

# **Introducción a la programación en C (II)**

Sistemas Operativos  
Ciclo 2023\_2 – Semana\_03



- **UNIDAD 1: Fundamentos de Sistemas Operativos**
  - Semana\_01: Introducción a los Sistemas Operativos, Virtualización
  - Semana\_02: Cloud y funciones del Sistema Operativo
  - **Semana\_03: Introducción a la programación en C**
  - Semana\_04: Programación en C

# Agenda



1. Flujos de datos : Secuencia, selección, repetición
2. Manejo de cadenas



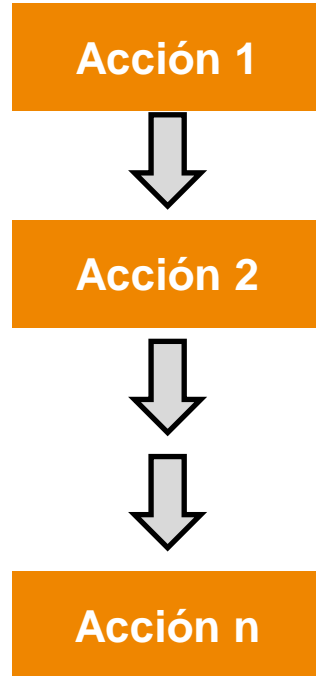
# **FLUJOS DE DATOS:**

## **SECUENCIA, SELECCIÓN, REPETICIÓN**

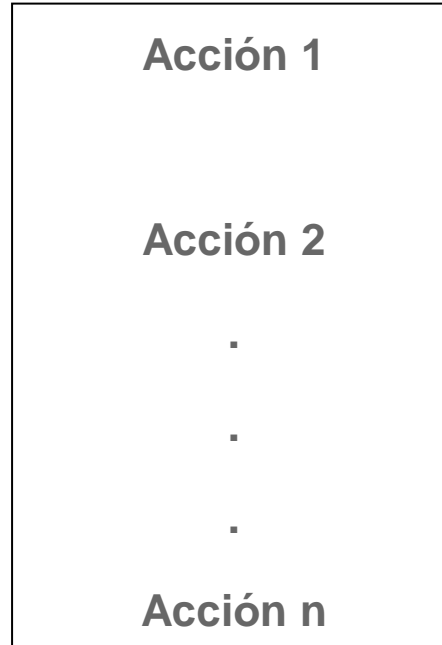


# SECUENCIA

## Diagrama de Flujo



## Pseudocódigo



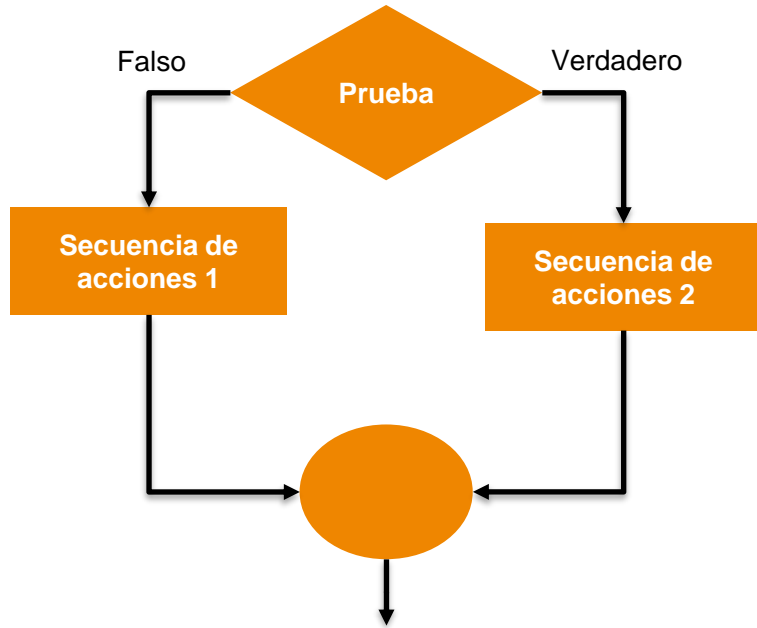
## Código C

```
#include<stdio.h>  
  
int main()  
{  
    printf("Ejemplo\n");  
    return 0;  
}
```

# SELECCIÓN



## Diagrama de Flujo



## Pseudocódigo

```
if (condición)
    then
        acción
        acción
        ...
    else
        acción
        acción
        ...
End if
```

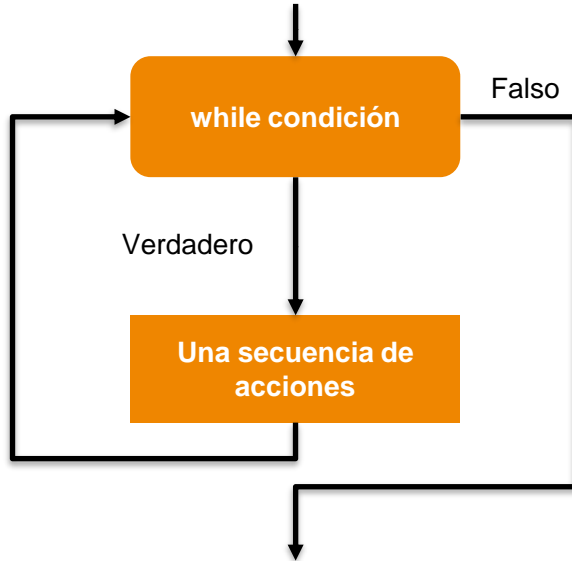
## Código C

```
#include <stdio.h>
int main ( )
{
    char a = 75;
    char b = 8;
    char resultado;
    printf ("a=%d b=%d",a,b);
    ➡ if (a & b == 8)
        printf ("El cuarto bit de la variable a es 1\n");
    else
        printf ("El cuarto bit de la variable a es 0\n");
    return 0 ;
}
```

# REPETICIÓN



Diagrama de Flujo



Pseudocódigo

```
while (condición)
    acción
    acción
    ...
End while
```

# EJERCICIO 1



Escriba un algoritmo en pseudocódigo que encuentre el promedio de dos números.





## EJERCICIO 2

Escriba un algoritmo para traducir una calificación numérica a una calificación aprobado/reprobado.

Calificaciones de 0 a 100. “Aprobado” si es mayor o igual a 70, “reprobado” si es menor que 70.

# EJERCICIO 3



Escriba un algoritmo para cambiar una calificación numérica a una calificación en letra.

Calificaciones de 90 a 100, A.

Calificaciones de 80 a 89, B.

Calificaciones de 70 a 79, C.

Calificaciones de 60 a 69, D.

Menor a 60, F.



## EJERCICIO 4

Escriba un algoritmo para imprimir los números pares  
mayores que 0 y menores que 100



# **MANEJO DE CADENAS**



# Cadenas de Caracteres

- Las cadenas de caracteres (también *strings*) en lenguaje C son vectores de *char*, con la particularidad que tienen una marca en el fin del mismo (el caracter '\0')
- El lenguaje nos permite escribir las cadenas como texto dentro de comillas dobles o son simples
- Ejemplos:

```
char cadena_hola[]="Hola";
```

```
char otro_hola[]={ 'H','o','l','a','\0'};
```

```
char vector[]={ 'H','o','l','a'};
```

```
char espacio_cadena[1024]="Una cadena en C";
```

```
char cadena_vacia[]="";
```

# Cadenas de Caracteres



- Ejemplo : El siguiente programa devuelve la cantidad de caracteres de una cadena sin contar el `'\0'`

```
int largo_cadena(char cadena[])
{
    int largo=0
    while (cadena[largo]!='\0') largo++;
    return largo;
}
```