

**UNIVERSIDAD DE LIMA**

Facultad de Ingeniería y Arquitectura

Carrera de Ingeniería de Sistemas

# Sistemas Operativos

## Laboratorio 03

Introducción al Lenguaje C (II)



Agosto 2023



## Objetivos

1. Programar estructuras de control en C y operadores lógicos.
2. Analizar la programación, desarrollo, funcionamiento y resultados de los algoritmos planteados.

### 01

## Estructuras de Control

Escriba los programas en C, compile, ejecute y explique la funcionalidad de los programas desarrollados.

### Ejemplo 01:

```
#include <stdio.h>

int main()
{
    int a;

    for( a = 0 ; a < 10 ; a++ )
    {
        puts("Repite 10 veces");
    }

    return(0);
}
```

### Ejemplo 02:

```
#include <stdio.h>

int main()
{
    int a;

    printf("Pon un valor: ");
    scanf("%d", &a);

    if( a < 10 )
    {
        puts("Pusiste un valor menor a 10");
        a -= 5;
        printf("El valor ahora es %d\n", a);
    }

    printf("Gracias");
    return(0);
}
```

### Ejemplo 03:

```
#include <stdio.h>

int main()
{
    int a = 10;

    here:
        printf("%d\n",a);
        a--;
        if( a > 0 )
            goto here;

    return(0);
}
```

### Ejemplo 04:

```
#include <stdio.h>

int main()
{
    int a;
    printf("Ingrese un valor: ");
    scanf("%d",&a);
    do
    {
        printf("A repetir\n");
        a--;
    }
    while(a > 0);

    return(0);
}
```

### Ejemplo 05:

```
#include <stdio.h>

int main()
{
    int a,b;
    b = 10;

    for( a=1 ; a<10 ; a++)
    {
        printf("%2d %2d\n",a,b);
        b--;
    }
    return(0);
}
```

## 02

## Cadenas de caracteres

Los **arrays** son variables estructuradas, cada elemento se almacena de forma consecutiva en memoria. Las cadenas de caracteres son declaradas en C como arrays de caracteres.

**Ejemplo 06:** Declarar una variable cadena en C.

```
#include <stdio.h>

int main()
{
    char cadena01[] = "Saludos" ;
    char cadena02[50] = "Saludos" ;
    char cadena03[] = {'S','a','l','u','d','o','s','\0'} ;

    char cadena04[50] = {'S','a','l','u','d','o','s','\0'} ;
    return 0 ;
}
```

**Ejemplo 07:** Ingresar una cadena por ventana de terminal y mostrarla en pantalla,

```
#include <stdio.h>

int main()
{
    // Declarar e inicializar una cadena
    char cadena[50] ;
    // Leer una cadena en pantalla
    printf("Ingrese nombre: ") ;
    scanf("%s", cadena) ;
    // Mostrar una cadena en pantalla
    printf("Nombre: ") ;
    printf("%s\n") ;
    return 0 ;
}
```

```
#include <stdio.h>

int main()
{
    // Declarar e inicializar un cadena
    char cadena[50] ;
    // Leer una cadena en pantalla
    printf("Ingrese nombre: ") ;
    gets(cadena) ;
    // Mostrar una cadena en pantalla
    printf("Nombre: ") ;
    puts(cadena) ;
    return 0 ;
}
```

**Ejemplo 08:** Escriba el programa en C utilizando. Compile, ejecute y analice los resultados.

```
#include <stdio.h>
#include <string.h>

int main ()
{
    char cadena01[12] = "Hola";
    char cadena02[12] = "Mundo";
    char cadena03[12];
    int len ;

    /* copy str1 into str3 */
    strcpy(cadena03, cadena01);
    printf("strcpy( cadena03, cadena01) : %s\n", cadena03 );

    /* concatenates str1 and str2 */
    strcat( cadena01, cadena02);
    printf("strcat( cadena01, cadena02): %s\n", cadena01);

    /* total length of str1 after concatenation */
    len = strlen(cadena01);
    printf("strlen(cadena01) : %d\n", len );

    return 0;
}
```

## 03

### Ejercicios propuestos

**Ejercicio 01:** Considere el siguiente algoritmo:

```
INICIO
    IMPRIMIR "Introducir diámetro en metros"
    LEER d
    IMPRIMIR "Introducir altura en metros"
    LEER h
     $r \leftarrow d/2$ 
     $PI \leftarrow 3.141592$ 
     $v \leftarrow PI \times r^2 \times h$ 
    IMPRIMIR "Volumen del cilindro: "
    IMPRIMIR v
FIN
```

Codificar el programa en C

**Ejercicio 02:** A continuación, se muestra el siguiente código de programa:

```
#include <stdio.h>
```

```

int main()
{
    int x, y=1 ;
    float z ;

    x = 3/2 + 10 - 3 ;
    z = 3.0/2.0 + 10 - 3 ;
    printf ("%d -- %f\n", x, z) ;
    printf ("%d\n", DATO*3+1/2) ;
    printf ("%d\n", 5+2-1>5) ;
    printf ("%d\n", 3>1 || 4<1) ;
    printf ("%d\n", 5!=1 && DATO>2) ;
    y=x ;

    printf ("%d\n", x+y<=z) ;

    return (0) ;
}

```

Indicar cuál será el resultado de la ejecución del programa y proponer el algoritmo que refleje el desarrollo del código mostrado líneas arriba

**Ejercicio 03:** La empresa que fabrica un modelo de máquinas expendedoras de refrescos necesita un programa para estas máquinas que realice el cálculo de cuántas monedas de cada tipo debe devolver. Para ello, en primer lugar, se introducirá por teclado la cantidad a devolver en soles (múltiplo de 10 céntimos, que es la moneda más pequeña de la que se dispone), es decir, se tecleará 1.90 para 1 sol con 90 céntimos. Este programa escribirá en pantalla cuántas monedas de cada tipo hay que devolver teniendo en cuenta que:

- Se dispone de monedas de 50 céntimos, 20 céntimos y 10 céntimos
- Siempre se dispone de las monedas necesarias de cada tipo
- Si hay una cantidad que falte brindar cambio, se utilizará redondeo a la moneda de menor denominación
- Se debe devolver el menor número de monedas posible, es decir, intentar devolver con las de mayor valor
- Ejemplos:
  - Si se introduce la cantidad de 1 sol con 90 céntimos, el programa debe imprimir: 3 monedas de 50 céntimos, 2 monedas de 20 céntimos
  - Si se introduce la cantidad de 1 sol con 85 céntimos, el programa debe imprimir: 3 monedas de 50 céntimos, 1 moneda de 20 céntimos, 2 monedas de 10 céntimos
  - Si se introduce la cantidad de 1 sol con 20 céntimos, el programa debe imprimir: 2 monedas de 50 céntimos, 1 moneda de 20 céntimos, 0 monedas de 10 céntimos

Se solicita lo siguiente:

- a. Proponer el algoritmo correspondiente
- b. Codificar el programa en Lenguaje C