 - ❖ *Javascript*: Siendo un lenguaje diferente a Java, es utilizado para la programación web. Es ejecutado desde la consola del usuario y añade interacciones en páginas web.
 - ❖ *PHP*: Lenguaje utilizado exclusivamente para el desarrollo web, para agregar dinamismo.
- **Medio nivel**: Estos lenguajes se caracterizan por tener características de ambos lados de los tipos de lenguaje, es decir, posee características de lenguajes de alto y bajo nivel. Se define como un lenguaje “equilibrado”, con características que facilitan la programación y el control sobre el hardware. Algunos ejemplos de este tipo de lenguaje son:
 - ❖ *C* : Es uno de los lenguajes más utilizados y antiguos. Como lenguaje de medio nivel, a pesar de acercarse más al nivel bajo , ofrece un gran control sobre el hardware. Nos permite la manipulación de la memoria y nos da acceso al sistema operativo.
 - ❖ *C++*: Como su nombre indica, es una extensión de “C”. Este lenguaje permite trabajar con ambos tipos de lenguaje, de alto y bajo nivel. Es principalmente utilizado para el desarrollo de juegos y aplicaciones de alto rendimiento a la vez que para softwares de sistemas.
 - ❖ *Rust*: Es un lenguaje relativamente moderno, utilizado para la seguridad, la ocurrencia y el rendimiento. Como los otros dos lenguajes, combina características de los lenguajes de alto y bajo nivel. Este lenguaje nos da control sobre la memoria, previendo accesos no seguros a la memoria y las fugas de la misma.
- **Bajo nivel**: Estos lenguajes se caracterizan por ser los más alejados al lenguaje humano, es decir, están más cerca del lenguaje de nuestros ordenadores , lenguaje máquina, siendo así menos entendible y menos portables de un dispositivo a otro. Este lenguaje se caracteriza por su capacidad para controlar el hardware. Algunos ejemplos de estos lenguajes son:

- ❖ *Lenguaje ensamblador*: Sirve para representar las instrucciones del procesador de una computadora utilizando códigos de operación. Algunos ejemplos de lenguaje ensamblador son: ARM y MIPS assembly.
- ❖ *Fortran*: Es un lenguaje creado para cálculos numéricos y está enfocado a tareas de bajo nivel en las cuales se necesita precisión sobre los datos
- ❖ *ADA*: Es un lenguaje diseñado para sistemas de defensa ofreciendo un gran control sobre el hardware.

Paradigmas de programación

Los paradigmas de programación son las diferentes maneras de afrontar problemas en la programación a la hora de hacer proyectos. Cada paradigma es diferente e independiente el uno que del otro, con el fin de adaptarse mejor a diferentes tipos de problemas. Los principales paradigmas son:

- ***Imperativo*** (Programación imperativa): Se utilizan fórmulas las cuales alteran el estado de un programa. Este paradigma se considera el más antiguo. Consiste en dar instrucciones a un ordenador para que nos muestre ciertos resultados en diferentes momentos programados. Algunos lenguajes son:
 - ❖ *C++*: Junto a C. Combina la programación imperativa con la orientación a objetos. Permite la herencia entre dispositivos. Posee una sintaxis sencilla. Nos da acceso directo a la memoria.
 - ❖ *Java*: Es muy portable y seguro. Utiliza clases y objetos con el fin de organizar.
 - ❖ *Pascal*: Es un lenguaje que se enfoca en la claridad y que el código se entienda. También nos proporciona secuencias, bucles y selecciones.
- ***Declarativa*** (*Programación declarativa*): A diferencia de los lenguajes imperativos los cuales nos dicen como hacer las cosas, el declarativo se centra en que se debe hacer sin prestar atención al control ni a los detalles de cómo hacerlo. Este paradigma posee ciertas características las cuales son que el código se puede reutilizar en diferentes partes del programa, también

suele tener menor errores, se suele optimizar automáticamente y posee un código más claro. Algunos ejemplos de este paradigma son:

- ❖ *Haskell(Programación funcional)*: Es una función para sumar elementos sin tener que decir como se debe hacer.
 - ❖ *SQL*: Nos permite recuperar datos de una base de datos sin especificar cómo se deben recuperar dichos datos.
 - ❖ *Minizinc*: Se fija un objetivo y ciertas restricciones sin importar lo que no se ha mencionado
- **Orientada a objetos**: Este tipo de programación se centra en organizar o modificar el código alrededor de entidades que es como se representan los objetos. Con este paradigma, los objetos pueden ocultar sus datos, y permite que los objetos se puedan implementar en una interfaz común, y podemos representar objetos abstractos y fomenta la creación de módulos. Algunos ejemplos son:
 - ❖ *Java*: Este lenguaje garantiza el control de acceso a los datos, y representa las acciones de los objetos.
 - ❖ *Python*: Este lenguaje nos permite controlar el acceso a datos y también nos permite definir un método.
 - **Funcional**: Este paradigma nos permite la resolución de problemas via operaciones matemáticas y modelar la lógica del programa, también, se compone de funciones las cuales se basan en tomar los datos de entrada y en base a eso, producir resultados. Las funciones se pueden pasar de unas a otras, nos describe qué hacer en lugar de como hacerlo (como la programación declarativa) , y se usa principalmente para tareas repetitivas. Algunos ejemplos de este tipo de programación son:
 - ❖ *Javascript*: Utiliza expresiones mas claras y concisas, a la vez que de alto nivel y se basa en introducir “argumentos” y que nos devuelva respuestas.
 - ❖ *Lisp*: Permite definir estructuras de lenguaje y utiliza listas con el fin de representar datos. También cabe destacar que es muy flexible y dinámico.

- **Lógica:** Este paradigma se basa en la lógica matemática construyéndose en torno a relaciones lógicas.

Importancia de seguir los estándares

Hoy en día, a los “estándares de programación” se les conoce como una serie de reglas y procedimientos a seguir a la hora de programar, marcados generalmente por desarrolladores más avanzados que los novatos. Esto se llevó a cabo con la finalidad de ayudar a los nuevos programadores integrándoles unos estándares básicos. Estos estándares también nos indican cómo debe de ser el código para que sea mantenible y comprensible para otros. Y estos estándares son necesarios si se hacen proyectos, con el fin de superar o al menos igualar dichos estándares. Algunos de estos estándares son:

- **Facilidad para leer el código y comprensión:** Aquellos que lean el código, podrían detectar errores en el código y corregirlos para mejorarlo. Estos errores se detectaron gracias a la legibilidad del código, el cual los más recomendables serían los de alto nivel por su proximidad al lenguaje humano, logrando así, como dije antes, un mayor entendimiento del código. Así, se comprendería mejor el código.
- **Estabilidad:** Para que un código tenga estabilidad, debe estar bien ordenado y estructurado. De esta manera, se tendrá un mayor control sobre cada parte del mismo a pesar de realizar cambios sobre él. Si se siguen los estándares, se logrará aplicar más cambios al código y nos aportará más flexibilidad a la hora de realizar cambios.
- **Calidad del código:** La calidad del programa se puede mejorar de diferentes maneras, como preguntando a un programador senior, o incluso a una IA. Si queremos entender cómo se mejora la calidad, hay que seguir las pautas que se marcan para alcanzar dichas mejoras ,como: comprender cómo funciona el programa/software para tener un margen más amplio, diseñar una estructura sólida para que a la hora de editarlo, no se derrumbe., intentar corregir errores diariamente,... También, junto a la calidad recae la seguridad del código, ya que los mejores se utilizan en diferentes campos importantes de

nuestro día a día como en la medicina. Así, se asegura un software seguro y estable.

- **Consistencia:** Este punto se centra en que no se cambie la forma de escribir en el código. Es decir, crear un código más “simple” para todos los usuarios, y también ayuda a nuevos usuarios a comprenderlo sin que necesiten adaptarse a los cambios constantes.

Por lo que adherirse a los estándares nos aporta mayor flexibilidad a la hora de programar, es decir, nos permite, a todos, incluyendo a los nuevos usuarios, utilizar diferentes códigos o editarlos. Es necesario en el mundo en el que vivimos adherirse a estos nuevos estándares, por un lado para “continuar el legado” y continuar avanzando, y al mismo tiempo, para mejorar la seguridad o los sistemas de bases de datos para garantizar una mayor seguridad de los mismos y evitar las caídas de los programas que se hagan en el futuro. También cabe destacar que no adherirse a estos nuevos estándares nos afecta a todos en el fondo. Por un lado, porque se contradiría lo que mencioné anteriormente y se empeoraría la calidad de los programas y softwares. Por otro lado, afectaría individualmente a la persona que no se adhiera quedándose atrás con las nuevas tecnologías y perdiendo oportunidades laborales. Al mismo tiempo, también nos afectaría a todos si uno deja de adherirse, es posible que otros lo hagan también ralentizando el progreso alcanzado.

Conclusión

En conclusión, con el avance del mundo, avanzan las nuevas tecnologías y con ellas los programas y métodos utilizados en las mismas. Es más recomendable utilizar lenguajes de alto nivel que los de bajo nivel, por sus numerosos beneficios a la hora de programar. También hay que variar respecto a los paradigmas que utilizamos, para acostumbrarnos a adaptarnos a diferentes ambientes y situaciones que los requieran, a la vez que hay que adaptarse a nuevos programas para el mismo propósito. En adición, también hay que adherirse a los estándares de programación, ya que esto es necesario para continuar progresando en la programación y no quedarnos en el paro en el futuro. Los estándares nos ayudan a mejorar en este

campo y a adaptarnos o ayudar a otros a adaptarse a otros nuevos campos. Por lo que hay que adherirse a los estándares para continuar progresando en la programación.

Fuentes

<https://desarrolloweb.com/articulos/2358.php>

<https://conceptobasicodecomputacion.weebly.com/lenguaje-de-alto-medio-y-bajo-nivel.html>

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

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

https://es.wikipedia.org/wiki/Paradigma_de_programaci%C3%B3n

https://keepcoding.io/blog/paradigmas-de-programacion/#Tipos_de_paradigmas_de_programacion

<https://cognosonline.com/co/blog/que-son-paradigmas-de-programacion/>

<https://es.parasoft.com/blog/an-ounce-of-prevention-software-safety-security-through-coding-standards/>

<https://blog.thedojo.mx/2021/10/05/estandares-de-calidad-en-el-software.html>