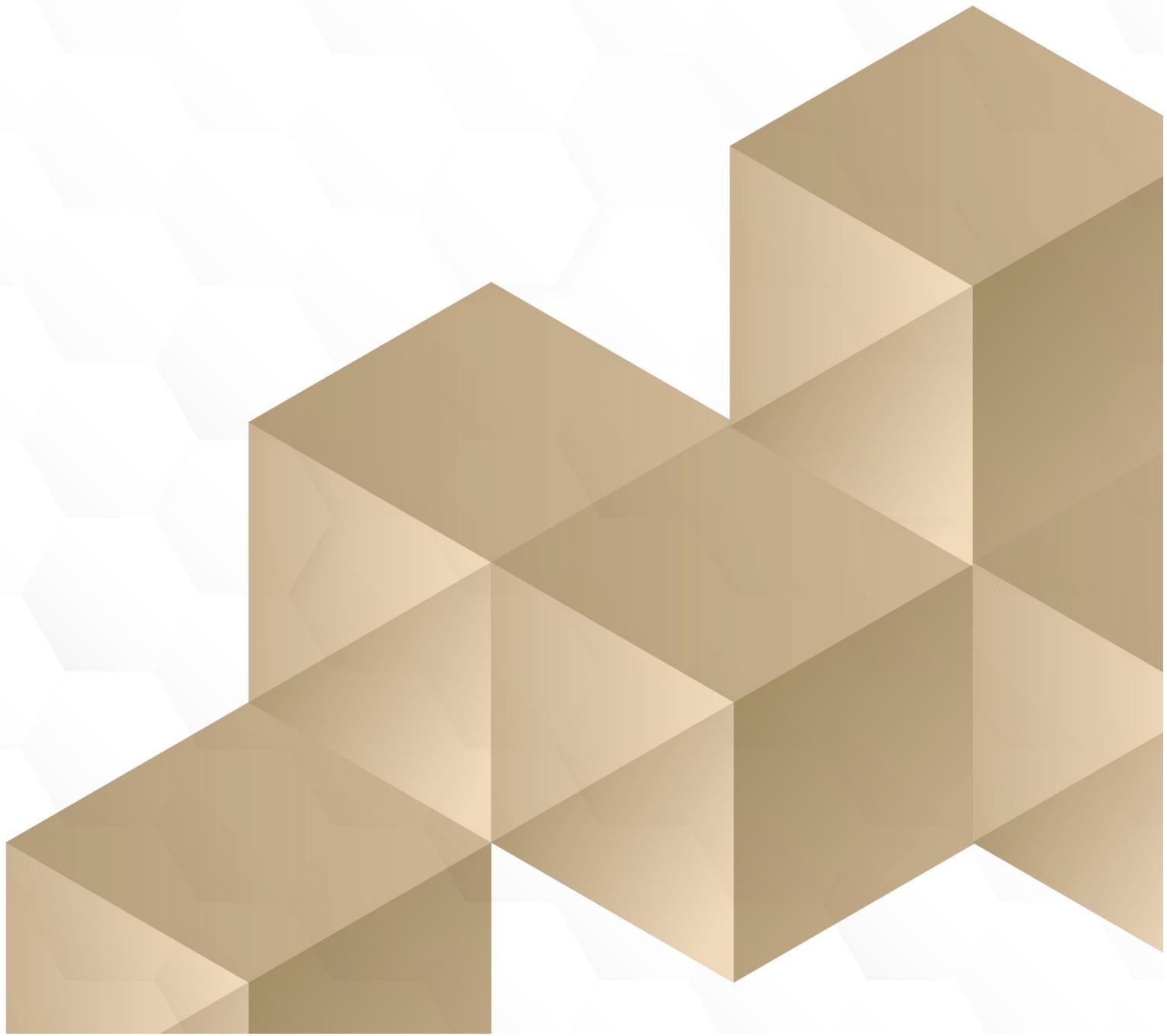


➤ Fundamentos de programación

INFOTEC. (2024)



Fundamentos de programación

Introducción a la lógica de la programación y pseudocódigo

La lógica de la programación es el fundamento sobre el cual se construyen todos los conceptos y técnicas en el mundo de la programación. Se trata de la capacidad de pensar de manera estructurada y lógica para resolver problemas utilizando instrucciones precisas y secuenciales. Explicado en otras palabras, es el trabajo que implica descomponer un problema complejo en pasos más simples y comprensibles que pueden ser interpretados y ejecutados por una computadora.

Una parte fundamental de la lógica de la programación es comprender y saber utilizar pseudocódigo. Este es un método de representación de algoritmos donde utilizamos un lenguaje informal combinando elementos del lenguaje humano y de la programación. Nos permite describir los pasos necesarios para resolver un problema de manera clara y concisa, sin estar ligado a la sintaxis de un lenguaje de programación específico.

Al utilizar el pseudocódigo podrás planificar y diseñar algoritmos antes de escribir código en un lenguaje de programación concreto. Esto te permitirá visualizar la estructura y la lógica de los programas antes de comenzar a codificarlos, lo que te puede ayudar a planear, evitar errores y facilitar la resolución de problemas.

Cabe recalcar que el pseudocódigo no es un lenguaje de programación en sí mismo, sino más bien, una herramienta para expresar algoritmos de una manera comprensible y universal. Puede ser utilizado por programadores con todos los niveles de experiencia y en una variedad de contextos, desde la enseñanza de conceptos básicos de programación hasta la planificación de sistemas complejos.

En resumen, comprender la lógica de la programación y el pseudocódigo es el primer paso crucial para convertirte en un programador hábil y eficiente. Esto, te



proporcionará la base sobre la cual construirás todas las demás habilidades y técnicas en el campo de la programación, permitiéndote abordar problemas de manera sistemática y encontrar soluciones efectivas, utilizando algoritmos bien diseñados.

Variables y tipos de datos

En programación, las variables son contenedores que almacenan información, que puede ser modificada durante la ejecución de un programa. Cada variable, tiene un nombre único y un tipo de dato asociado, el cual define qué tipo de valores se pueden almacenar, como números enteros, decimales, cadenas de texto o valores booleanos. Las variables son fundamentales para el desarrollo de programas, ya que permiten manipular y trabajar con datos de manera dinámica.

Los tipos de datos en programación, representan los diferentes tipos de valores que pueden ser almacenados y procesados por un programa.

Algunos de los tipos de datos más comunes son:

- Enteros:** representan números enteros sin decimales
- Flotantes:** representan números decimales
- Cadenas de texto:** representan secuencias de caracteres
- Booleanos:** representan valores de verdadero o falso



Es importante que comprendas los diferentes tipos de datos disponibles en programación y cómo se utilizan para almacenar y manipular información en un programa. Por ejemplo, puedes utilizar las variables de tipo entero para almacenar números como la edad de una persona o el número de elementos en

una lista, o utilizar variables de tipo cadena los cuales se utilizan para almacenar valores de texto, como nombres o direcciones.

Además de los tipos de datos básicos, existen los tipos de datos compuestos, por ejemplo: los arreglos, listas, diccionarios o estructuras que permiten almacenar colecciones de datos relacionados entre sí. Estos tipos de datos compuestos son útiles para trabajar con conjuntos de datos más complejos y estructurados en un programa.

Comprender los conceptos de variables y tipos de datos, te proporcionará la base sobre la cual se construyen las demás estructuras en un programa. Y te permitirá manipular y trabajar con datos de manera efectiva, lo que es fundamental para el desarrollo de un software eficiente y robusto.

Operadores

Los operadores en programación: son símbolos especiales los cuales se utilizan para realizar operaciones sobre variables y valores. Estas operaciones pueden ser matemáticas, relacionales o lógicas, y permiten manipular y transformar los datos en un programa, de acuerdo con las necesidades que requieras.

Los operadores aritméticos: se utilizan para realizar operaciones matemáticas básicas, como la suma, la resta, la multiplicación y la división, sobre valores numéricos. Además de estos operadores básicos, cada lenguaje de programación ofrece operadores para realizar operaciones más avanzadas, como el módulo, la potencia, etc., que son útiles en una variedad de contextos de programación.

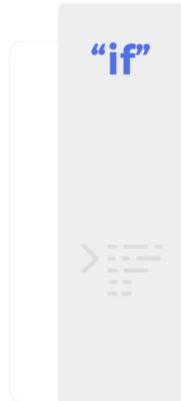


Los operadores relacionales: se utilizan para comparar dos valores y determinar la relación entre ellos. Estos operadores devuelven un valor booleano por medio del cual, se indica si la relación especificada es verdadera o no. Los operadores relacionales más comunes incluyen los signos: igual que (==), diferente que (!=), mayor que (>), menor que (<), mayor o igual que (>=) y menor o igual que (<=). Estos operadores son fundamentales para la toma de decisiones en programación, ya que permiten comparar valores y ejecutar diferentes acciones, en función de los resultados de las comparaciones.

Los operadores lógicos: se utilizan para combinar o modificar expresiones lógicas que lleven a producir un resultado booleano, el cual puede ser utilizado para controlar el flujo de la ejecución de un programa. Los operadores lógicos más comunes son el AND lógico (&&), el OR lógico (||) y el NOT lógico (!). Son utilizados para construir condiciones más complejas y expresiones booleanas en un programa de tal forma, que se puedan combinar múltiples condiciones y determinar si se cumplen todas ellas (AND), alguna de ellas (OR) o ninguna de ellas (NOT).

Estructuras de control

En programación, las estructuras de control son la manera en que se organizan y controlan las instrucciones dentro de un programa para determinar su flujo de ejecución. Estas estructuras permiten tomar decisiones, repetir acciones y ejecutar bloques de código en función de ciertas condiciones, lo que proporciona flexibilidad y control sobre el comportamiento del programa.



"if"

Una de las estructuras de control más comunes es la instrucción "if", que permite ejecutar un bloque de código si una condición especificada es verdadera. Esta estructura se utiliza para realizar pruebas condicionales y tomar decisiones en función de los resultados obtenidos en dichas pruebas. También existen variantes como el "if-else", que permite ejecutar un bloque de código alternativo si la condición es falsa, y el "if-else... if-else...", con el que se puede encadenar múltiples condiciones y bloques de código.

Otra estructura de control importante es el bucle "for", este permite repetir en un bloque de código, durante un número específico de veces. Este bucle se utiliza cuando se conoce de antemano el número de iteraciones que se deben realizar.

"for"

"while"

Por otro lado, el bucle "while", se utiliza cuando el número de iteraciones no es conocido y deberá repetirse mientras una condición especificada sea verdadera.

Además de estas estructuras básicas, también existen otras estructuras de control más avanzadas, como el bucle "**do-while**": es similar al "while", pero se diferencia de éste, porque el proceso de evaluar la condición se realiza después de ejecutar el código del bloque, y la estructura "**switch**", que permite seleccionar uno de varios bloques de código para ejecutar, en función del valor de una variable.

Debes entender las estructuras de control y las ventajas que ofrecen una sobre la otra, ya que son fundamentales en la programación pues te permiten controlar el flujo de la ejecución con la finalidad de escribir un código eficiente y funcional.

Funciones y métodos

Las funciones y métodos son bloques de código reutilizables, que llevan a cabo una tarea específica dentro de un programa. Permiten modularizar el código, lo que significa dividir un programa en partes más pequeñas y manejables, las cuales cumplen tareas específicas. Esto facilita la organización, la depuración y la reutilización del código, lo que contribuye a la eficiencia y la legibilidad del programa.

Una función es un bloque de código que se puede llamar desde otras partes del programa para ejecutar una tarea específica. Las funciones pueden aceptar datos de entrada, llamados parámetros, realizar cálculos o acciones en esos datos y devolver un resultado. Son fundamentales para la modularidad y la abstracción en programación, ya que permiten encapsular la lógica de una tarea y ocultar los detalles de su implementación.



Los métodos, en cambio, son funciones que están asociadas a objetos o instancias de una clase. Estos métodos pueden acceder a los datos del objeto y manipularlos de acuerdo con la funcionalidad específica del objeto. Por ejemplo, un método de un objeto de la clase "Automóvil" puede ser "acelerar" o "frenar", por medio de los cuales, se modifica el estado del automóvil.



Al escribir funciones y métodos, es importante seguir las mejores prácticas de programación, como utilizar nombres descriptivos, evitar la duplicación de código y mantener las funciones y métodos cortos y enfocados en una tarea específica. Esto facilita la comprensión del código y la identificación de posibles errores durante el proceso de depuración.

Las funciones y métodos son componentes fundamentales en la construcción de programas. Su utilización permite modularizar y reutilizar la lógica de programación, con el fin de escribir un código limpio, eficiente y fácil de mantener en proyectos de desarrollo de software.

Elaboró

Contenido: Marco Antonio Ruiz Valdez

