



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE INGENIERÍA

---

## ESTRUCTURA DE DATOS Y ALGORITMOS I

Actividad 1: Acordeón de lenguajes de programación.

Laura Mildred Moreno Razo

**FECHA: 26/02/2021**

## Lenguaje C

Es un lenguaje de propósito general basado en el paradigma estructurado. El teorema del programa estructurado dicta que todo programa puede desarrollarse utilizando únicamente 3 instrucciones de control:

Secuencia

Selección

Iteración

Consiste en una o más funciones, de las cuales una de ellas debe llamarse `main( )` y es la principal.

Comentarios. Una línea `/*`      Varias líneas `/*`    `*/`

Agregar bibliotecas. `#include`    Entrada y salida estándar. `stdio.h`

Otras `stdlib.h` `math.h` `string.h`

Declarar variables.      `[modificadores] tipoDeDato identificador [= valor];`

Tipos de variable

Enteras

<i>Tipo</i>	<i>Bits</i>	<i>Valor Mínimo</i>	<i>Valor Máximo</i>
<i>signed char</i>	8	-128	127
<i>unsigned char</i>	8	0	255
<i>signed short</i>	16	-32 768	32 767
<i>unsigned short</i>	16	0	65 535
<i>signed int</i>	32	-2,147,483,648	2 147 483 647
<i>unsigned int</i>	32	0	4 294 967 295
<i>signed long</i>	64	9 223 372 036 854 775 808	9 223 372 036 854 775 807
<i>unsigned long</i>	64	0	18 446 744 073 709 551 615
<i>enum</i>	16	-32 768	32 767

Si se omite el clasificador por defecto se considera 'signed'.

Reales

<i>Tipo</i>	<i>Bits</i>	<i>Valor Mínimo</i>	<i>Valor Máximo</i>
<i>float</i>	32	3.4 E-38	3.4 E38
<i>double</i>	64	1.7 E-308	1.7 E308
<i>long double</i>	80	3.4 E-4932	3.4 E4932

Tipo de dato	Especificador formato
Entero	%d Entero en base 10 %i
	%ld Entero largo en base 10 %li
	%o Entero base 8
	%x Entero base 16
Flotante	%f reales de precisión simple
	%lf a reales de doble precisión
	%e notación científica
	%g (redondea la parte fraccionaria a 3 dígitos significativos)
Carácter	%c carácter
	%d valor del código ASCII del carácter en base 10 %i
	%o valor del código ASCII del carácter en base 8
	%x valor del código ASCII del carácter en base 16
Cadena de caracteres	%s (Arreglo)

Identificador. Notación de camello, los nombres de cada palabra empiezan con mayúscula y el resto se escribe con minúsculas

**printf** es una función para imprimir con formato

```
printf("El valor de la variable real es: %lf", varReal);
```

**scanf** es una función que sirve para leer datos de la entrada estándar (teclado)

```
scanf ("%i", &varEntera);
```

Secuencias escape.

\a carácter de alarma

\b retroceso

\f avance de hoja

\n salto de línea

\r regreso de carro

\t tabulador horizontal

\v tabulador vertical

'\0' carácter nulo

Modificadores. Van al inicio de la declaración de variables son const (impide que una variable cambie su valor durante la ejecución del programa) y static (indica que la variable permanece en memoria desde su creación y durante toda la ejecución del programa, es decir, permanece estática en la memoria)

## Operadores

### Aritmético

Operador	Operación
+	Suma
-	Resta
*	Multipliación
/	División
%	Módulo

Operador	Operación
==	Igual que
!=	Diferente a
<	Menor que
>	Mayor que
<=	Menor o igual
>=	Mayor o igual

### Lógico

Operador	Operación
>>	Corrimiento a la derecha
<<	Corrimiento a la izquierda
&	Operador AND
	Operador OR
~	Complemento ar-1

Operador	Operación
!	No
&&	Y
	O

## Estructuras de selección

### If

```
if (expresión_lógica) {  
    // bloque de código a ejecutar  
}
```

### If-else

```
if (expresión_lógica) {  
    // bloque de código a ejecutar
```

```
// si la condición es verdadera
} else {
// bloque de código a ejecutar
// si la condición es falsa
}
```

### Switch

```
switch (opcion_a_evaluar){
case valor1:
/* Código a ejecutar*/
break;
case valor2:
/* Código a ejecutar*/
break;
...
case valorN:
/* Código a ejecutar*/
break;
default:
/* Código a ejecutar*/
}
```

**Operador ternario.** Condición ? SiSeCumple : SiNoSeCumple

### *Estructuras de repeticion*

#### While

```
while (expresión_lógica) {
// Bloque de código a repetir
// mientras que la expresión
```

```
// lógica sea verdadera.
```

```
}
```

### Do-while

```
do {
```

```
/*
```

Bloque de código que se ejecuta

por lo menos una vez y se repite

mientras la expresión lógica sea

verdadera.

```
*/
```

```
} while (expresión_lógica);
```

### For

```
for (inicialización ; expresión_lógica ; operaciones por iteración) {
```

```
/*
```

Bloque de código

a ejecutar

```
*/
```

```
}
```

Arreglos.    tipoDeDato nombre[tamaño]

Multidimensionales.    tipoDato nombre[ tamaño ][ tamaño ]...[tamaño];

Declarar apuntadores. TipoDeDato \*apuntador, variable; apuntador = &variable;

Cuando a una variable le antecede un ampersand, se hace es accede a la dirección de memoria de la misma

Funciones

valorRetorno nombre (parámetros){ // bloque de código de la función }

Firma función valorRetorno nombre (parámetros);

Abrir archivo `*FILE fopen(char *nombre_archivo, char *modo`

Cerrar archivo `int fclose(FILE *apArch);`

Modos

r: Abre un archivo de texto para lectura.

w: Crea un archivo de texto para escritura.

a: Abre un archivo de texto para añadir.

r+: Abre un archivo de texto para lectura / escritura.

w+: Crea un archivo de texto para lectura / escritura.

a+: Añade o crea un archivo de texto para lectura / escritura.

rb: Abre un archivo en modo lectura y binario.

wb: Crea un archivo en modo escritura y binario.

Leer y escribir archivos

`char *fgets(char *buffer, int tamaño, FILE *apArch);`

`char *fputs(char *buffer, FILE *apArch);`

`int fprintf(FILE *apArch, char *formato, ...);`

`int fscanf(FILE *apArch, char *formato, ...);`

## R

R es un lenguaje Orientado a Objetos, es interpretado y no compilado

Los argumentos pueden ser objetos ("datos", formulas, expresiones, . . . ), algunos de los cuales ´ pueden ser definidos por defecto en la función

El nombre de un objeto debe comenzar con una letra (A-Z and a-z) y puede incluir letras, dígitos (0-9), y puntos (.)

Definir carpeta de trabajo. `setwd("ruta de acceso a la carpeta especificada")`

Sintaxis básica. `comando (datos a utilizar)`

Comentarios una línea : `"#"`

## Tipos de valores

1. numeric: valores numéricos, incluye decimales.
2. integer: números enteros, no incluye decimales.
3. character: valores alfanuméricos, es decir, letras, números y signos mezclados.
4. logical: valores lógicos, TRUE o FALSE.

Asignar valores en cierto sentido ><

objeto <- valores

Crear un vector.      función concatenar      identificador <- c(valores)

Tabla de frecuencias.      table (vector)

Crear base de datos.      objeto <- data.frame(variables)

Visualizar la base creada.      View(objeto)

Guardar base de datos.      save(objeto, file = "nombre\_archivo.extensión")

Eliminar elementos.      remove(objeto)

Eliminar todos los objetos.      rm(list = ls())

Las operaciones aritméticas en R se escriben con los siguientes operadores:

- Suma: +
- Diferencia: -
- Producto \*
- División /
- Potencia ^

Las operaciones de comparación en R se escriben con los siguientes operadores:

- Mayor que: >
- Mayor o igual que: >=
- Menor que: <
- Menor o igual que: <=
- Igual a: ==
- Distinto de: !=

Estas operaciones devuelven un valor lógico TRUE o FALSE.



Algunas funciones útiles con vectores son:

- Longitud o número de elementos: `length()`
- Mínimo y máximo: `min()`, `max()`
- Índices del valor máximo y mínimo: `which.max()`, `which.min()`
- Suma de los elementos o conteo de TRUEs: `sum()`
- Media y mediana: `mean()`, `median()`

Función `nombre.funcion(argumento.1, argumento.2, ..., argumento.n)`

Las variables definidas dentro del cuerpo de una función son locales, y desaparecen al terminar la ejecución de la función.

Estructuras de control.

Para crear un bucle repetitivo (un bucle `for`), la sintaxis es la siguiente:

```
for (i in listadevalores) { secuencia de comandos }
```

`While`

```
while ( condicion logica) { expresiones a ejecutar }
```

`If`

```
if (condicion) comando1 else comando2
```

## Referencias

- Facultad de ingeniería. (2018). Manual de prácticas del laboratorio de Fundamentos de programación [Ebook] Ciudad de Mexico. Recuperado de: <http://lcp02.fi-b.unam.mx/>
- Bruzzone, G. Capítulo 4 Uso básico de RStudio | RStudio para Estadística Descriptiva en Ciencias Sociales. Consultado el 27 Febrero 2021, de: <https://bookdown.org/gboccardo/manual-ED-UCH/uso-basico-de-rstudio.html>
- García, A. M. (25/02/2008). Algunas estructuras de programación. Creación de funciones en R. Obtenido el 01/03/2021, desde el sitio Web de : <http://ocw.uc3m.es/estadistica/aprendizaje-del-software-estadistico-r-un-entorno-para-simulacion-y-computacion-estadistica/algunas-estructuras-de-programacion-creacion-de-funciones-en-r>.