



**Universidad Nacional Autónoma de México**

**Facultad de ingeniería**

**Estructura de Datos y Algoritmos 1**

**Actividad #1**

**Acordeón Lenguaje C**

**José Carlos Avalos Jasso**

**26/02/2021**

## **Lenguaje C**

### **Características**

#### **Alfabeto o conjunto de caracteres:**

- Caracteres alfabéticos
- Caracteres numéricos
- Caracteres especiales

#### **Gramática**

- Esquema de un Programa Fuente
- Directivas
- Declaraciones Globales
- Función Principal
- Funciones Secundarias

```
#include <biblioteca1.h>
```

```
#include <biblioteca2.h>
```

```
... declaraciones de funciones...
```

```
... definiciones (cuerpos de funciones)...
```

```
... declaraciones de variables globales...
```

```
main() /* nombre de la función principal*/
```

```
{
```

```
... cuerpo del main...
```

```
}
```

```
...otras definiciones de funciones...
```

#### **Esquema de la Función “main”**

```
main()
```

```
{
```

```
...declaraciones de variables locales...
```

```
...instrucciones a ejecutar...
```

```
}
```

#### **Directivas más comunes**

```
#include <librería.h>
```

```
#define expresión
```

#### **Archivos de cabecera**

- Entrada y salida de datos (stdio.h)
- Rutinas matemáticas (math.h)

#### **Manejo de Salida de datos –Texto-**

printf(“Texto que se muestra en pantalla”);cadenas (string.h)

#### **Tipos de datos**

Int	enteros (números enteros positivos y negativos)	4 byte -> 32 bits
char	caracteres (letras)	1 byte -> 8 bits
Float	números en coma flotante (números reales)	4 byte -> 32 bits
double	números en coma flotante de doble precisión	8 byte -> 64 bits

#### **Tipos modificados**

Modificador	Significado
short	entero corto (rango más pequeño)
Long	entero largo (rango más amplio)
unsigned	entero sin signo (0..N)
signed	entero con signo (- N-1 .. +N)

#### **Operadores aritméticos**

Operador	Operación
()	Paréntesis
%	Módulo (resto de la división entera)
*	Multiplicación

/	División
+	Suma
-	Resta

### Asignaciones

La forma de dar valor a una variable es  
variableSuma= expresión\_aritmética;

### Salida de datos –Contenido de Variables-

printf ( “cadena de formato”, arg1, arg2, ...  
argN );

cadena de formato aparecen:

- el texto que se desea imprimir
- caracteres especiales
- secuencias de escape
- indicaciones del formato de los argumentos

### Formatos de printf

- %d %i Número Entero
- %c %s Carácter, cadena
- %f Número Real

Secuencias de escape

- \n Salto de línea
- \t Tabulación
- \a Sonido

### Entrada de datos

scanf ( “formato”, & arg1, & arg2, ... );

### Sentencia simple

/\* sentencia compuesta: varias sentencias  
entre llaves\*/

{

Sentencia1

Sentencia2

...

SentenciaN

}

### Sentencia if

if (condición)

### Operadores booleanos

Operador	
A == B	si A es igual a B
A != B	si A es distinto de B
A > B	si A es mayor que B
A < B	si A es menor que B
A >= B	si A es mayor o igual que B
A <= B	si A es menor o igual que B

### Sentencia if –compuesta-

if (condición)

{

sentencia1

sentencia2

}

### Construcción if-else

if (condición)

sentencia1

else

sentencia2

### Condiciones complejas

Expresión	Resultado
E1 && E2	Cierta si E1 y E2 son ciertas (AND)
E1    E2	Cierta si E1 o E2 son ciertas (OR)

### Construcción switch

switch (expresión)

```
{  
case valor1:  
... sentenciasA...  
break; case valor2:  
... sentenciasB ...  
break;  
.  
.  
.  
default:  
... sentenciasN ...  
break;  
}
```

### *Estructuras de Repetición*

#### Ciclo while

```
while ( condición ){  
sentencias  
}
```

#### Ciclo for

```
for ( expresión_inicial; condición;  
expresión_de_paso )  
{  
sentencias ejecutables  
}
```

#### Ciclo do-while

Sintaxis:  
do {

sentencias ejecutables.

} while ( condición );

### Operadores avanzados

Operadores	Significado
A++, ++A	Incrementa en 1 el valor de A (A=A+1)
A--, --A	Disminuye en 1 el valor de A (A=A-1)
A+=x	A=A+x
A-=x	A=A-x
A*=x	A=A*x
A/=x	A=A/x

### *EXPRESIONES Y OPERADORES*

#### PRECEDENCIA DE LOS OPERADORES

Precedencia	Operadores	Asociatividad
0	() [] -> .	Izq. a derecha
1	sizeof (tipo) ! ~ ++ -- signo* &	derecha a izq.
2	* / %	izq. a derecha
3	+-	izq. a derecha
4	>	izq. a derecha
5	>=	izq. a derecha
6	== !=	izq. a derecha
7	&	izq. a derecha
8	^	izq. a derecha
9		izq. a derecha

10	&&	izq. a derecha
11		izq. a derecha
12	?:	Derecha a izq
13	= += -= *= etc	Derecha a izq.

### Símbolos especiales

\a	Alert(bit)
\b	Backspace
\f	Formfeed
\n	Newline
\r	carriage return
\t	horizontal tab
\v	vertical tab
\\	Backslash
\'	single quote
\"	double quote
\OOO	visualiza un carácter cuyo código ASCII es OOO en octal.
\xHHH	visualiza un carácter cuyo código ASCII es HHH en hexadecimal.
\n	Newline
\r	carriage return
\t	horizontal tab
\v	vertical tab
\\	Backslash
\'	single quote
\"	double quote
\OOO	visualiza un carácter cuyo código ASCII es OOO en octal.
\xHHH	visualiza un carácter cuyo código ASCII es HHH en hexadecimal.

### Formatos (para Datos) para lectura e impresión

%d, %i %o	entero decimal con signo entero octal sin signo
%u %x	entero decimal sin signo entero hexadecimal sin signo (en minúsculas)
%X	entero hexadecimal sin signo (en mayúsculas)
%f	coma flotante en la forma
%e	coma flotante en notación científica
%g %E	usar %e o %f, el más corto como e pero en mayúsculas
%G %c	como g pero en mayúsculas un carácter
%s	cadena de caracteres terminada en '\0'
%%	imprime el carácter %
%p	puntero
%l	entero largo

### Librería “math” ANSI C

#### Funciones

acos	asin	atan	atan2	pow	sqrt
ceil	cos	cosh	exp	sin	tan
fabs	floor	fmod	frexp	sinh	
ldexp	log	log10	modf	tanh	

### Librería “stdio” ANSI C

#### Funciones

clearerr	fclose	feof	ferror	ARREGLOS UNIDIMENSIONALES	Qena Le	
fflush	fgetc	fgetpos	fgets	Formato:	setbuf	setvbuf
fopen	fformato	fprintf	fputc	Tipo nombre	tmpfile	tmpnam
fputs	fread	freopen	fscanf	Formato:	printf[tam2];	vsprintf
fseek	fsetpos	ftell	fwrite	Declaración de apuntadores:	rewind	scanf
getc	getchar	gets	perror		sprintf	sscanf
printf	putc	putchar	puts	• int *a; //a apunta a un valor de		

## Librería “string” ANSI C

### Funciones

memchr	memcmp	memcpy	memmove	strpbrk	strchr
memset	strcat	strchr	strcmp	Operadores unitarios en apuntadores:	
strcoll	strcpy	strcspn	strerror	• & Para decir dirección de memoria.	strxfrm
strlen	strncat	strncmp	strncpy	• Para decir contenido de la dirección de memoria.	

## Librería “stdlib” ANSI C

### Funciones

abort	abs	atexit	atof	system	wcst
atoi	atol	bsearch	call	strtod	strtol
div	exit	free	getenv	realloc	srand
labs	ldiv	malloc	mb	qsor	rand
mbstowcs	mbtowc	strtol			

## Librería “time” ANSI C

### Funciones

asctime	clock	ctime	difftime	time	strftime
gmtime	localtime	mktime			

## ARREGLOS UNIDIMENSIONALES

### Formato:

TipoDato nombre[tamaño];

### Declaración de una estructura.

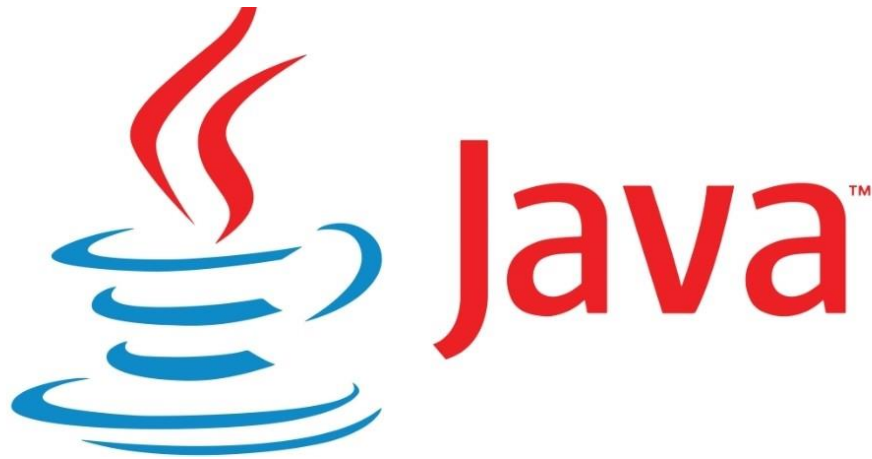
//Global

```
struct nombreEstructura{
variablesInvulocradas;
};
```

## *Lenguaje de programación con la letra J*

### **JAVA**

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.



## ***Referencias***

- ¿Qué es la tecnología Java y para qué la necesito? (s. f.). java.com. Recuperado 28 de febrero de 2021, de [https://www.java.com/es/download/help/whatis\\_java.html](https://www.java.com/es/download/help/whatis_java.html)