



INTRODUCCIÓN

La programación es una disciplina (de tantas hoy en día) que pertenece a la informática, ciencia que se dedica al análisis y transformación de datos e información a partir del uso de computadoras.

En dicha disciplina se desarrollan algoritmos en un determinado lenguaje de programación, los cuales serán implementados en una computadora para de esa manera realizar los llamados programas.

Algoritmo

Un **algoritmo** es una secuencia finita y ordenada de **instrucciones** que deben ser seguidas en pos de resolver un problema.

- [\[Video\] ¿Qué es un algoritmo?](#)
- [\[Video\] ¿Qué es una instrucción?](#)

Programa

Un **programa** es, técnicamente hablando, uno o más (al menos uno) **algoritmos** de programación escrito en un lenguaje que puede ser interpretado y ejecutado por una computadora. En otras palabras, decimos que un **programa** es la solución a un problema, ya que cada vez que se requiere desde sumar números hasta realizar traducciones, grandes cálculos o enviar mensajes de texto, se recurre a un programa (o aplicación, o web, lo que corresponda).

- [\[Video\] ¿Qué es un programa?](#)

Existen distintos tipos de programas o aplicaciones (podemos decir que son sinónimos), por ejemplo:

- **Programas de escritorio:** Se ejecutan en windows, por ejemplo Word, Excel, el buscaminas.
- **Programas web:** Facebook, Mercado Libre, son aplicaciones o sistemas web.
- **Programas de celular:** también llamados apps, o aplicaciones, por ejemplo Whatsapp, Instagram, Google Maps.

Todos estos son parte de un gran grupo dentro de la informática que se denomina **Software**.

Sistema Informático

El sistema informático está compuesto necesariamente por tres partes:

- Hardware
- Software
- Humano



Ante la ausencia de alguna de éstas partes, no podríamos completar el sistema y este no tendría sentido dado que:

Software es todo lo que creamos para que una computadora funcione, pero requiere justamente una computadora donde hacerlo, ahí entra el Hardware.

Hardware es “todo lo que se puede tocar” en una computadora: el monitor, el teclado, incluso todos los componentes internos dentro de la misma.

Sin embargo, sin programas para utilizar, todo ese poder de procesamiento, no nos sería de utilidad.

Humano. Se completa el ciclo con el humano que es quien hace uso de ese Software gracias a ese Hardware, y quizá pueda sonar a una obviedad, pero es clave tener en cuenta este círculo.

Como ya dijimos, el Software es lo que le da vida a la computadora. Son los programas que nos permiten utilizarlas, y existen distintas categorizaciones dentro del software. Las más comunes son:

- **Software de base:** esto corresponde a las cuestiones básicas y de sistema operativo que se requiere para poder encender y usar una computadora (Windows, Linux, Mac OS, Android, iOS).
- **Software utilitarios:** son aplicaciones complementarias para el correcto funcionamiento y/o administración de los sistemas. Podemos hablar del desfragmentador de disco, o utilidades de análisis de errores en memoria, entre otros.
- **Software de aplicación:** los programas, sistemas o aplicaciones que utilizamos a diario.

Existen más categorías y sub-categorías y además ciertas características adicionales como si son libres, o no, o si son open source, o multiplataforma, pero eso es para indagar en otro contexto.

También podemos mencionar los “**programas para programar**”. Desde ya que hoy en día para poder desarrollar programas, vamos a necesitar una computadora, pero además vamos a necesitar uno o más (según corresponda) programas que nos sirvan para ello. Hay aplicaciones específicamente preparadas para, justamente, desarrollar otras aplicaciones, en definitiva, para programar. Estas aplicaciones se suelen conocer como **IDE**, de sus siglas en inglés “**Entorno de Desarrollo Integrado**”.

Dependiendo de lo que quisiéramos programar, será la aplicación que deberíamos utilizar, y esto aplica del mismo modo para la tecnología o **lenguaje de programación** a utilizar.



- [\[Video\] ¿Qué es programar?](#)
- [\[Video\] ¿Qué es un lenguaje de programación?](#)

Tipos de datos

Como veremos más adelante, al programar no solo hay que escribir en un lenguaje determinado, sino que vamos a tener que utilizar muchas herramientas distintas, dentro de la propia programación, que iremos adquiriendo a medida que vayamos avanzando. Una de esas herramientas y de la que más vamos a hablar y utilizar de entrada (y para siempre si seguís con la programación) es la variable.

Una variable, técnicamente hablando, un espacio en la memoria de la computadora destinado a ser utilizado por nuestro programa para poder guardar un dato. La variable tiene ciertas características como que debe contar con un nombre, que puede guardar un solo dato a la vez (el mismo puede ir cambiando, de allí su nombre) y que debe tener un tipo. Este tipo se refiere a qué es lo que precisamente va a guardar esa variable. Los primeros tipos que aprendemos son los más básicos y elementales pero que nos acompañarán, junto a las variables, en toda la vida dentro de la programación. Estos tipos son:

- **Int** para guardar números enteros.
- **Float** o **double** (dependiendo del lenguaje, hay otras variantes) para guardar números con coma.
- **Char** para guardar letras.
- **Bool** para guardar verdadero o falso (que es lo mismo a 1 o 0 respectivamente).

Que le asignemos un tipo de dato a una variable afecta y tiene una relevancia puntual. Si estoy creando una variable de tipo entera, la misma solamente podrá guardar valores enteros. Qué sucederá si quiero guardarle un número con coma? Bueno, dependiendo del momento, el lenguaje o la operación, podrá dar error o bien podrá redondear el valor quitando los decimales y quedándose solo con la parte entera. Ninguno de los escenarios serían pertinentes en nuestro programa. Lo ideal sería que si la variable solo admite valores enteros, pues que se la use con valores enteros.

- [\[Video\] ¿Qué es una variable?](#)
- [\[Video\] ¿Qué es un tipo de dato?](#)



En este módulo comenzamos a manejar conceptos más precisos y técnicos, puntuales y necesarios para la confección de nuestros primeros programas.

En la unidad anterior comprendimos las herramientas a utilizar y llevamos a cabo la creación del primer proyecto para corroborar que todo estaba en orden.

Lo primero que tenemos que comenzar a comprender en cuanto a los programas es que existe un principio básico de todo algoritmo, de todo programa, de toda aplicación; el cual se basa en el esquema que se muestra a continuación:



Todo programa, para poder operar, necesita contar con una entrada de datos con los cuales realizará un posterior procesamiento, una transformación de esos datos, a partir de los cuales terminará obteniendo información para finalmente brindar una salida.

Nuestro ejemplo más sencillo parte de la necesidad de sumar dos números. Para poder sumar dos números, el programa deberá solicitarlos (o recibirlos), es decir, contar con una entrada de datos que serán esos dos números. Luego aplicará un procesamiento, que en este caso será sumar esos dos valores que fueron ingresados; para finalmente obtener el resultado de la suma y mostrarlo en pantalla. Esta última sería la salida de información.

¿Dato o Información?

Por qué se hace hincapié en la diferencia de que lo que entra son Datos y lo que sale es Información? Podríamos tomarlos como sinónimos?

La definición de datos dice que: un dato es un conjunto de símbolos sin significado alguno, mientras que por otro lado, la definición de información afirma que: la información está compuesta por símbolos a los que se le ha atribuido algún significado.

Yo siempre suelo dar un ejemplo un tanto fantástico, pero simple y que, a mí entender, funciona para comprender la idea.

Se trata de una persona que, caminando por la calle, se encontró con un pergamino tirado. Cuando lo abre se trata de un documento bastante elegante, con firmas y



demás pero con un texto que esta persona no podía comprender. Pensó en hacer un cuadro, ya que era lindo, vistoso; terminó por regalárselo a un amigo hijo de japoneses. Este amigo resulta que manejaba la lengua nativa de Japón por sus padres y al ver el pergamino entendió de qué se trataba y no solo eso, pudo leerlo pues estaba escrito en un lenguaje que manejaba y comprendía. Se trataba de un documento correspondiente a un testamento el cual otorgaba una fortuna a quién lo encuentre y, desde ya, pueda reclamarlo por saber de qué se trataba.

A qué quiero llegar con todo esto? Para la primera persona ese pergamino no era más que un papel "lindo" que a lo sumo podría utilizar para decorar una sala. Un papel con un conjunto de símbolos sin sentido. Pero por otro lado, para su amigo, quien contaba con los elementos necesarios para comprender ese papel, lo que decía en él, par aplicar un procesamiento (a partir de traducir -aplicar un algoritmo-) el contenido del mismo, pasó a atribuirle un significado valiosísimo, en este caso monetario, transformando esos datos en información, mediante un procesamiento.

El fin de toda aplicación es siempre obtener información.

El proceso y las variables

Para poder realizar las transformaciones de las que hablamos necesitamos construir nuestros algoritmos con las instrucciones necesarias. Desde ya que tendremos, generalmente, para comenzar un ingreso de datos y para terminar una salida de información. Pero en el medio es donde vamos a tener que establecer las instrucciones necesarias para darle forma al proceso pertinente que realice la transformación, o "la magia".

Volviendo a nuestro ejemplo de la suma, el proceso sería justamente sumar, valga la redundancia, pero hay varias cosas que se encuentran implícitas en esa operación.

Para poder sumar dos valores necesitamos previamente contar con ellos, y para eso no alcanza con pedirlos. Una vez que los pedimos, tenemos que guardarlos. Para poder guardar esos datos vamos a necesitar hacer uso de un elemento fundamental en la programación: la variable. Una **variable** es un espacio en memoria que permite alojar un valor. Es un medio que utilizaremos en nuestros programas para ir guardando todas las cosas que necesitamos dentro y a lo largo de todo **algoritmo**.

Imaginen la memoria **RAM** (si no conocen la memoria RAM repasen el video de la PC del módulo 1) como una hoja excel, la cual cuenta con columnas representadas por letras y filas representadas por números y en cada intersección encontramos una celda.



	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Ok. Si yo digo **C5**, estoy haciendo referencia a la **columna C** en la **fila 5**. Podemos decir que la memoria RAM trabaja parecido a esto. C5 sería la **dirección de memoria** de esa celda. Sin embargo, al querer usar esas "**celdas**" (**variables**) en nuestros programas no vamos a estar todo el tiempo usando direcciones de memoria, para ello, lo que haremos es ponerle nombre a las variables. De ese modo, podremos hacer referencia a esas celdas mediante un nombre significativo y fácil de recordar. Significativo porque deberíamos nombrar nuestras variables teniendo en cuenta qué es lo que contienen dentro.

Volviendo una vez más a nuestro ejemplo de la suma, para guardar los dos valores ingresados podemos decir que lo haremos en las variables **numero1** y **numero2**. Los nombres de las variables pueden contener cualquier letra, varias palabras y también números, aunque hay ciertas restricciones:

- No se puede comenzar el nombre de una variable con números.
- El nombre no puede estar compuesto solo por números.
- El nombre no puede contener símbolos especiales (por ejemplo * o +).

El nombre no puede ser igual a una palabra reservada. Dependiendo del lenguaje de programación que usemos, hay palabras que son específicas para hacer algo puntual, con lo cual no podemos usarlas como nombres de variables. Por ejemplo:

- [\[Video\] ¿Qué es la memoria RAM?](#)
- [\[Video\] ¿Qué es una VARIABLE?](#)

Operadores Aritméticos

Son los primeros operadores que comenzaremos a utilizar. Estos son:



- + para sumar
- - para restar
- * para multiplicar
- / para dividir

Estas son las operaciones básicas que el procesador ya conoce y que tenemos disponibles para utilizar. De allí para arriba, es decir, un cuadrado, un logaritmo, etc., etc., son operaciones que tendremos que construir a mano porque el procesador no las sabe. Es verdad que en la actualidad existen muchas cosas ya hechas las cuales podemos utilizar (se llaman librerías, las veremos más adelante), pero generalmente cuando se está comenzando en la programación, una de las primeras cosas se trata de recrear algunas de esas funcionalidades para entender cómo trabajan.

- [\[Video\] ¿Qué es el CPU?](#)

Asignación

La asignación es una acción mediante la cual se da un valor a una variable. Por ejemplo

```
jota = 10;
```

Lo que estamos haciendo en esa instrucción es tomar el valor 10 y asignarlo, o sea guardarlo, en la variable llamada jota. Hay varios orígenes de datos para realizar una asignación. La que acabamos de ver directamente asigna un valor. También se puede asignar el resultado de una operación matemática o simplemente asignar lo que se pide que una persona ingrese por teclado:

```
resultado = jota + 7;
```