**Qué es un algoritmo informático: características, tipos y ejemplos**

Un algoritmo es un procedimiento paso a paso para conseguir un fin. Es un concepto comúnmente relacionado con el ámbito de la informática. En este post, explicamos **en qué consiste un algoritmo informático** y recopilamos los principales tipos y algunos ejemplos.

Definición de algoritmo informático.

Un **algoritmo informático** es un conjunto de instrucciones definidas, ordenadas y acotadas para resolver un problema o realizar una tarea.

En programación, supone el **paso previo a ponerse a**[**escribir el código**](https://www.instagram.com/p/CGkHG_lg8xY/). Primero debemos encontrar la solución al problema (definir el algoritmo informático), para luego, a través del código, poder indicarle a la máquina qué acciones queremos que lleve a cabo. De este modo, un programa informático no sería más que un **conjunto de algoritmos ordenados y codificados** en un[**lenguaje de programación**](https://profile.es/blog/los-lenguajes-de-programacion-mas-queridos-y-mas-odiados/) para poder ser ejecutados en un ordenador.

No obstante, los algoritmos no son algo exclusivo de los ámbitos de las matemáticas, la lógica y la computación. Utilizamos numerosos algoritmos para resolver problemas en**nuestra vida cotidiana**. Algunos de los ejemplos más habituales son los manuales de instrucciones o las recetas de cocina.

Partes de un algoritmo informático

Las **tres partes de un algoritmo** son:

1. **Input (entrada).** Información que damos al algoritmo con la que va a trabajar para ofrecer la solución esperada.
2. **Proceso.** Conjunto de pasos para que, a partir de los datos de entrada, llegue a la solución de la situación.
3. **Output (salida).** Resultados, a partir de la transformación de los valores de entrada durante el proceso.

De este modo, un algoritmo informático parte de un estado inicial y de unos valores de entrada, sigue una serie de pasos sucesivos y llega a un estado final en el que ha obtenido una solución.

Características de los algoritmos

Asimismo, los algoritmos presentan una serie de**características comunes**. Son:

* **Precisos**. Objetivos, sin ambigüedad.
* **Ordenados**. Presentan una secuencia clara y precisa para poder llegar a la solución.
* **Finitos**. Contienen un número determinado de pasos.
* **Concretos**. Ofrecen una solución determinada para la situación o problema planteados.
* **Definidos**. El mismo algoritmo debe dar el mismo resultado al recibir la misma entrada.

Tipos de algoritmos y ejemplos

Existen diversas clasificaciones de algoritmos, en función de diferentes criterios. Según su sistema de signos (cómo describen los pasos a seguir), se distingue entre algoritmos **cuantitativos**y**cualitativos**, si lo hacen a través de cálculos matemáticos o secuencias lógicas. Asimismo, si requieren o no el empleo de un ordenador para su resolución, se clasifican en **computacionales** y **no computacionales**.

Pero, si nos fijamos en su **función**(qué hace) y su **estrategia**para llegar a la solución (cómo lo hace), encontramos muchos más tipos de algoritmos.

Destacamos cinco algoritmos informáticos:

Algoritmos de búsqueda

Los **algoritmos de búsqueda** localizan uno o varios elementos que presenten una serie de propiedades dentro de una estructura de datos. Existen diversos tipos de búsquedas, entre las que sobresalen:

* **Búsqueda secuencial.** En la que se compara el elemento a localizar con cada elemento del conjunto hasta encontrarlo o hasta que hayamos comparado todos.
* **Búsqueda binaria.**En un conjunto de elementos ordenados, hace una comparación con el elemento ubicado en el medio y, si no son iguales, continúa la búsqueda en la mitad donde puede estar. Y así sucesivamente en intervalos cada vez más pequeños de elementos.

Algoritmos de ordenamiento

Reorganizan los elementos de un listado según una relación de orden. Las más habituales son el orden numérico y el orden lexicográfico. Un orden eficiente optimiza el uso de algoritmos como los de búsqueda y facilitan la consecución de resultados legibles por personas y no solo máquinas.

Algunos **algoritmos de ordenamiento** son:

* **Ordenamiento de burbuja.**Compara cada elemento de la lista a ordenar con el siguiente e intercambia su posición si no están en el orden adecuado. Se revisa varias veces toda la lista hasta que no se necesiten más intercambios.
* **Ordenamiento por selección.**Vamos colocando el elemento más pequeño disponible en cada una de las posiciones de la lista de forma consecutiva.
* **Ordenamiento rápido.**Elegimos un elemento del conjunto (pivote) y reubicamos el resto a cada uno de sus lados, en función de si son mayores o menores que el elemento que estamos tomando como referencia. Repetimos el procedimiento en cada subconjunto.

Algoritmos voraces

Los **algoritmos voraces** consisten en una estrategia de búsqueda que sigue una heurística en la que se elige la **mejor opción óptima en cada paso local** con el objetivo de llegar a una solución general óptima. Es decir, en cada paso del proceso escogen el mejor elemento (elemento prometedor) y comprueban que pueda formar parte de una solución global factible. Normalmente se utilizan para resolver **problemas de optimización.**

En ocasiones, estos algoritmos no encuentran la solución global óptima, ya que al tomar una decisión solo tienen en cuenta la información de las decisiones que han tomado hasta el momento y no las futuras que puede adoptar. Algunos casos en los que los algoritmos voraces alcanzan soluciones óptimas son:

* **Problema de la mochila fraccional (KP).**Disponemos de una colección de objetos (cada uno de ellos con un valor y un peso asociados) y debemos determinar cuáles colocar en la mochila para lograr transportar el valor máximo sin superar el peso que puede soportar.
* **Algoritmo de Dijkstra**. Utilizado para determinar el camino más corto desde un vértice origen hasta los demás vértices de un grafo, que tiene pesos en cada arista.
* **Codificación Huffman**. Método de compresión de datos sin perder información, que analiza la frecuencia de aparición de caracteres de un mensaje y les asigna un código de longitud variable. Cuanto mayor sea la frecuencia le corresponderá un código más corto.

Programación dinámica

La**programación dinámica** es un método de resolución de problemas en el que dividimos un problema complejo en subproblemas y calculamos y almacenamos sus soluciones, para que no haga falta volver a calcularlas más adelante para llegar a la solución del problema. La programación dinámica reduce el tiempo de ejecución de un algoritmo al **optimizar la recursión.**

Eso sí, para poder aplicarse a un problema, éste debe tener **subestructuras óptimas** y **subproblemas superpuestos**. Es decir, que en él se puedan usar soluciones óptimas de subproblemas para encontrar la solución óptima del problema en su conjunto y que el problema se pueda dividir en subproblemas que se reutilizan para ofrecer el resultado global.

Algunos casos en los que se utiliza son:

* **La serie de Fibonacci.** Sucesión de números que comienza con “0” y “1” y, a partir de ellos, cada número es resultado de la suma de los dos que le preceden. La relación de recurrencia la define.
* **Problema de la mochila**.

Algoritmos probabilísticos

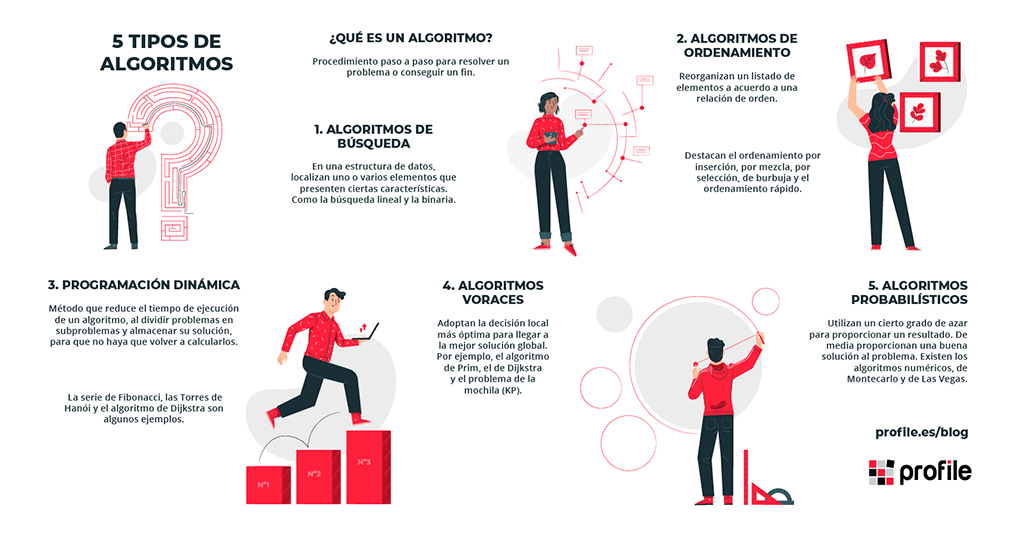
Es una técnica que usa una fuente de aleatoriedad como parte de su lógica. Mediante un muestreo aleatorio de la entrada llega a una solución que puede no ser totalmente óptima, pero que es adecuada para el problema planteado.

Se utiliza en situaciones con**limitaciones de tiempo o memoria** y cuando se puede aceptar una buena solución de media, ya que a partir de los mismos datos se pueden obtener soluciones diferentes y algunas erróneas. Para que sea más probable ofrecer una solución correcta, se repite el algoritmo varias veces con diferentes submuestras aleatorias y se comparan los resultados.

Existen dos tipos principales de algoritmos probabilísticos:

* **Algoritmo de Montecarlo.** Dependiendo de la entrada, hay una pequeña probabilidad de que no acierte o no llegue a una solución. Se puede reducir la probabilidad de error aumentando el tiempo de cálculo.
* **Algoritmo de Las Vegas.** Se ejecuta en un periodo de tiempo concreto. Si encuentra una solución en ese tiempo ésta será correcta, pero es posible que el tiempo se agote y no encuentre ninguna solución.

Infografía de los principales tipos de algoritmos

En esta **infografía,**puedes conocer de un vistazo estos **cinco** **tipos de algoritmos informáticos**:

*Ilustraciones de Freepik.*

Conclusión

Como hemos visto, un **algoritmo informático** no es más que un conjunto de instrucciones para conseguir un fin. Los **algoritmos**están muy presentes en el ámbito de la informática, pero también en nuestra vida cotidiana. Existen numerosos tipos y ejemplos de algoritmos y, dependiendo de la situación en que nos encontremos, unos u otros nos ayudarán a llegar a la solución que necesitemos.