

# INSTITUTO SUPERIOR DE FORMACIÓN TÉCNICA N° 177

---

---

## **Curso Introdutorio**

---

---

- **Conceptos básicos**
- **Introducción a la Programación**

**Prof. Lic. Walter Carnero**  
wdcarnero@gmail.com

## INTRODUCCIÓN

Antes de Comenzar a redactar este pequeño apunte, consideramos conveniente realizar algunas aclaraciones con respecto al título que obtendrán, y junto con él a las capacidades que desarrollarán en esta institución relacionadas a la carrera de Técnico Superior en Análisis de Sistemas. En primera instancia citaremos cuáles son las competencias que un Analista de Sistemas debe tener, las mismas se ven reflejadas en el perfil profesional de la carrera:

- Analizar el flujo de información dentro de una organización o empresa, de manera de poder optimizar el mismo, determinando cuales son los procesos factibles de ser sistematizados.
- Comprender y analizar los requerimientos planteados por el usuario respecto a problemas que involucren sistemas de información.
- Optimizar el comportamiento de aplicaciones y sistemas, incluyendo aquellos que operan en plataformas cliente/servidor.
- Diseñar, desarrollar e implementar programas para dar solución a problemáticas específicas.
- Demostrar y realizar asistencia al usuario sobre la operatoria de equipos, sistemas de redes y programas.
- Diagnosticar fallas y problemas que se le presentan al usuario durante la operatoria habitual, evaluando alternativas de solución.
- Planificar, con herramientas adecuadas, el tiempo de las actividades.
- Programar los componentes de la solución.
- Anticipar problemas derivados de los cambios de tecnología.
- Reconocer y definir componentes y equipamientos informáticos funcionando como estaciones solitarias o formando parte de una red.
- Probar la solución alcanzada, integrándola en el entorno previsto para su funcionamiento.

Teniendo en claro el perfil profesional del Analista de Sistemas, podemos destacar la importancia de los sistemas informáticos en la tarea cotidiana del analista y junto a ella la importancia que tiene la computadora como instrumento de trabajo del mismo.

Es por ello que se le debe prestar especial importancia al conocimiento, la instalación y manipulación, la programación, y el mantenimiento de la computadora, ya que será, de aquí en más, una de nuestras más importantes herramientas.

Podemos citar, en base al conocimiento del programa general de la carrera (Tema a tratarse dentro del espacio institucional), que de su estructura curricular se desprenden tres espacios curriculares (materias) específicos, los mismos son:

- Introducción a los Sistemas de Información.

- Arquitectura de computadores.
- Programación.

Estos tres espacios, cada uno con su respectiva carga horaria, serán los pilares en el primer año de la carrera, ya que a través de ellos se podrán desarrollar las capacidades que lleven a obtener las competencias citadas anteriormente. Es por ello que se les debe prestar mucha atención.

Este pequeño apunte servirá de introducción al espacio curricular Programación 1 y es importante que se entienda todo su contenido, como consejo, realizar una primera lectura y tomar nota de los puntos que no queden claros o no se comprendan para luego aclararlos en las clases presenciales. El curso introductorio que tratará los temas incluidos en este pequeño apunte consta de 4 horas, sobre las cuales se trabajarán cada uno de los puntos que aquí se desarrollan.

## REFERENTE A PROGRAMACIÓN 1

Citamos anteriormente que la computadora será una de las más importantes herramientas que utilizaremos en nuestra tarea como analistas, por lo cuál trataremos de responder en primera instancia el interrogante ¿Qué es una computadora?

Una computadora es un sistema electrónico, capaz de almacenar, procesar y entregar resultados en forma extremadamente rápida. Esto lo hace mediante la realización de cálculos y toma de decisiones lógicas a velocidades de millones, incluso miles de millones de veces más rápido que el ser humano. Pero una máquina al fin, que sin un conjunto ordenado de instrucciones es imposible que realice alguna tarea. Ahora bien, ¿De qué manera se le pueden dar órdenes a una computadora?, la respuesta es muy simple, a través de un programa.

Las computadoras procesan los datos bajo el control de un conjunto de instrucciones que indican paso a paso las acciones que se deben llevar a cabo. Un programa justamente es un conjunto ordenado de acciones que le indican a la computadora cuales son los pasos que se deben seguir para realizar la concreción de una determinada tarea. Un programa define los pasos a seguir mediante la aplicación de algún algoritmo, siendo éste un vocablo de origen árabe (al-jwarizmi)<sup>1</sup> que define en un conjunto finito de instrucciones los pasos a seguir para resolver un problema determinado. La diferencia existente entre programa y algoritmo es que un programa indica a la computadora los pasos a seguir mediante la utilización del algoritmo, el programa debe estar escrito (ingresado a la computadora) mediante un lenguaje de programación determinado. Un lenguaje de programación nos da la manera en que debemos escribir las instrucciones del algoritmo, a esto se lo denomina *sintaxis del lenguaje* y cada lenguaje de programación utiliza una sintaxis distinta.

Existen en la actualidad diversos lenguajes de programación, los cuales pueden diferenciarse en base al acercamiento que pueda tener el programador con el hardware<sup>2</sup> de la computadora. Podemos definir estos niveles en Bajo, Medio y alto, así de esta manera tendremos lenguajes de programación de bajo nivel, de nivel medio y de alto nivel siendo la programación de los mismos más compleja en cuanto bajemos de nivel, por supuesto

<sup>1</sup> Abu Abdallah Muhammad ibn Mūsā al-Jwārizmī, matemático, astrónomo y geógrafo musulmán, (1780-1850).

<sup>2</sup> En informática, todos los componentes físicos que conforman una computadora o un sistema de red.

esto tiene algún sentido determinado y cada lenguaje de programación esta preparado para aplicarse a la resolución de distintas tareas. Otra clasificación quizá más minuciosa esta dada por diferenciar a los lenguajes de programación en: Lenguaje de máquina, Lenguajes ensambladores y Lenguajes de alto nivel. Una computadora puede entender únicamente su lenguaje de máquina, es decir que es este el lenguaje natural de esa computadora, el mismo se define por el diseño del hardware de la misma. Por lo tanto el lenguaje de máquina estará definido por la arquitectura de cada computadora, aunque es importante comentar que las computadoras modernas se basan (aunque no todas) en la misma arquitectura siguiendo estándares de fabricación.

Los lenguajes de máquina son difíciles de entender para los seres humanos y su programación es lenta y tediosa, para resolver este problema se desarrollaron programas traductores llamados ensambladores, los cuales se utilizan para convertir programas escritos en lenguaje ensamblador a lenguaje de máquina aunque dicho lenguaje es más comprensible para los seres humanos es incomprensible para la computadora hasta que el ensamblador lo convierta al lenguaje de máquina.

Podemos deducir hasta aquí que para hacer más simple la programación se crearon los lenguajes ensambladores, pero éstos siguen siendo largos y tediosos de programar, por este motivo es que se desarrollaron los lenguajes de alto nivel donde simples instrucciones individuales realizan importantes tareas, estos programas de alto nivel también requieren ser traducidos a lenguaje de máquina para que la misma pueda comprenderlos, el encargado de dicha traducción se denomina compilador. Existen de cada lenguaje de programación varias versiones del compilador, ya que el mismo también es un programa y por lo tanto requiere continuas revisiones y mejoras, dichas mejoras se logran mediante la actualización de la versión del mismo.

Obviamente es más fácil para el programador escribir los programas en lenguajes de alto nivel ya que los mismos le permiten trabajar con instrucciones (generalmente en inglés) de uso común. Los lenguajes de programación de alto nivel se desarrollan teniendo en cuenta su finalidad, es decir, que aplicaciones o problemáticas podrá resolver el programador con un lenguaje determinado, a continuación se detallan algunos de los lenguajes de programación más populares y cual es su finalidad:

Lenguajes de Programación	Programación Científica	Orientados a objetos	Manipulación de Datos	Plataformas Visuales
	ENSAMBLADOR	C++	COBOL	VISUAL BASIC
	FORTRAN	JAVA	CLIPPER	DELPHI
	C	C#	DBASE	VISUAL C++
	PASCAL	PERL	PHP	VISUAL FOX
	ALGOL	PYTHON	XML	GAMBAS
		ADA		

El lenguaje de programación con el cual trabajaremos durante el año es el C, siendo este uno de los más difundidos en el mundo entero, su creador fue Dennis Ritchie<sup>1</sup>, el cual, para su creación, se baso en el lenguaje B anteriormente desarrollado por Ken Thompson<sup>2</sup>.

<sup>1</sup> Dennis MacAlistair Ritchie, Científico de Computación estadounidense, (1941 - 2011).

<sup>2</sup> Kenneth Lane Thompson, Master en ciencias de la computación, (n. en 1943).

Es importante citar que la mayoría de los Sistemas Operativos de la actualidad se escriben en C. Una de las características principales de este lenguaje es que el mismo es independiente de la plataforma de hardware, además de ser estandarizado, esto permite que las aplicaciones desarrolladas en él puedan ejecutarse en cualquier computadora, con lo que se logra una importante portabilidad.

## COMENZANDO A PROGRAMAR

Se comentó en párrafos anteriores que en un programa se escriben las instrucciones para resolver un problema determinado utilizando un algoritmo, ejemplos de algoritmo pueden ser el algoritmo de la división, o el algoritmo de la suma, a continuación escribiremos un algoritmo que determinará si un número N, ingresado arbitrariamente es par o no, por supuesto debemos seguir para diseñar nuestro algoritmo una serie de pasos, expresados como la estructura básica de un algoritmo:

1. inicio
2. constantes (datos que no se pueden alterar)
3. variables (datos que pueden variar en el desarrollo del algoritmo)
4. ingreso de datos (datos ingresados por el usuario que se guardarán en las variables)
5. proceso de operaciones (ejecución de algoritmo sobre las variables y constantes)
6. mostrar resultados (resultados de la operación realizada mediante el algoritmo)
7. fin

Ahora que conocemos cuales son los pasos a seguir para crear un algoritmo podemos crear el algoritmo que puede determinar si un número es par o no:

Inicio del algoritmo:

Variables:

`Par`  $\leftarrow$  0 (se declara la variable entera `Par` y se le asigna el valor 0, el símbolo  $\leftarrow$  significa asignación)  
`N`  $\leftarrow$  0 (se declara la variable entera `N` y se le asigna el valor 0)  
`Auxiliar`  $\leftarrow$  0 (se declara la variable entera `Auxiliar` y se le asigna 0)

Ingreso de datos:

Ingresar `N` (el usuario debe ingresar un `N` entero arbitrario)

Proceso de operaciones:

`Auxiliar`  $\leftarrow$  `N`/2 (a la variable `Auxiliar` se le asigna `N` dividido 2)  
`Par`  $\leftarrow$  (`Auxiliar` \* 2) - `N` (a la variable `Par` se le asigna el resultado del producto entre `Auxiliar` y 2, menos `N`)

Informes:

Si  $Par = 0$  (se determina por si, o por no, si el  $N$  es par o impar y se informa)

El  $N$  ingresado es par

Sino

El  $N$  ingresado es impar

Fin del algoritmo.

Se debe tener en cuenta que los cálculos se realizan con variables (datos) enteras ej. 4, 22, 150, 10 etc. Y no con variables reales (números con decimales) ej. 1.5, 32.65, 4.5, 3.3 etc., por lo tanto los resultados también serán expresados con números enteros.

Los algoritmos pueden ser expresados también en lenguaje coloquial, dicho lenguaje expresa mediante una serie de normas léxicas y gramaticales la concreción de un algoritmo de manera similar a la cual se utilizaría para escribirlo mediante una determinada sintaxis, pero sin llegar a la rigidez que dicha sintaxis requiere. Por tal motivo se convierte en una herramienta básica para el programador, ya que posibilita al mismo desarrollar programas que luego puedan ser pasados a cualquier lenguaje de programación fácilmente.

A continuación se muestra el algoritmo anteriormente desarrollado en lenguaje coloquial:

Inicio del algoritmo de búsqueda de números pares:

Declaración de variables:

A **Par**, **N** y **Auxiliar** se le asigna 0

Ingreso de datos :

Se ingresa **N**

Proceso de operaciones:

**Auxiliar** es igual a **N** dividido 2

**Par** es igual a **Auxiliar** por 2 menos **N**

Si **Par** es igual a 0

El **N** ingresado es un número par

Sino

El **N** ingresado es un número impar

Informar los resultados:

Se informa **N**

Fin del algoritmo.

Una vez diseñado el algoritmo y luego de corroborar su funcionamiento, se procede a codificar el mismo utilizando la sintaxis del lenguaje de programación elegido para tal propósito, pero antes de esto resolvamos algunos inconvenientes que tienen que ver con la programación.

Cuando programamos le indicamos a la máquina mediante instrucciones, órdenes, lo que debe realizar, éstas órdenes deben estar lo más detalladas posible. Una manera simple de aprender a programar es comenzar por escribir paso a paso una determina tarea, por ejemplo, podríamos expresar con lenguaje coloquial los pasos que se debieran seguir para cambiarle la pila a un mouse inalámbrico, el conjunto de instrucciones sería el siguiente:

1. Tomar al mouse.
2. Identificar donde se encuentra el porta pilas.
3. Quitarle la tapa al porta pilas.
4. Tomar la pila y Quitarle el envoltorio.
5. verificar la posición (polaridad) en el porta pilas.
6. Colocar la pila en el porta pilas del mouse respetando la polaridad de la misma.
7. Colocar la tapa al porta pilas.
8. Verificar que el mouse funcione correctamente.

Puede observarse que el listado de instrucciones contiene paso a paso las tareas a realizarse, es importante que las mismas sigan una secuencia lógica y se debe tener en cuenta de no dejar nada por sabido, recordemos que la computadora es una máquina y no efectuará ninguna tarea o proceso a no ser que se lo indiquemos, debemos saber también que el compilador (el programa que revisa el código y le traduce a la computadora las instrucciones a realizar) no salta ninguna línea de código, a no ser que se lo indiquemos, es decir que la programación “es secuencial”.

Como práctica resolver los siguientes ejercicios:

- I. Escribir en Lenguaje coloquial las instrucciones para lavar un vaso en una canilla con una esponja, agua y detergente.
- II. Escribir los pasos a seguir para cebar un mate, suponiendo que el mismo se servirá con azúcar.

## CONCEPTO DE VARIABLE

La programación se basa en la utilización de variables, una variable es un elemento al cual le damos un nombre y le atribuimos un determinado tipo de información, más técnicamente, cuando declaramos una variable, reservamos un espacio en la memoria de la computadora para almacenar un dato del tipo con el cual fue declarado. En todos los lenguajes de programación existen distintos tipos de datos, de variables, es decir que en algunas podremos almacenar números, en otras letras y en otras hasta una cadena de caracteres. Un ejemplo de variables sería:

A ← "Hola"

```
B ← "Mundo"
```

Una vez comenzado el programa podemos manipular las variables como deseemos, incluso las variables irán cambiando de valor en base a los procesos que vaya realizando el programa, por ejemplo:

```
C ← A + B  
Mostrar C
```

En este caso lo que hemos informado con la variable C es "Hola Mundo" ya que la misma contiene la suma de A y B.

Otro ejemplo, pero con variables numéricas sería el siguiente:

```
N1 ← 10  
N2 ← 20  
SUMA ← 0
```

Aquí finalizó el proceso de declaración de variables, ahora se realizará el proceso de operaciones.

```
SUMA ← N1+N2
```

Y para finalizar, la salida de datos

```
Mostrar SUMA
```

Como puede observarse, la salida por pantalla que verá el usuario será 30.

Dijimos que al declarar una variable se reserva un espacio en memoria, siempre que un valor se coloca en una posición de memoria, dicho valor reemplaza al valor anterior de esa ubicación. Debido a que la información almacenada en una posición de memoria se destruye cuando es reemplazada, al proceso de sobrescribir una variable se conoce con el nombre de "escritura destructiva".

Podemos desarrollar, con el uso de distintas variables estructuras que nos serán muy útiles en programación, dos de las estructuras comúnmente utilizadas al escribir un programa son las llamadas CONTADOR y SUMADOR.

## VARIABLE CONTADOR

Un contador en programación suele utilizarse para contar eventos que cumplen una determinada condición en el programa, por ejemplo:

```
Contador ← 0  
  
Contador ← Contador+1  
Contador ← Contador+1  
Contador ← Contador+1  
  
Mostrar Contador
```

Si se observa el programa, en el mismo se declara una variable contador a la cual se le asigna el valor 0, luego se ejecuta el proceso contador tres veces esta repetición hará que Si seguimos con detenimiento este pequeño programa la variable contador tiene al principio valor 0 luego se incrementa de 1 en 1 ya que la segunda línea indica la operación  $\text{Contador} \leftarrow \text{contador}+1$ , es decir que contador en la primer operación vale  $0+1$ , es decir 1, en la segunda operación vale  $\text{contador} +1$ , es decir 2, ya que contador tenía valor 1, en la tercera  $2+1$  y así mientras la variable contador siga incrementándose.

## VARIABLE SUMADOR

Un sumador en programación se utiliza de manera similar al contador pero en vez de contabilizar eventos los suma, como se puede observar en el siguiente ejemplo

```
sumador ← 0
valor ← 2
  sumador ← sumador + valor
valor ← 5
  sumador ← sumador + valor
valor ← 10
  sumador ← sumador + valor

Mostrar sumador
```

Observar que en este caso, a diferencia del contador, la variable sumador acumula el valor ingresado en `valor`, es decir que en primera instancia `sumador` vale 0 luego, suponiendo que a `valor` se le ingresa el número 2,  $\text{Sumador} \leftarrow \text{Sumador} + \text{valor}$  por lo tanto `sumador` vale 2, y así sucesivamente para cualquier valor que tome la variable `valor`, el mismo será acumulado en `sumador`.

## BIBLIOGRAFÍA

- Deitel & Deitel. (2004). *Cómo programar en C/C++ y Java*. Pearson. México.
- Herbert Schildt. (1996). *Borland C++ Manual de Referencia*. McGraw-Hill. México.
- Byron S. Gottfried. (2005). *Programación en C*. McGraw-Hill. España.
- Savitch Walter. (2007). *Resolución de problemas con C++*. Pearson. México.