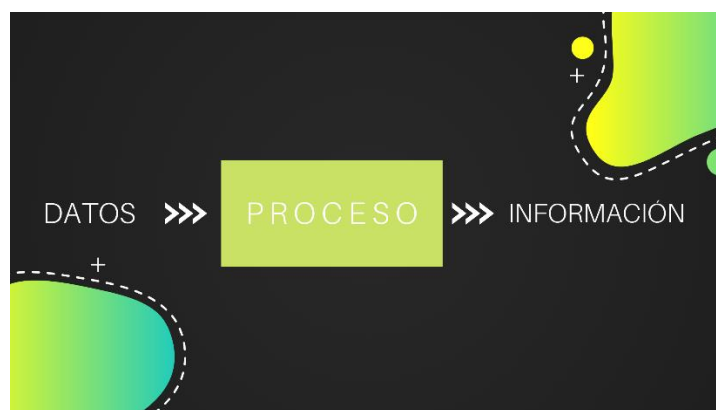


## Primer Programa

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:



## Algoritmo

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 mi 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, para 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:

**cout:** Palabra que encontraremos al realizar un nuevo proyecto en **Codeblocks**, en ese primer Hello World:

```
cout << "Hola Mundo";
```

[\[Video\] ¿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;  
cin >> variable;
```

## El primer programa

Ya sabemos que tenemos que pedir y guardar, y que tenemos que procesar y mostrar. Y tenemos casi todas las herramientas. Hagamos nuestro primer programa en código y luego terminamos de analizar.

```
int numero1;  
int numero2;  
int resultado;  
cin >> numero1;  
cin >> numero2;  
resultado = numero1 + numero2;  
cout << resultado;
```

En las primeras tres líneas lo que hacemos es declarar las variables y darles un tipo. Trabajaremos con números enteros (recordemos que los tipos que manejaremos por ahora son int, float, char y bool).

Luego pedimos datos por teclado con la función **cin**. Para que funcione correctamente necesita el operador ">>" y luego va la variable a la que se le va a asignar dicho valor. Esto corresponde a lo que ya conocimos como "**caja de ingreso**".

Luego de pedir y guardar los datos en nuestras variables, volvemos a usar éstas últimas ahora para leer el contenido que antes guardamos y, haciendo uso del operador **aritmético para sumar**, realizamos la suma. El resultado lo guardamos con el **operador =** (que en programación no es igual, sino **asignar**).

Finalmente, con **cout** (seguido por "<<") volvemos a leer, esta vez el valor que quedó guardado en **resultado**, para mostrarlo en pantalla. Recordá que acá lo estamos viendo a modo de ejemplo, el **cout** es la "**caja de egreso**".

Y listo, ese es nuestro primer programa. Todos estos pasos que tenemos detallados aquí, los tenés súper explicados en el video "**Primer Programa**" que te sugiero que veas.

Luego de todo esto, ya estamos en condiciones para seguir avanzando en la creación de otros programas de este tipo.

Tenés algunos ejercicios para practicar.

## Otras cosas interesantes que deberías saber

[\[Video\] ¿Qué es un Lenguaje de Programación?](#)

[\[Video\] ¿Qué es COMPILAR?](#)

[\[Video\] ¿Qué es SITÁXIS?](#)