# Algoritmos no computacionales

Sitio: Agencia de Aprendizaje a lo largo de la Vida
Curso: Técnicas de Programación - Turno mañana
Libro: Algoritmos no computacionales

Imprimido por: Tomas Friz
Día: viernes, 17 de septiembre de 2021, 12:44

# Tabla de contenidos

# 1. ¡Activá tus ideas!

## Hacé el siguiente ejercicio

Pensá en un tarea que realices cotidianamente, por ejemplo: ir a trabajar, ponerte a estudiar, ir a un encuentro con amigos/as, lavarte los dientes, cocinar una rica torta y muchas más.

Ahora, pregúntate: ¿Cuáles son los pasos que realizás para lograr esa tarea?

Te compartimos un ejemplo muy concreto para que puedas armar el tuyo propio.

1. Ir al baño
2. Prender la luz del baño
3. Tomar el cepillo de dientes
4. Tomar la pasta dentífrica
5. Colocar pasta en el cepillo
6. Dejar la pasta
7. Cepillar los dientes
8. Abrir la canilla
9. Enjuagar la boca
10. Lavar el cepillo
11. Dejar el cepillo
12. Mirarte al espejo para ver cómo quedaron tus dientes.

Estas acciones que otras que realizamos implican definir pasos, pequeñas decisiones

Entonces, ¿Qué es un algoritmo?

- Un algoritmo  es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, generalmente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades. (Wikipedia)
- Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. (Real Academia Española)
- Conjunto de pasos lógicos que tienen la finalidad de resolver un problema dado.

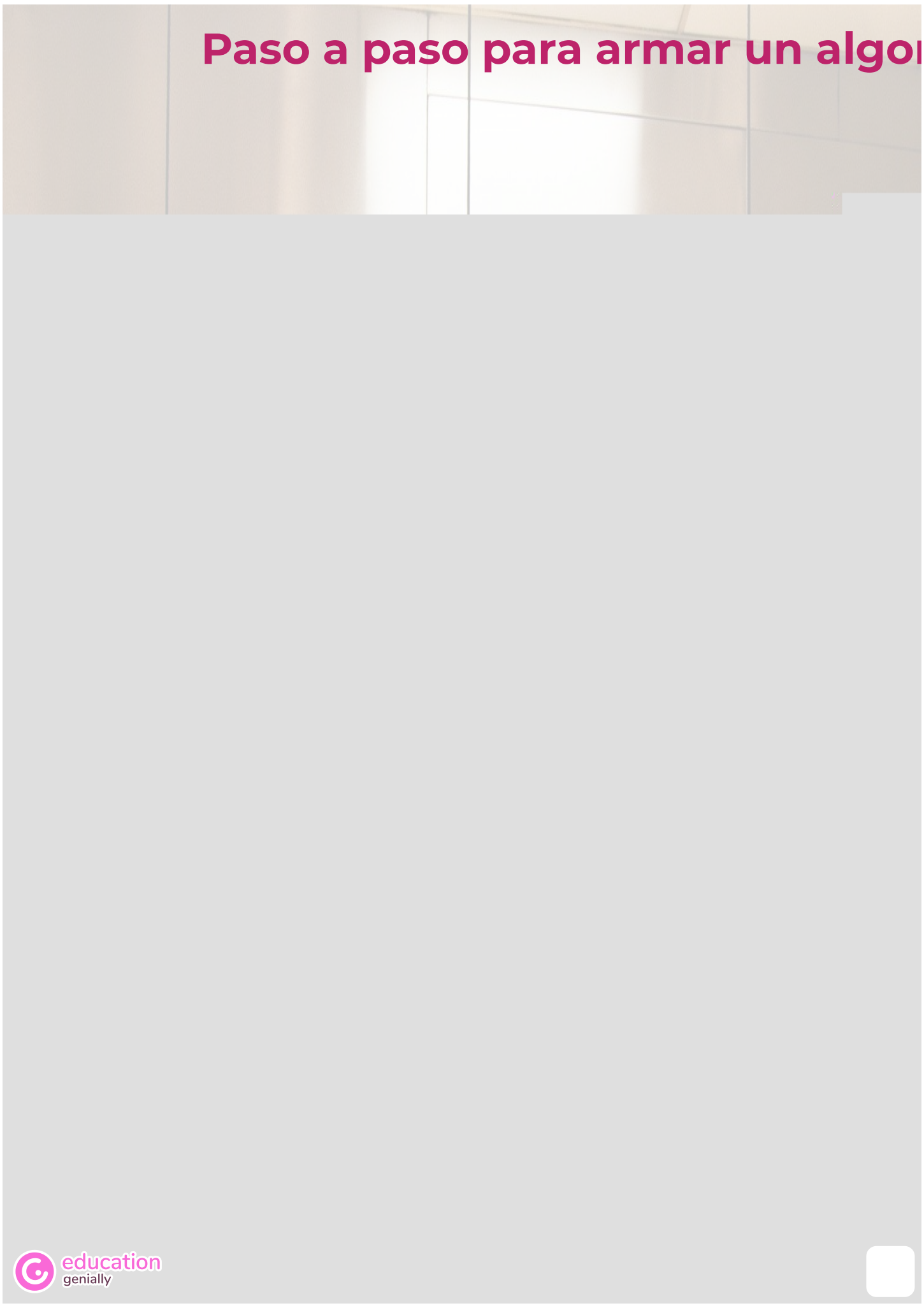
# 2. ¿Para qué sirven los algoritmos?

## ¿Por qué son importantes los Algoritmos? ¿Para qué sirven?

Crear algoritmo es el paso previo a la programación.

Antes de ponernos a programar en un lenguaje determinado, debemos encontrar la solución a un problema o situación que se nos plantea y cuáles son los pasos necesarios que debemos seguir para llegar a tal solución. Esto nos permite ahorrar tiempo de programación, evitar caer en prueba y error. Para esto utilizamos los algoritmos. Por otra parte, si bien hay distintas maneras en las que podemos modelar/graficar  nuestros algoritmos, muchas veces podemos hacerlo de una manera sencilla, utilizando un lenguaje coloquial y sin utilizar tecnicismos. Esto hace que sean una forma sencilla de desarrollar nuestras soluciones y al mismo tiempo de corroborarlas con otros participantes de la solución como colegas o clientes.

# 2.1. Identificar y describir



# 2.2. Luego de describir, verificar

### Una vez que terminamos de describir los pasos necesarios para resolver nuestro problema. ¿Qué deberíamos hacer?

Si analizamos la definición de algoritmo vamos a encontrar una palabra que es muy importante y que está relacionada con el orden de los pasos detallados.

- Verificar que cada uno de los pasos de nuestro algoritmo estén ordenados correctamente es imprescindible para que nuestro algoritmo nos de el resultado esperado.
- Si tomamos los pasos detallados anteriormente y los cambiamos de orden, fácilmente nos daríamos cuenta que dependiendo de los pasos que se cambien llegaríamos al mismo resultado o no.

### ¿Qué más tendríamos que tener en cuenta al momento de formular nuestros algoritmos?

Una vez que verificamos que los pasos que forman nuestro algoritmo deberíamos verificar que dichos pasos sean precisos. Esto quiere decir que cada uno de los pasos que componen un algoritmo debe ser una tarea única y no debe dar lugar a ambigüedades.

# 3. Representación de algoritmos

Los algoritmos los podemos escribir en lenguaje natural, pero esto los vuelve un tanto imprecisos.

Es cierto que cuando trabajamos con algoritmos de la vida cotidiana en general es suficiente, pero, la forma precisa y sin ambigüedades es representarlos mediante dos maneras formales: como diagrama de flujo o como pseudocódigo .

Un diagrama de flujo es una representación gráfica de un algoritmo. Son bastante útiles al principio, dado que son fáciles de entender, gracias a que presentan las instrucciones de una manera gráfica. En el diagrama de flujo se emplean distintas figuras para representar las diferentes acciones.

Un pseudocódigo es una descripción de las instrucciones de manera tal de ser muy similar al formato que se obtiene al utilizar un lenguaje de programación. El punto es que el pseudocódigo no tiene un estándar de reglas sintácticas a seguir, sino que es constituido por convención por uno o más programadores para tener una solución abstracta del problema, algo así como una base para luego transcribir ese algoritmo en un lenguaje de programación real.

# 3.1. Ejemplo de representación

## Formas de representación de un a