**INTRODUCCIÓN A LA PROGRAMACIÓN**

MANUAL DEL LENGUAJE JAVA

Autor: Hugo Pelayo Aseko

Fecha: 1 de abril de 2023

Asignatura: Programación

Este manual cubre aspectos básicos y generales relacionados con la programación, con un enfoque alrededor del lenguaje de programación Java. Se empieza por introducir la programación en general abstrayendo de los detalles de los lenguajes de programación.

Al final del manual se agrega un apéndice donde se pueden consultar las definiciones de términos relevantes al manual que sean de difícil comprensión.

Índice

[Introducción 4](#_Toc131432926)

[Apéndice 5](#_Toc131432927)

[Bibliografía 6](#_Toc131432928)

[Bibliografía2 6](#_Toc131432929)

# Introducción

Se entiende por software a aquel sistema dotado de componentes lógicos que posibilitan la realización de tareas bien definidas, constituye entonces el conjunto de instrucciones que debe ser ejecutado por el **hardware**.

El software por lo general se desarrolla utilizando lenguajes de programación de alto nivel. Son lenguajes que utilizan una estructura semántica más próxima a los lenguajes humanos y, por tanto, más fácil de entender para los programadores. Debido a que el hardware entiende únicamente el lenguaje máquina, que está formado exclusivamente por cadenas de unos y ceros, el software escrito en lenguaje de alto nivel debe ser compilado primero antes de ser ejecutado por nuestro hardware, interpretado o, en el caso de algunos lenguajes de alto nivel, ambos a la vez.

Aunque sea poco común, para el desarrollo de software se utiliza también el lenguaje ensamblador, que es un lenguaje que constituye una capa inmediatamente superior al lenguaje máquina. Una de las diferencias principales entre estos dos lenguajes (que también motiva su uso) y los de alto nivel se basa en que estos últimos abstraen al programador de la arquitectura para la cual se desarrolla la aplicación, de tal modo que podemos desarrollar aplicaciones en lenguajes de alto nivel sin tener mucho conocimiento sobre la arquitectura destino. Sin embargo, este no es el caso para el lenguaje ensamblador que al igual que el lenguaje máquina, va de la mano con la arquitectura.

El software por lo general se clasifica en tres tipos acorde a las funciones que a las cuales este ha de realizar, tenemos entonces el software de sistema, software de programación y el software de aplicación.

El software de sistema está destinado a administrar el sistema informático, esto es, proveer al programador de una interfaz de alto nivel o conjunto de herramientas para su diagnóstico y mantenimiento aislando al mismo tiempo al programador de los detalles internos del sistema.

El software de programación es el conjunto de aplicaciones que permiten al programador desarrollar programas informáticos, es decir, desarrollar software, en este grupo se incluyen los editores de texto, los compiladores, entornos de desarrollo integrados (IDE), entre otros. Este software dota al programador de herramientas prácticas para el desarrollo de aplicaciones permitiendo el uso de lenguajes de programación tanto de alto nivel como de bajo nivel mencionados con anterioridad.

Por último, tenemos el software de aplicación. Se trata de un conjunto de software destino a un usuario final para la realización de tareas específicas que pueden ser automatizadas o realizadas mediante la asistencia del software. Se incluye dentro de este grupo las aplicaciones ofimáticas como el Microsoft Word, aplicaciones para la manipulación de bases de datos como MySQL Workbench, entre otros.

Como ya se había mencionado anteriormente todo software consta de programas. Existe cierta confusión entre programa y el software en sí. Se considera programa a una secuencia de instrucciones u órdenes escritas en un lenguaje de programación cualquiera que han de ser interpretadas por la computadora para realizar una tarea específica, esta secuencia de pasos ha de ser finita, bien definida y concisa, se conocen generalmente con el nombre de algoritmo.

El conjunto de expresiones que componen un programa se conoce como el código del programa. Este se escribe en lenguajes de programación que sigue un paradigma de programación que se adapte a las necesidades del programador. Los paradigmas de programación más comunes se clasifican en dos grandes grupos: el imperativo y el declarativo. En el paradigma imperativo el programador instruye a la máquina a través del código a cómo cambiar su estado; en el paradigma declarativo se instruye a la máquina sobre las propiedades del resultado esperado de ciertas operaciones y no a cómo realizarlas.

Entre las herramientas más comunes para el diseño de algoritmos destacan los diagramas de flujo y el pseudocódigo:

El pseudocódigo, también conocido con el nombre de lenguaje algorítmico, es una forma de describir con lenguaje natural el flujo de ejecución de un algoritmo o programa de forma compacta. Normalmente utiliza las construcciones de un lenguaje de programación real, pero está diseñado para la interpretación por parte de humanos ya que abstrae las particularidades de los lenguajes de programación facilitando el enfoque en la lógica y funcionamiento del programa. Por lo general se utiliza el pseudocódigo en libros, textos científicos que exponen algoritmos y también es usado para la planificación de algoritmos para programas complejos, esto permite, por ejemplo, descubrir posibles errores de lógica en nuestro algoritmo antes de proceder a su implementación. Como se ha mencionado con anterioridad, el pseudocódigo sigue una sintaxis arbitraria y no existe cierta convención ya que el objetivo principal es simplemente exponer la solución a un problema de la forma más sencilla posible. Sin embargo, cabe recalcar que ciertos IDE como PSeInt utilizan pseudocódigo con sintaxis propia para el desarrollo de algoritmos.

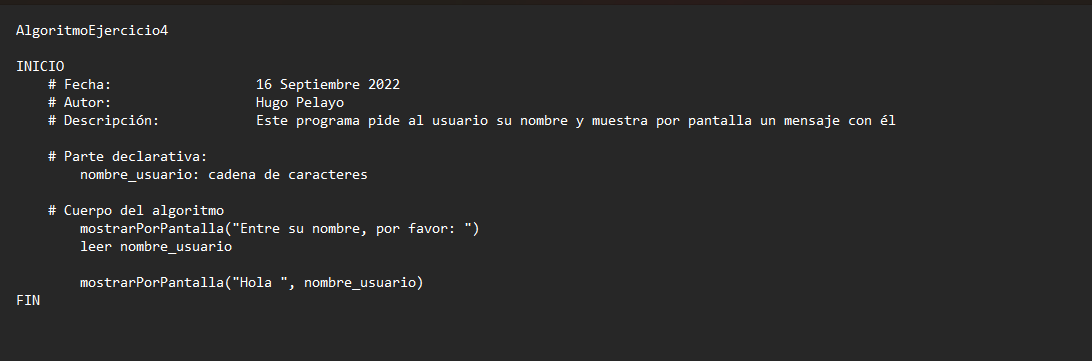


Figura 1. Ejemplo de pseudocódigo. Fuente: creación propia

El cuerpo del pseudocódigo está constituido normalmente por estructuras de control y estructuras e iterativas o bucles. Las estructuras de control se dividen principalmente en dos grupos, secuencias donde tenemos un flujo de ejecución de sentencias continua o, pueden ser selectivas, en cuyo caso el flujo de ejecución de las sentencias varía acorde al resultado de evaluar ciertas expresiones lógicas.

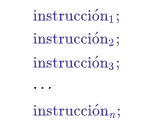
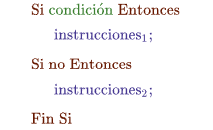


Figura 3. Estructura selectiva con alternativa. Fuente: Wikipedia

Figura 2. Estructura secuencial. Fuente: Wikipedia

El otro grupo de grupo elementos está constituido por las estructuras iterativas, en las cuales se ejecuta un cierto grupo de instrucciones mientras se cumpla cierta condición, tenemos entonces: el **bucle mientras** donde ejecutamos su cuerpo mientras una condición se satisfaga; tenemos el **bucle hacer** donde, a diferencia del bucle anterior, se ejecuta el cuerpo al menos una vez, cabe destacar que en el caso del bucle anterior si la condición especificada en la cabecera del bucle no se cumple ni en la primera iteración entonces nunca se entra al cuerpo de este; para finalizar tenemos el **bucle repetir** donde ejecutamos un grupo de sentencias un número finito de veces.

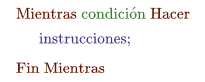


Figura 4. Formato del bucle mientras. Fuente: Wikipedia

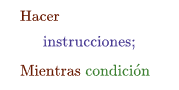


Figura 5. Formato del bucle hacer. Fuente: Wikipedia

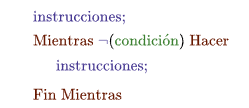


Figura 6. Formato del bucle repetir. Fuente: Wikipedia

# Apéndice

* Algoritmo:
* Expresión:
* Expresión lógica:
* Hardware:
* IDE:
* Paradigma:

# Bibliografía

|  |  |
| --- | --- |
| [1] | Wikipedia, «Software,» Wikipedia, 2023 Marzo 31. [En línea]. Available: https://es.wikipedia.org/wiki/Software. [Último acceso: 2023 Abril 3]. |