

GLOSARIO DE TÉRMINOS DE PROGRAMACIÓN

ASCII (Código): Las computadoras solamente entienden números. El código ASCII es una representación numérica de un carácter como 'a' o '@', y muchos otros.

Como otros códigos de formato de representación de caracteres, el ASCII es un método para una correspondencia entre cadenas de bits y una serie de símbolos (alfanuméricos y otros), permitiendo de esta forma la comunicación entre dispositivos digitales así como su procesamiento y almacenamiento.

Para escribir un código ASCII, sea letra, carácter, signo o símbolo. Ejemplo: "@" : (Arroba) en ordenadores con sistema operativo Windows: 1) Presiona la tecla "Alt" en el teclado, y no soltarla. 2) Sin dejar de presionar "Alt", presiona en el teclado numérico el número "64", que es el número de la letra o símbolo "@" en el código ASCII.

Tabla de Códigos ASCII

Caracteres ASCII imprimibles					ASCII extendido (Página de código 437)								
32	espacio	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
33	!	65	A	97	a	129	ú	161	í	193	ł	225	ó
34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ô
35	#	67	C	99	c	131	â	163	ú	195	ł	227	Ò
36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ö
37	%	69	E	101	e	133	à	165	Ñ	197	+	229	Ö
38	&	70	F	102	f	134	â	166	"	198	ä	230	µ
39	'	71	G	103	g	135	ç	167	°	199	Ä	231	þ
40	(72	H	104	h	136	è	168	¿	200	ł	232	þ
41)	73	I	105	i	137	ë	169	®	201	Œ	233	Û
42	*	74	J	106	j	138	è	170	¬	202	ł	234	Ü
43	+	75	K	107	k	139	ÿ	171	½	203	Œ	235	Û
44	,	76	L	108	l	140	ı	172	¼	204	Œ	236	ý
45	-	77	M	109	m	141	ı	173	ı	205	=	237	ÿ
46	.	78	N	110	n	142	Ă	174	«	206	†	238	—
47	/	79	O	111	o	143	Ă	175	»	207	□	239	·
48	0	80	P	112	p	144	É	176	█	208	ø	240	≡
49	1	81	Q	113	q	145	æ	177	█	209	Ð	241	±
50	2	82	R	114	r	146	Æ	178	█	210	È	242	—
51	3	83	S	115	s	147	ö	179	—	211	È	243	¼
52	4	84	T	116	t	148	ö	180	—	212	È	244	¶
53	5	85	U	117	u	149	ò	181	—	213	ı	245	§
54	6	86	V	118	v	150	ù	182	—	214	ı	246	+
55	7	87	W	119	w	151	ù	183	—	215	ı	247	·
56	8	88	X	120	x	152	ÿ	184	—	216	ı	248	·
57	9	89	Y	121	y	153	Ö	185	—	217	ı	249	·
58	:	90	Z	122	z	154	Ü	186	—	218	ı	250	·
59	;	91	[123	{	155	ø	187	—	219	ı	251	·
60	<	92	\	124		156	£	188	—	220	ı	252	·
61	=	93]	125	}	157	ø	189	—	221	ı	253	·
62	>	94	^	126	~	158	x	190	—	222	ı	254	·
63	?	95	_			159	f	191	—	223	ı	255	nbsp

Algoritmo: La palabra algoritmo se deriva de la traducción al latín de la palabra árabe alkhwarizmi, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

La solución a cualquier problema de cómputo involucra la ejecución de una serie de acciones en orden específico. Un procedimiento para resolver un problema en términos de: a) Las acciones a ejecutarse y b) el orden en el cual estas acciones deben ejecutarse se llama algoritmo.



La palabra algoritmo se deriva de la traducción al latín de la palabra árabe alkhwarizmi, nombre de un matemático y astrónomo árabe que escribió un tratado sobre manipulación de números y ecuaciones en el siglo IX.

La solución a cualquier problema de cómputo involucra la ejecución de una serie de acciones en orden específico. Un procedimiento para resolver un problema en términos de: a) Las acciones a ejecutarse y b) el orden en el cual estas acciones deben ejecutarse se llama algoritmo.

Las características de un buen algoritmo son:

- Debe tener un punto particular de inicio.
- Debe ser definido, no debe permitir dobles interpretaciones.
- Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.
- Debe ser finito en tamaño y tiempo de ejecución.

Codificación: La codificación es la operación de escribir la solución del problema (de acuerdo a la lógica del diagrama de flujo o pseudocódigo), en una serie de instrucciones detalladas, en un código reconocible por la computadora, la serie de instrucciones detalladas se le conoce como código fuente, el cual se escribe en un lenguaje de programación o lenguaje de alto nivel.

Comentarios: Pueden aparecer en cualquier parte del programa, mientras estén situados entre los delimitadores `/* comentario */`. Los comentarios son útiles para identificar los elementos principales de un programa o para explicar la lógica subyacente de estos.

Constantes: Son los valores que no pueden ser modificados. Pueden ser de cualquier tipo de datos.

Diagrama de flujo: Es la representación gráfica de un algoritmo. También se puede decir que es la representación detallada en forma gráfica de cómo deben realizarse los pasos en el programa para producir resultados.

Esta representación gráfica se da cuando varios símbolos (que indican diferentes procesos en la computadora), se relacionan entre sí mediante líneas que indican el orden en que se deben ejecutar los procesos.

- Todo diagrama debe tener un inicio y un fin.
- Se deben usar solamente líneas de flujo horizontales y/o verticales.
- Se debe evitar el cruce de líneas utilizando los conectores.
- Se deben usar conectores solo cuando sea necesario.
- No deben quedar líneas de flujo sin conectar.

Se deben trazar los símbolos de manera que se puedan leer de arriba hacia abajo y de izquierda a derecha.

Documentación: Es la guía o comunicación escrita en sus variadas formas, ya sea en enunciados, procedimientos, dibujos o diagramas.

A menudo un programa escrito por una persona, es usado por otra. Por ello la documentación sirve para ayudar a comprender o usar un programa o para facilitar futuras modificaciones (mantenimiento).

La documentación se divide en tres partes:

- **Documentación Interna:** Son los comentarios o mensaje que se añaden al código fuente para hacer mas claro el entendimiento de un proceso.
- **Documentación Externa:** Se define en un documento escrito los siguientes puntos:
 - Descripción del Problema
 - Nombre del Autor
 - Algoritmo (diagrama de flujo o pseudocódigo)
 - Diccionario de Datos
 - Código Fuente (programa)
- **Manual del Usuario:** Describe paso a paso la manera como funciona el programa, con el fin de que el usuario obtenga el resultado deseado.

Estructura:

- 1) Una clase Python la cual controla la generación de un árbol de carpetas que contiene archivos.
- 2) Una unidad de carpetas y archivos proveídos por el sistema templer para ser usado en una plantilla o plantillas. Las estructuras proporcionan recursos estáticos compartidos, que pueden ser utilizados por cualquier paquete en el sistema de templer.

Las estructuras se diferencian de las plantillas en que no proporcionan las variables

- **Estructuras de control:** Son los métodos que existen para dirigir el flujo de acciones que la computadora deberá ejecutar sobre los datos manejados por el programa.
- **Estructuras de datos:** Los hechos reales, representación en forma de datos, manera en que se organizan los datos.

Fases para la creación de un programa:

1. **Definición del Problema:** Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa. Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se conozca del todo no tiene mucho caso continuar con la siguiente etapa.
2. **Análisis del Problema:** Una vez que se ha comprendido lo que se desea de la computadora, es necesario definir:
 - Los datos de entrada.
 - Cuál es la información que se desea producir (salida).
 - Los métodos y fórmulas que se necesitan para procesar los datos.

Función: En programación, es una subrutina o subprograma (también llamada procedimiento, **función** o rutina), como idea general, se presenta como un subalgoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica. Se la invoca cada vez que se necesita realizar la tarea resuelta en la función. Las **funciones** son la parte central de la **programación**.

Identificador: Un identificador es una serie de caracteres formados por letras, dígitos y el carácter subrayado (`_`) que no inicie con dígito, así mismo es el nombre que damos a todo lo que manipulamos dentro de un programa (variables, constantes, funciones, etc). Por ejemplo variables, constantes, funciones, tipos definidos por el usuario etc.

Lenguaje de programación: Sistema de símbolos y reglas que permite la construcción de programas con los que la computadora puede operar así como resolver problemas de manera eficaz.

Estos contienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada / salida, calculo, manipulación de textos, lógica / comparación y almacenamiento / recuperación.

Los lenguajes de programación se clasifican en:

- **Lenguaje Máquina:** Son aquellos cuyas instrucciones son directamente entendibles por la computadora y no necesitan traducción posterior para que la CPU pueda comprender y ejecutar el programa. Las instrucciones en lenguaje maquina se expresan en términos de la unidad de memoria más pequeña el bit (dígito binario 0 ó 1).
- **Lenguaje de Bajo Nivel** (Ensamblador): En este lenguaje las instrucciones se escriben en códigos alfabéticos conocidos como mnemotécnicos para las operaciones y direcciones simbólicas.
- **Lenguaje de Alto Nivel:** Los lenguajes de programación de alto nivel (Python, Java, Visual Studio, etc.) son aquellos en los que las instrucciones o sentencias a la computadora son escritas con palabras similares a los lenguajes humanos (en general en ingles), lo que facilita la escritura y comprensión del programa.

Librerías o bibliotecas. (del inglés *library*): Es un conjunto de subprogramas utilizados para desarrollar software. Las librerías contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular. Algunos programas ejecutables pueden ser a la vez programas independientes y librerías, pero la mayoría de éstas no son ejecutables. Ejecutables y librerías se comunican entre sí a través de un proceso conocido como enlace (o link), que por lo general es realizado por un software denominado enlazador o linker.

Tipos de Librerías:

- 1- **Estática**, también conocido como **archivo** es un fichero contenedor con varios archivos de código objeto empaquetados, que en el proceso de enlazado durante la compilación serán copiados y relocalizados (si es necesario) en el fichero ejecutable final, junto con el resto de ficheros de código objeto.

En este caso, la biblioteca actúa simplemente como un recipiente para ficheros de código objeto que no se diferencian (más que semánticamente) de los ficheros objeto intermedios producidos durante la etapa previa de compilación del programa.

- 2- **Dinámicas, vinculadas dinámicamente, o de vínculos dinámicos** son ficheros que contienen código objeto construido de forma independiente de su ubicación, de tal modo que están preparadas para poder ser requeridas y cargadas en tiempo de ejecución por cualquier programa, en lugar de tener que ser enlazadas, previamente, en tiempo de compilación. Por tanto, han de estar disponibles como ficheros independientes al programa ejecutable (generalmente en directorios del sistema). En el proceso de enlazado (en tiempo de compilación) se genera un fichero ejecutable con anotaciones de qué librerías dinámicas requiere (pero no de dónde encontrarlas), y funciones de "esbozo" que se encargan de delegar la llamada a la función al cargador dinámico (o **dynamic-loader**).

En el resto del programa, las llamadas a las funciones de la librería se cambian por una llamada a la función de esbozo generada por el enlazador.

Mantenimiento: Se lleva a cabo después de terminado el programa, cuando se detecta que es necesario hacer algún cambio, ajuste o complementación al programa para que siga trabajando de manera correcta. Para poder realizar este trabajo se requiere que el programa este correctamente documentado.

Operadores: Son símbolos que indica al compilador que realice operaciones lógicas o matemáticas específicas.

- **Operadores Lógicos:** se utilizan para establecer relaciones entre valores lógicos.
 - And Y
 - Or O
 - Not Negación
 - Prioridad de los Operadores Lógicos
 - Not And Or
- **Operadores de Asignación:** Se utilizan para formar expresiones en las que se asigna el valor de una expresión a un identificador.

Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las operaciones indicadas.

Una expresión consta de operadores y operandos. Según sea el tipo de datos que manipulan, se clasifican las expresiones en:

- Aritméticas
- Relacionales
- Lógicas

Operaciones primitivas elementales: Son acciones que se ejecutan sobre los datos para transformarlos en información.

Palabras reservadas: Son palabras que tienen un significado especial para el lenguaje y no se pueden utilizar como identificadores.

Programación estructurada: Método disciplinado de escribir programas que sean claros, que se demuestren que son correctos y fáciles de modificar

Prueba y Depuración: Los errores humanos dentro de la programación de computadoras son muchos y aumentan considerablemente con la complejidad del problema. El proceso de identificar y eliminar errores, para dar paso a una solución sin errores se le llama depuración.

La prueba consiste en la captura de datos hasta que el programa no presente errores (los más comunes son los sintácticos y lógicos).

Pseudocódigo: Es la representación narrativa de los pasos que debe seguir un algoritmo para dar solución a un problema determinado. El pseudocódigo utiliza palabras que indican el proceso a realizar.

Tipos de datos: Todos los datos tienen un tipo asociado con ellos. Un dato puede ser un simple carácter, tal como b, un valor entero tal como 35. El tipo de dato determina la naturaleza del conjunto de valores que puede tomar una variable.

Variables: Una variable es una posición de memoria donde se puede almacenar una valor que se usará en el programa. Es un identificador que puede tomar diferentes valores durante la ejecución del programa, dependiendo del tipo de datos que se le declaro a dicha variable.

- **Inicialización de variables:** Inicializar una variable es darle un valor inicial, después que se ha declarado pero antes de que se ejecuten las sentencias en las que se emplea.