

Algoritmos

Algoritmos	1
¿Qué aprenderás?	2
Introducción	2
Algoritmo	3
Diagrama de flujo	4
Símbolos de un diagrama de flujo	5
Pseudocódigo	6
Enfrentándose a un problema	7



¡Comencemos!

¿Qué aprenderás?

- Codificar un programa en Python utilizando diagramas de flujo con iteraciones para la implementación de un algoritmo que resuelve un problema.

Introducción

¿Cómo hacen los programadores para resolver un problema? No siempre un programador es capaz de pensar en una solución y plasmarla en código de una vez, ya que normalmente es un proceso iterativo en el cual durante varios ciclos de desarrollo se implementan distintas alternativas de solución.

Una manera de poder llegar a estas soluciones es mediante los algoritmos, los que corresponden a una solución lógica que permite a los programadores organizar el proceso para llevar a cabo la solución para finalmente plasmarla en su código. Los algoritmos son soluciones agnósticas, es decir, pueden ser implementadas en distintos lenguajes de programación.

Es por esto que en este capítulo, conoceremos los conceptos de algoritmo y diagrama de flujo, lo que nos permitirán codificar programas para resolver problemas.

Algoritmo

Un algoritmo no es más que una serie de pasos finitos y ordenados para resolver un problema. Por ejemplo, en el siguiente algoritmo veremos que se define cómo un estudiante de colegio se enfrenta al día a día.

Entendamos el algoritmo:

1. Primero se verifica qué día es.
2. Si es día de semana hay que prepararse para ir al colegio, y dependiendo de si llueve o no, se irá al colegio en bicicleta o en bus.
3. Si es sábado, se irá a jugar fútbol, y dependiendo de si el papá ya se fue a su trabajo, se irá con el papá o en taxi.
4. En otro caso, es decir, si es Domingo, el niño va a la piscina.

Cuando generamos un algoritmo en el computador, debemos especificar paso a paso qué es lo que el computador debe realizar. **Si el algoritmo utilizado para construir nuestro programa omite algún paso o no sigue el orden correcto, nuestro programa fallará o no cumplirá con el resultado esperado.** Es por ello que saber el algoritmo, es decir, conocer cada paso para crear nuestro programa, es de suma importancia.

Existen distintas formas de implementar un algoritmo, estas son:

- Representarlo mediante un diagrama de flujo
- Escribirlo en pseudocódigo
- Escribirlo directamente en un lenguaje de programación

Diagrama de flujo

El diagrama de flujo es una representación gráfica de los pasos detallados en un algoritmo, ya que permite visualizarlo y de esta forma, reducir su complejidad.

A continuación, se presenta otro algoritmo, para el caso en que queremos arreglar una lámpara, donde el proceso algorítmico conlleva los siguientes pasos.

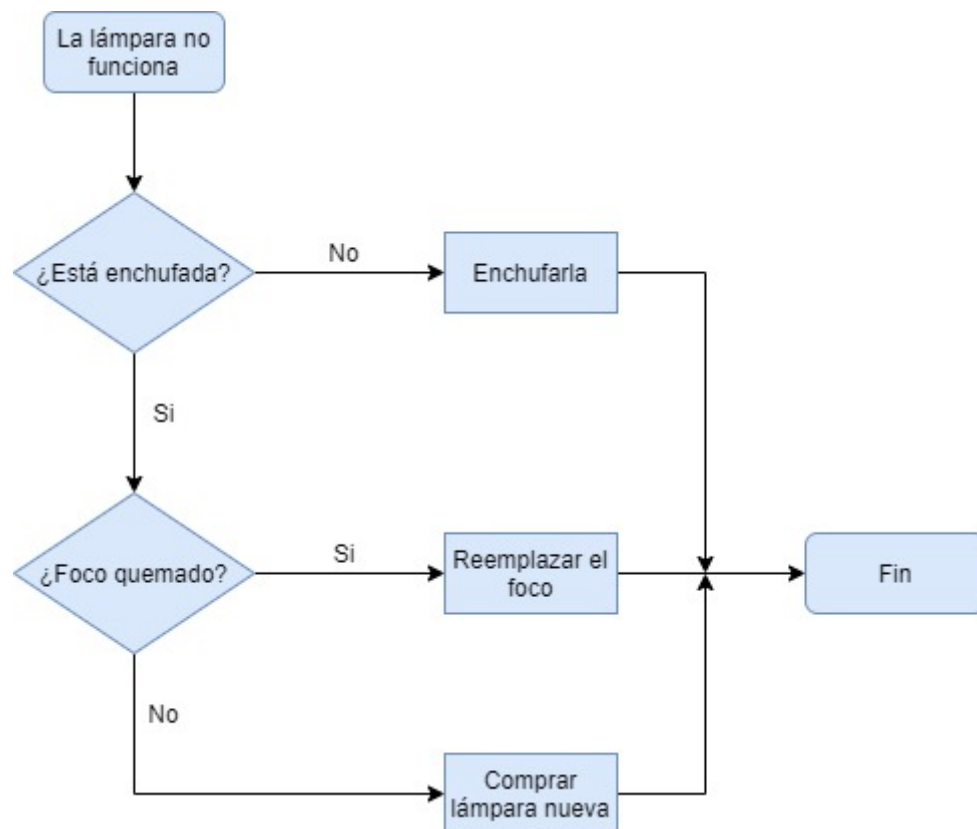


Imagen 01. Diagrama de una lámpara.

Fuente: Desafío Latam.

En algunos diagramas, como en el anterior, el fin es implícito; al llegar al último paso, se supone que el algoritmo concluye. Esta distinción es necesaria para destacar que existen procesos donde no hay fines explícitamente declarados.

Símbolos de un diagrama de flujo

Cada uno de los símbolos en un diagrama de flujo representa alguna acción a realizar en el algoritmo.

- **Inicio y fin del programa:** por defecto, todo algoritmo debe tener una serie de elementos finitos, los que se declaran mediante el inicio y el fin del programa.
- **Datos de entrada y salida:** cada paso que se genera dentro del algoritmo implica tomar algún dato y devolver otro dato.
- **Procesos (la instrucción que le damos a la máquina):** definición de los pasos a seguir en un algoritmo.
- **Decisiones:** eventualmente, el flujo lógico podrá encauzar los siguientes pasos en base a la resolución de una decisión.

La siguiente imagen muestra de forma gráfica los símbolos que componen un diagrama de flujo:

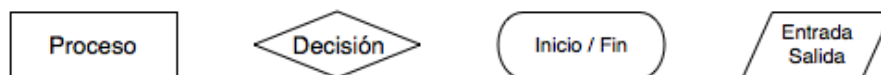


Imagen 02. Símbolos de un diagrama.
Fuente: Desafío Latam.

Pseudocódigo

Otra forma de representar un algoritmo es mediante el pseudocódigo, el cual consiste en una serie de frases que describen el flujo:

```
Algoritmo Suma
Leer valor1
Leer Valor2
Mostrar valor1 + valor2
FinAlgoritmo
```

En pseudocódigo, se utiliza la instrucción `leer` para especificar que el usuario tiene que ingresar un valor y mostrar para imprimir el valor en la pantalla.

El pseudocódigo, permite pensar en términos independientes al lenguaje de programación y permite concentrarnos en describir lo que estamos tratando de hacer y los pasos necesarios en lugar de cómo lograrlo.

Para programar aprenderemos que es vital saber cómo comunicarle al computador la secuencia de pasos a seguir, y que escribir pseudocódigo y diagramas de flujo ayuda a reforzar esta habilidad.

Si bien la especificación del pseudocódigo no es formal, existen algunos programas que nos permiten escribir pseudocódigo y ejecutarlo, como **PSEINT**. Si quieres experimentar con el pseudocódigo, puedes hacerlo descargando el programa de forma gratuita en la [página oficial](#).

Enfrentándose a un problema

El desarrollo del pensamiento lógico es una habilidad **imprescindible** al momento de aprender a programar y está directamente relacionado con nuestra manera de solucionar problemas. Antes de escribir código, debemos abstraernos del código y pensar en el problema en los siguientes pasos:

1. Analizar el problema
2. Descomponer el problema en partes
3. Resolver el problema

Cuanto más desarrollemos nuestro pensamiento lógico, más rápido obtendremos soluciones a problemas cotidianos en programación, y nuestros programas harán más a menudo lo que esperamos.