



Universidad Tecnológica Nacional  
FACULTAD REGIONAL CORDOBA

# **PARADIGMAS DE PROGRAMACION**

## **Unidad II Paradigma Imperativo**

# Introducción

## El paradigma imperativo

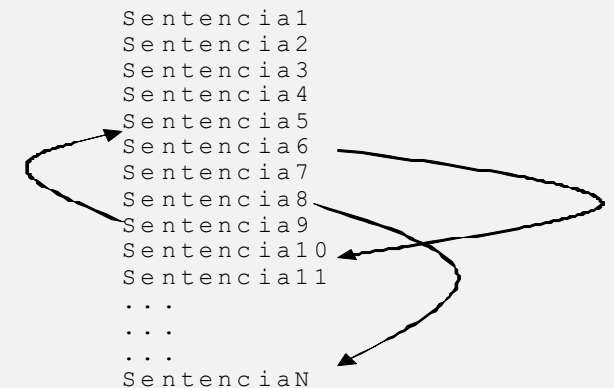
- Se basa en resolver los problemas a partir de la ejecución de programas que indican a la computadora un conjunto de acciones precisas a realizar.
- La ejecución de los programas se limita a procesar cada instrucción en el orden en que se han escrito.
- Al construir un programa se indica a la computadora “como” realizar el proceso.

**Algoritmos + Estructura de Datos = programa**

# Introducción

## Programación no estructurada

- Utiliza una sola instrucción para modificar la secuencia de ejecución de las instrucciones mediante una transferencia incondicional de su control (instrucción goto).
- Para comprender como funciona un programa es necesario simular su ejecución.
- Lenguajes, como Fortran, están basados en el uso de la instrucción goto.



# Introducción

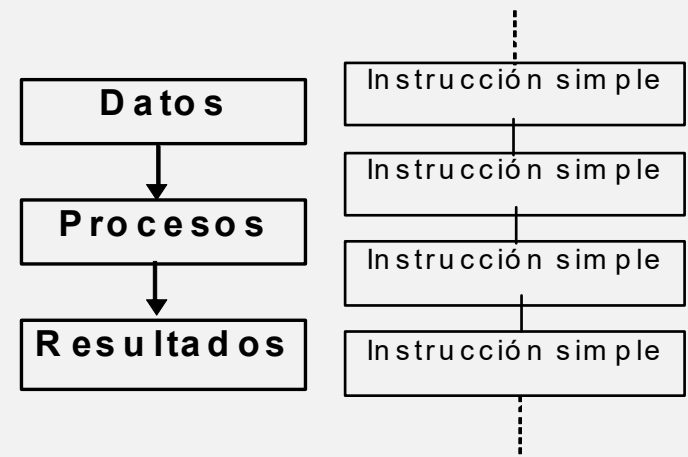
## Programación Estructurada

- Se basa en el teorema llamado “El teorema del programa estructurado” que establece que toda función computable puede ser implementada en un lenguaje de programación que combine sólo tres estructuras lógicas:
  - Secuencia: ejecución de una instrucción tras otra.
  - Selección: ejecución de una de dos instrucciones, según el valor de una variable booleana.
  - Iteración: ejecución de una instrucción (o conjunto) mientras una variable booleana sea 'verdadera'.

# Introducción

## Programación Estructurada

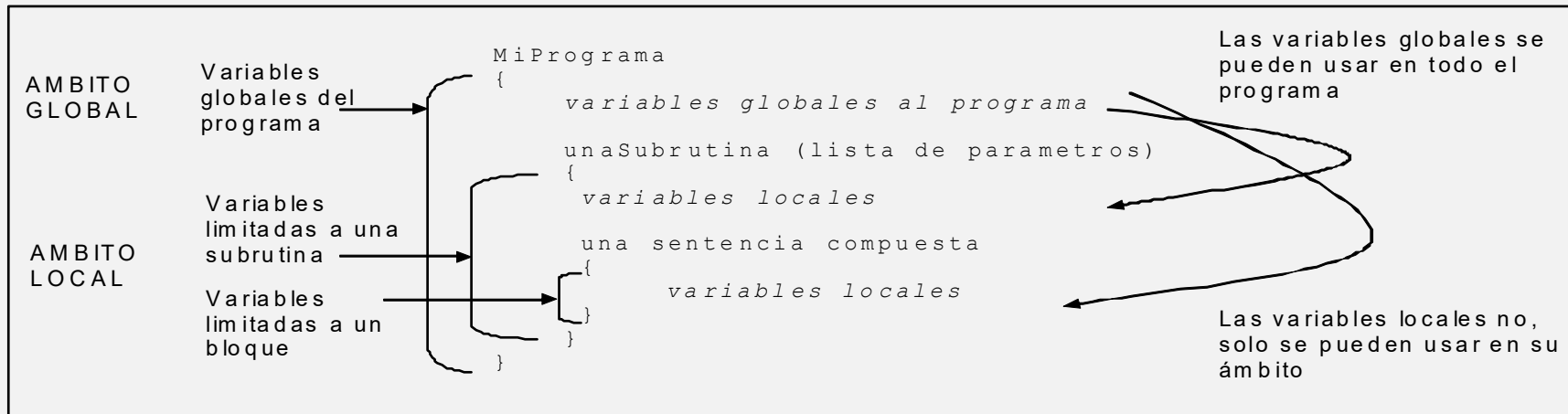
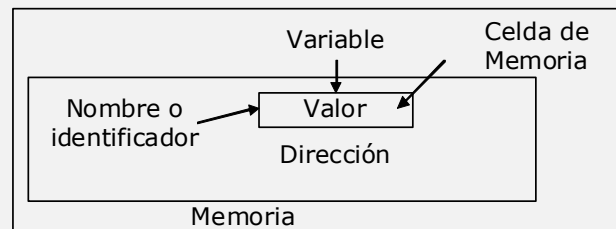
- Tiende a ser orientada a la **acción**, es la unidad fundamental.
- Los programadores se concentran en escribir **procesos**.
- Basada en la metodología de desarrollo de refinamientos sucesivos: Se plantea una operación como un todo y se divide en segmentos más sencillos o de menor complejidad.



# Programación Estructurada: Características Generales

- Variables Locales y Globales.
- Estructuras de control.
- Modularización.

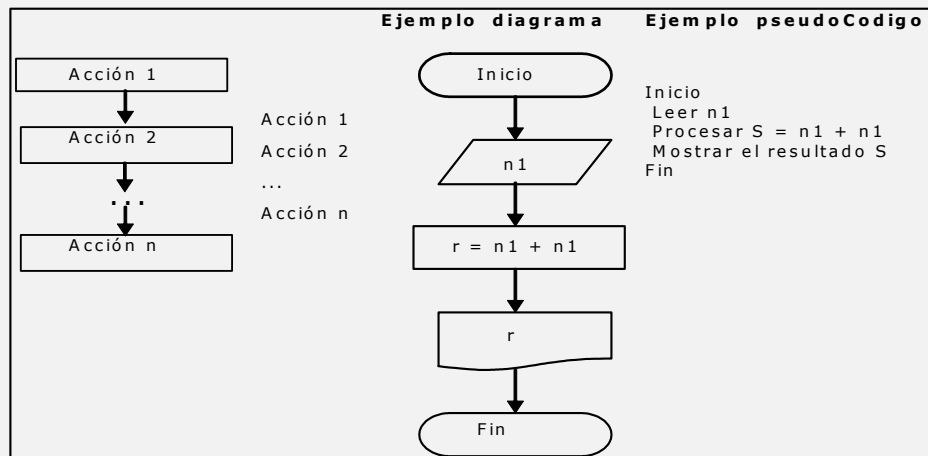
# Variables Locales y Globales



# Estructuras de control

## Estructura secuencial

- Es aquélla en la que una acción sigue a otra en secuencia.

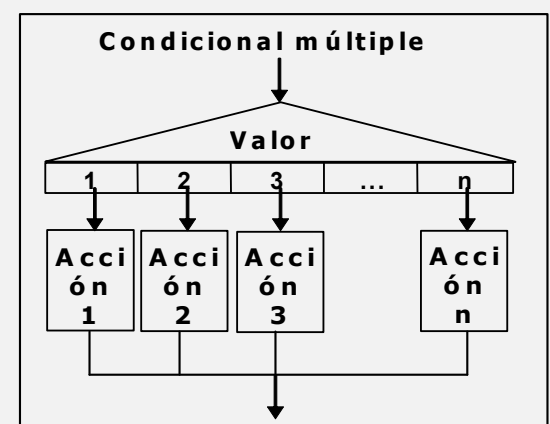
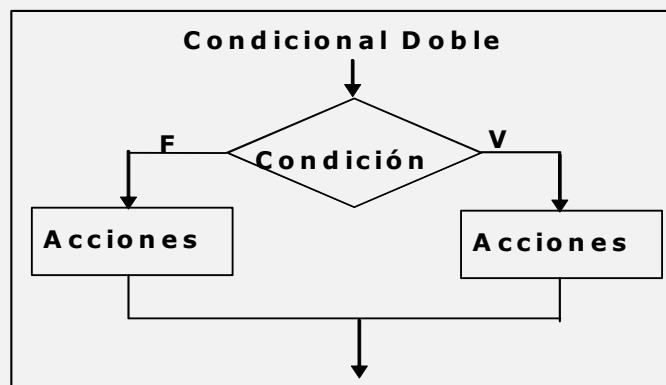
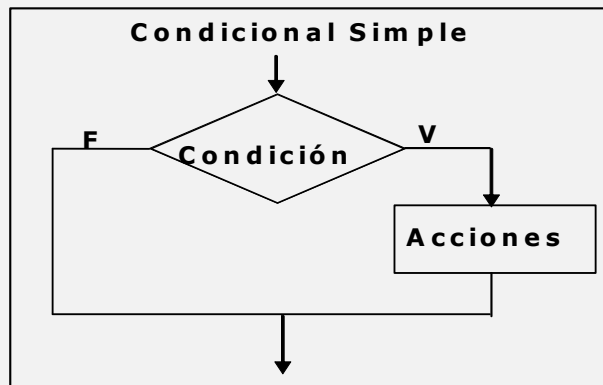




# Estructuras de control

## Estructuras condicionales, selectivas o alternativas

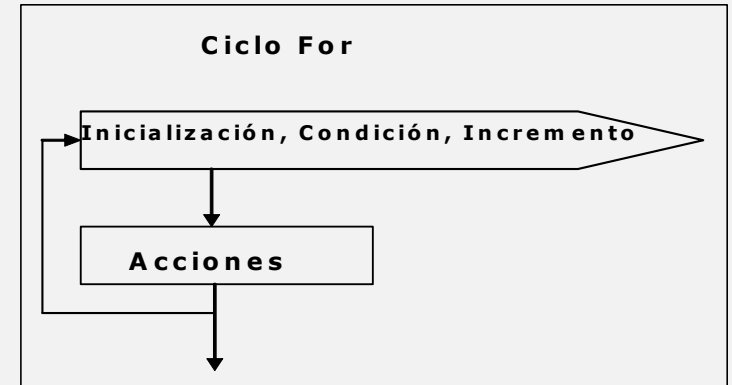
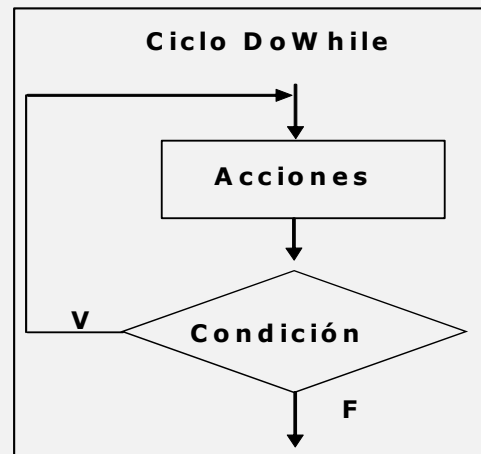
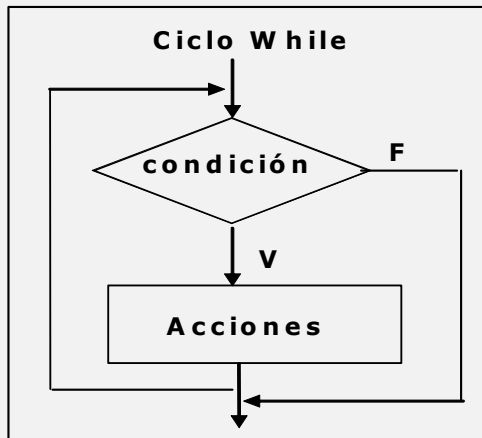
- Permiten evaluar condiciones y ejecutar acciones en función del resultado que arrojen estas condiciones.



# Estructuras de control

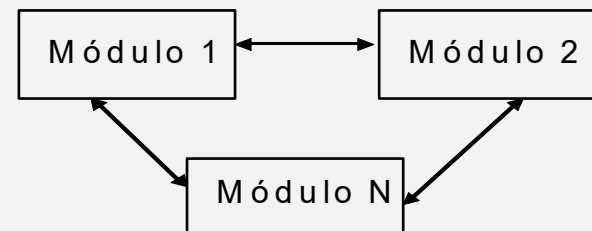
## Estructuras repetitivas

- Permiten repetir un conjunto de instrucciones y suelen llamarse ciclos, bucles, lazos, etc.



# Modularización

- Un módulo es una unidad de software que resuelve uno de los subproblemas en que se divide el problema complejo original.
- Cada uno tiene una tarea bien definida, se comunican con otros con una interfaz o protocolo.
- Cuando un módulo es invocado, recibe como entrada los datos proporcionados por otro módulo y le responde con la acción de una tarea y/o el envío de un resultado.



# Modularización: Características

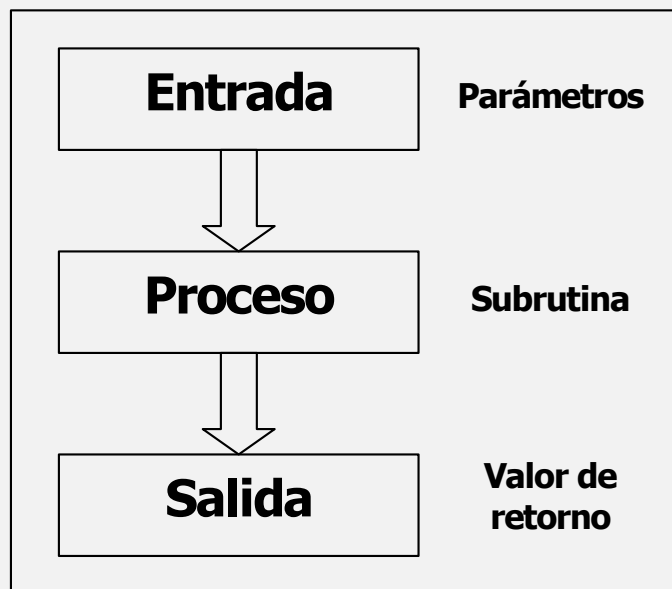
- **Tamaño relativamente pequeño:** Facilita aislar el impacto de cambios y/o rediseños en el programa.
- **Independencia modular:** Facilita el trabajo, no es necesario conocer los detalles internos de otros módulos, se cumplirá:
  - Características de **caja negra**, es decir abstracción y aislamiento de los detalles mediante encapsulamiento
  - Mejoras en el rendimiento humano, pudiendo realizarse programación en equipo y desarrollo de módulos paralelamente.
  - Contribuye a la reutilización de software.

# Procedimientos y Funciones

## Subrutina o Subprograma

- Es una porción de código que realiza una tarea específica y forma parte de un programa más grande.
- También llamada **procedimiento**, **función** o **rutina**.
- Una subrutina al ser llamada dentro de un programa hace que el código principal se detenga y se dirija a ejecutar el código de la subrutina, cuando esta termina el código principal continua con su ejecución.
- ***Ventajas***
  - Reducción de código duplicado.
  - Descomposición de problemas complejos en simples piezas (lo que aumenta la mantenibilidad y extensibilidad).
  - Aumenta la legibilidad del código de un programa.
  - Permite la reutilización de código en múltiples programas.

# Procedimientos y Funciones



- Un nombre único en el ámbito.
- Un tipo de dato de retorno.
- Una lista de parámetros.
- El código u órdenes de procesamiento.

# Procedimientos y Funciones

- **Procedimiento:** es un subproceso que puede recibir o no parámetros de entrada o de entrada / salida y realiza un conjunto de acciones sin retornar un valor.
- **Función:** es un subproceso que puede recibir o no parámetros de entrada o de entrada / salida y realiza un conjunto de acciones retornando un único resultado.

