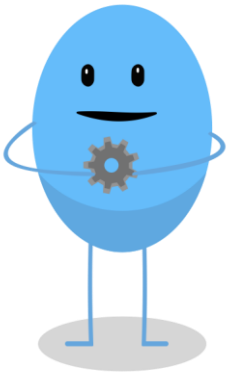
# Introducción a la Programación

Microsoft Student Partners

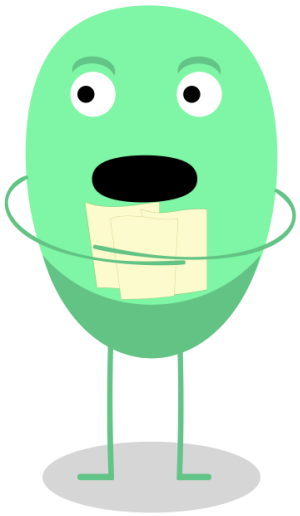
Microsoft

when technology becomes absolute passion



# Índice





## ¿Qué es programar?

Programar consiste en escribir un conjunto de **órdenes** o **instrucciones** utilizando un lenguaje capaz de ser comprendido por un ordenador.

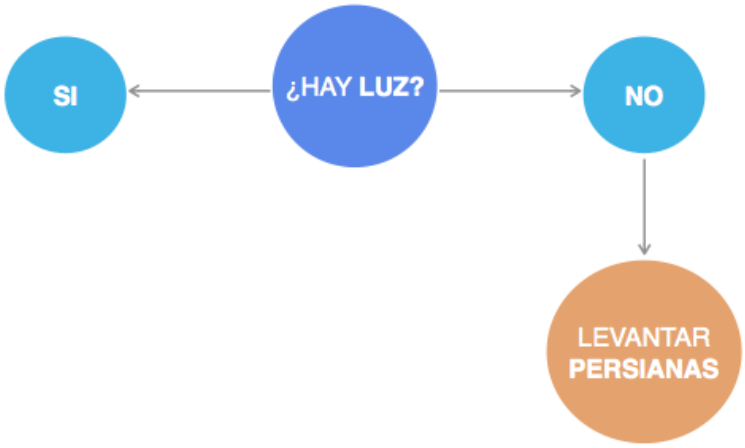
Mientras que nuestro lenguaje está compuesto por palabras, letras, signos de puntuación…etc., las "palabras" que entienden los ordenadores están formadas únicamente por **ceros** y **unos**.

Por tanto, las órdenes de programación tienen una sintaxis mucho más sencilla que nuestro lenguaje.

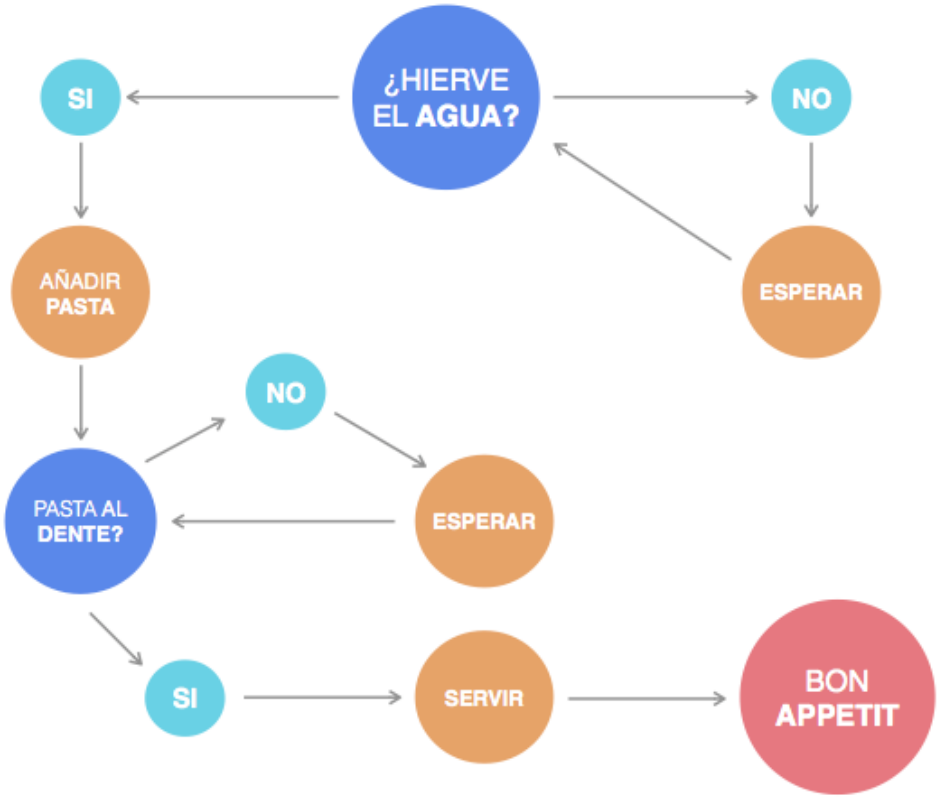
# Introducción a los algoritmos

Los algoritmos son la descripción ordenada de un conjunto de instrucciones que permiten realizar una determinada actividad paso por paso.

La forma más sencilla de representar un algoritmo es mediante un **diagrama de flujo:**



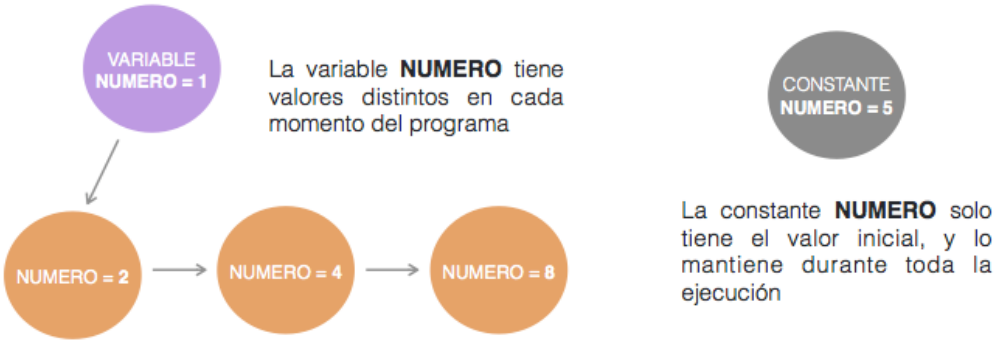
Por ejemplo, un algoritmo para hacer pasta sería:



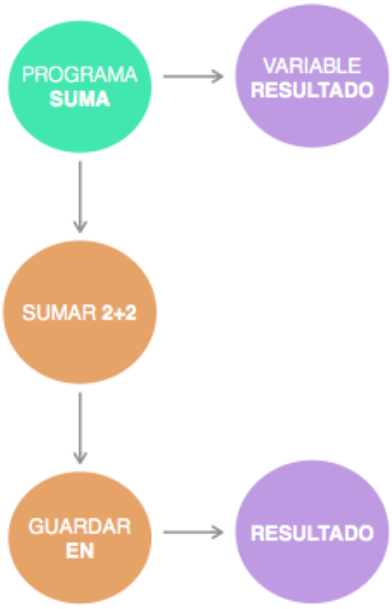
## Variables y constantes

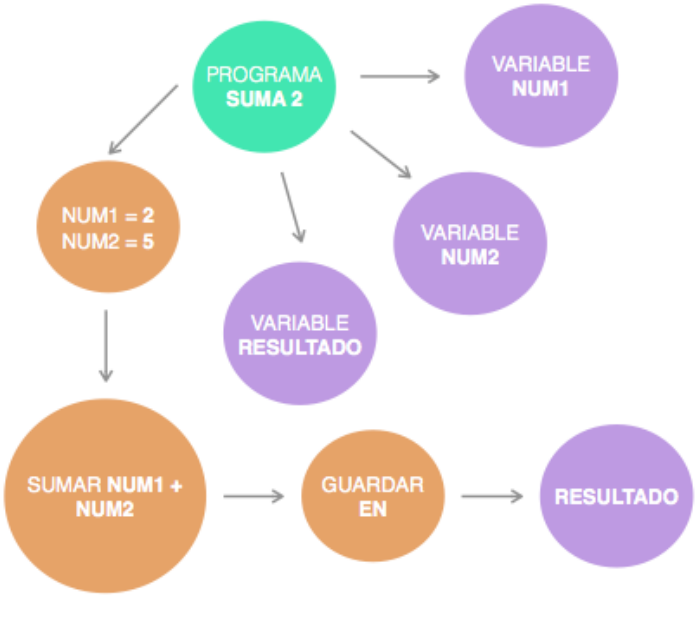
Además de las instrucciones, a menudo necesitamos almacenar datos o información relativa a dichas instrucciones. Para ello podemos usar:

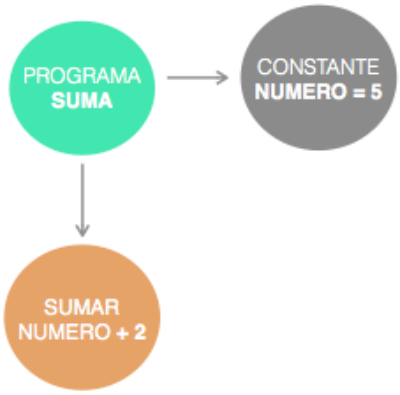
- Las **variables**: pueden cambiar de valor.
- Las **constantes:** no cambian de valor a lo largo del algoritmo.

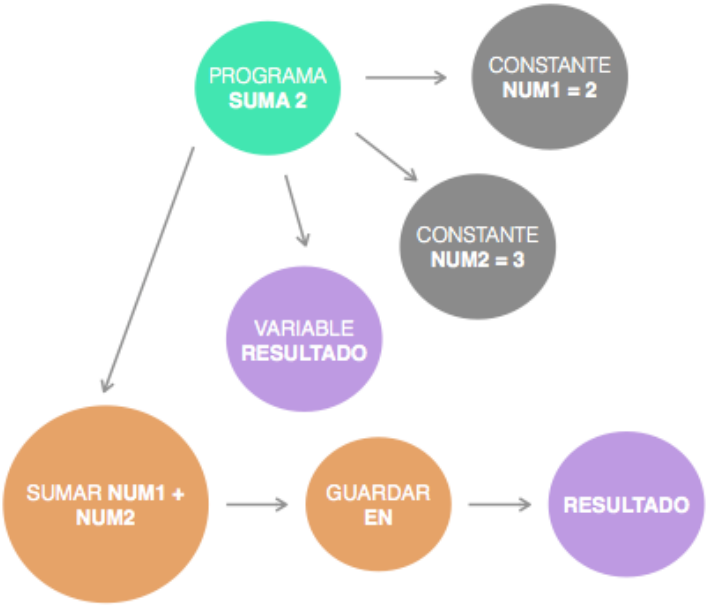


Algunos ejemplos de uso variables y constantes:



## Instrucciones condicionales

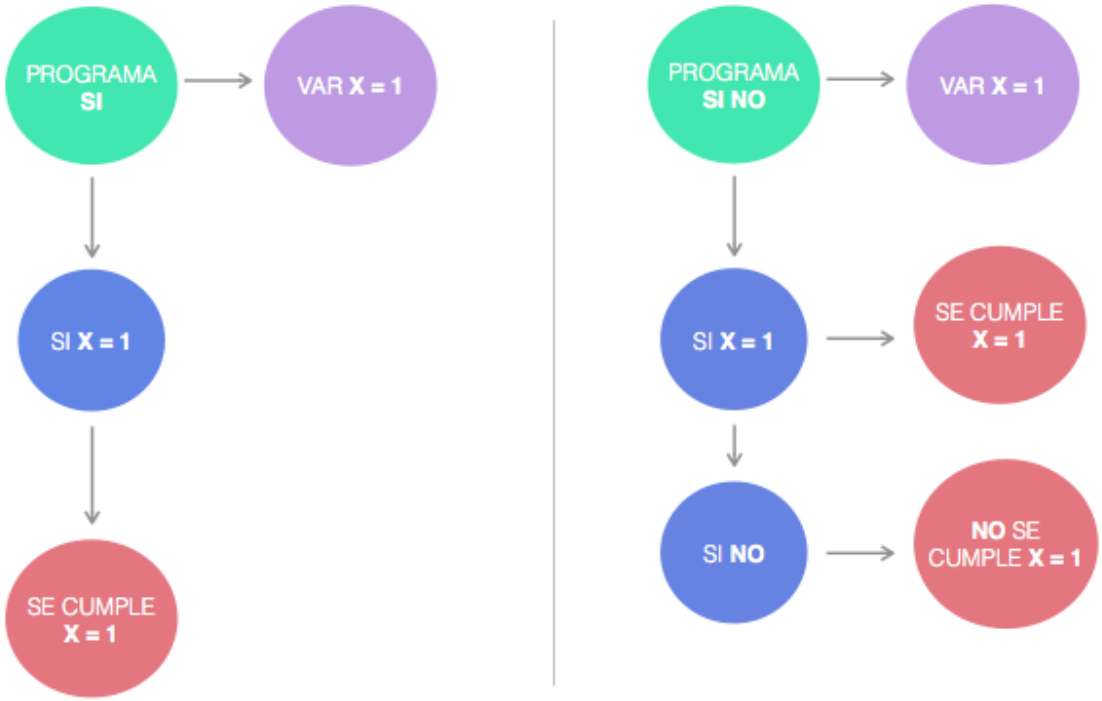
Si solo tuviéramos instrucciones simples tipo “suma esta variable con esta otra” o “espera 5 segundos” los programas serían bastante aburridos, ¿verdad?

Sin embargo, existen instrucciones más ricas que nos permiten realizar muchas más acciones.

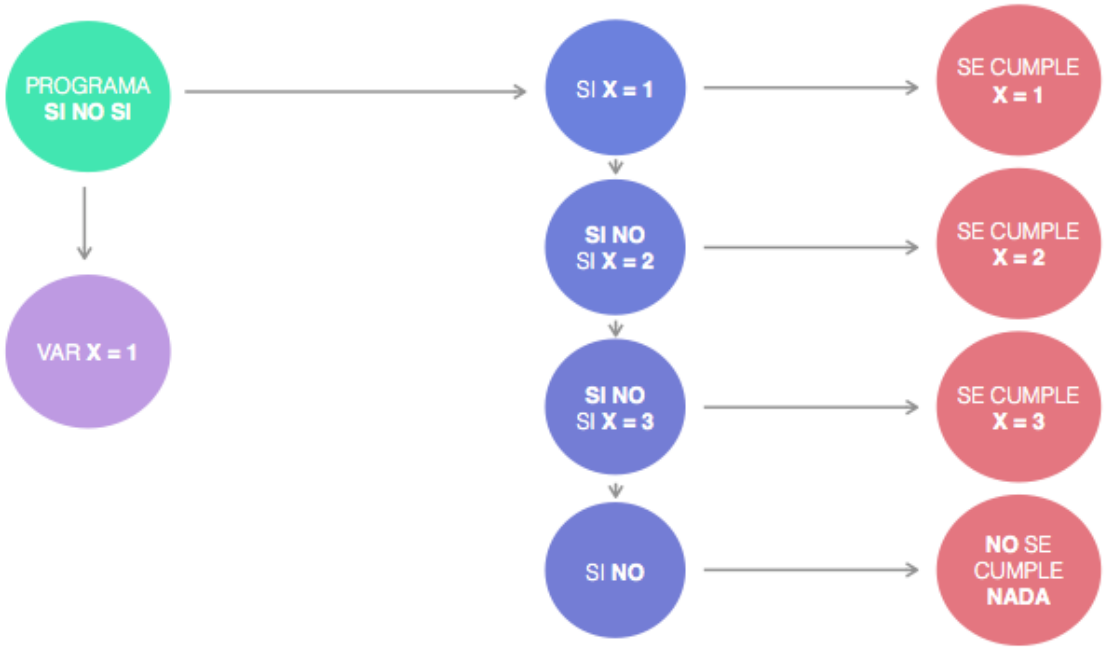
En el caso de las instrucciones condicionales, podremos indicar si se va a realizar una instrucción o no, dependiendo del cumplimiento de un requisito o condición previo.

Así tendremos instrucciones complejas o estructuras de tipo: **si, si no, si no si, en caso de.**

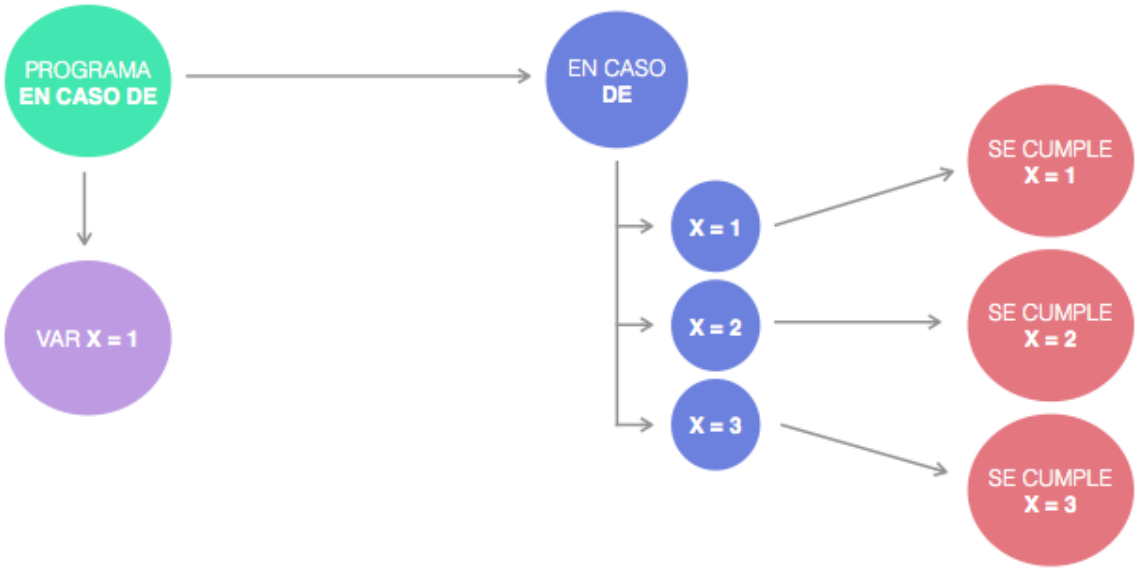
Ejemplos de programas **si** y **si no**:



Ejemplo de **si no si:**




Ejemplo de **en caso de:**

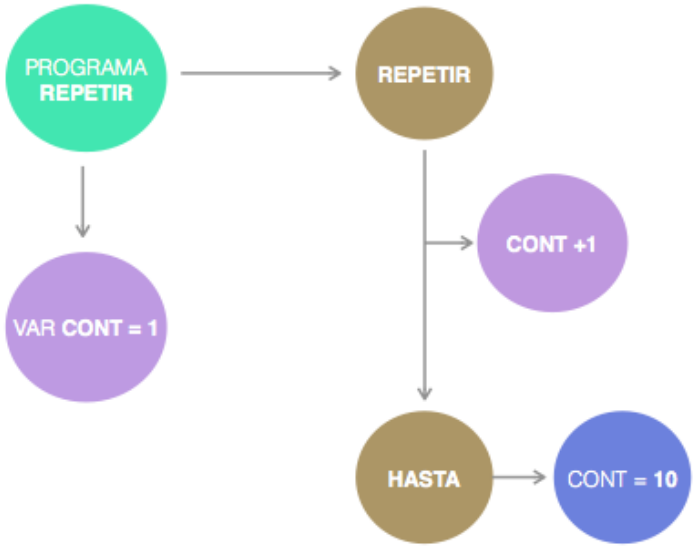


## Estructuras de control

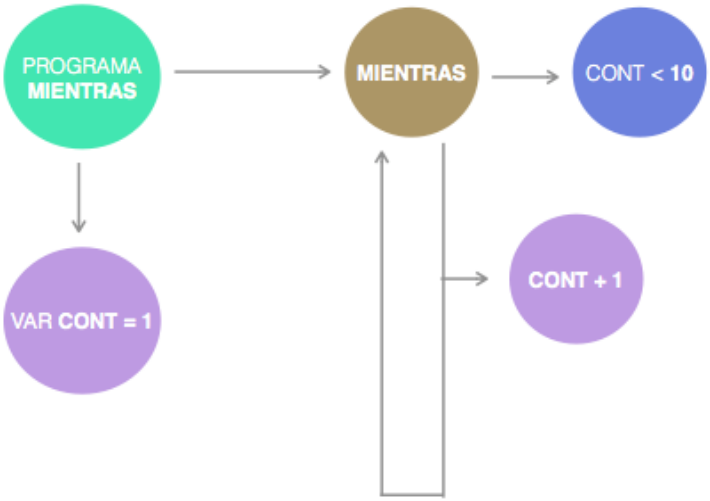
Hasta ahora hemos visto instrucciones que solo se repiten una vez, pero muchas veces tenemos que volver a ejecutar una parte del código varias veces. Para ello contamos con las **estructuras de control**: **repetir – hasta, mientras** y **para.**

Ejemplo de **repetir**:

Se ejecutará lo que haya entre **repetir - hasta** si se cumple la condición. En este caso añadimos un contador, si no el programa seguiría funcionando eternamente.

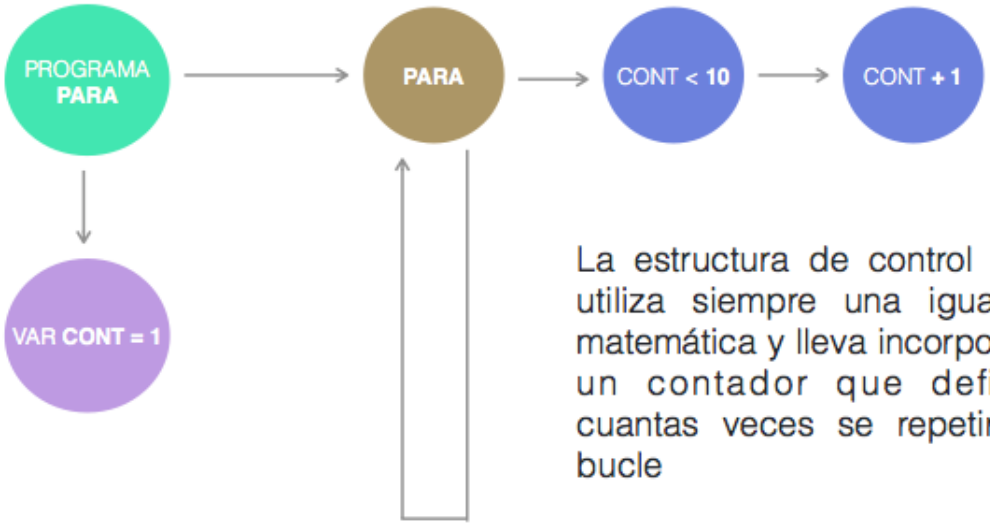


Ejemplo de **mientras**:




La estructura de control **mientras** analiza la condición antes de ejecutar lo que haya a continuación.

Ejemplo de **para**:




La estructura de control **para** utiliza siempre una igualdad matemática y lleva incorporada un contador que definirá cuantas veces se repetirá el bucle

## Subprogramación

Conforme vamos creando nuestro programa, el número de instrucciones crece de tal manera que dicho programa comienza a ser demasiado largo e incomprensible.

¿Cómo solucionamos este problema?

Muy sencillo: creando diferentes mini-programas dentro del programa completo e ir llamándolos cuando los necesitemos.

Estos mini-programas se denominan **funciones**, y a esta técnica de división del trabajo, **subprogramación**.

Ejemplo de un **programa principal** y su **función** sumar: