

Computación I (CI-2125)

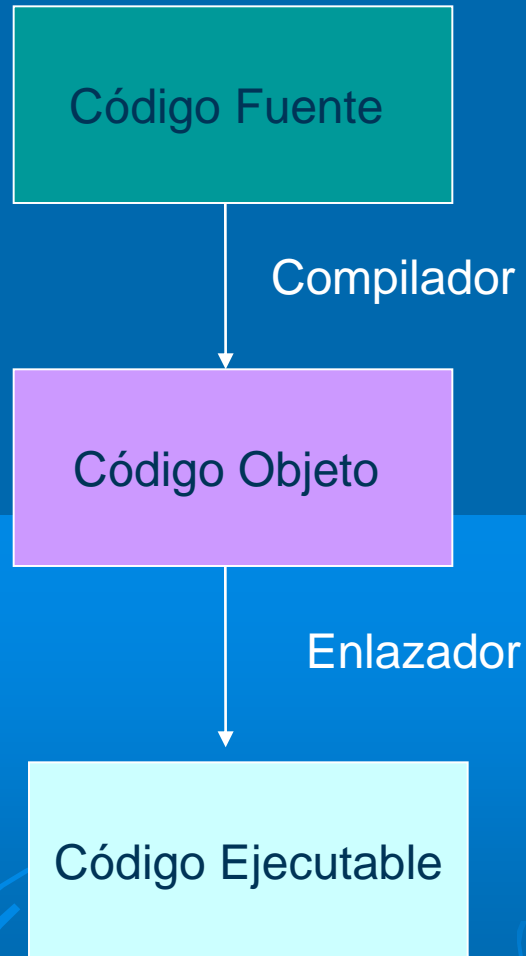
Clase 3

Prof. Mireya Morales Primera

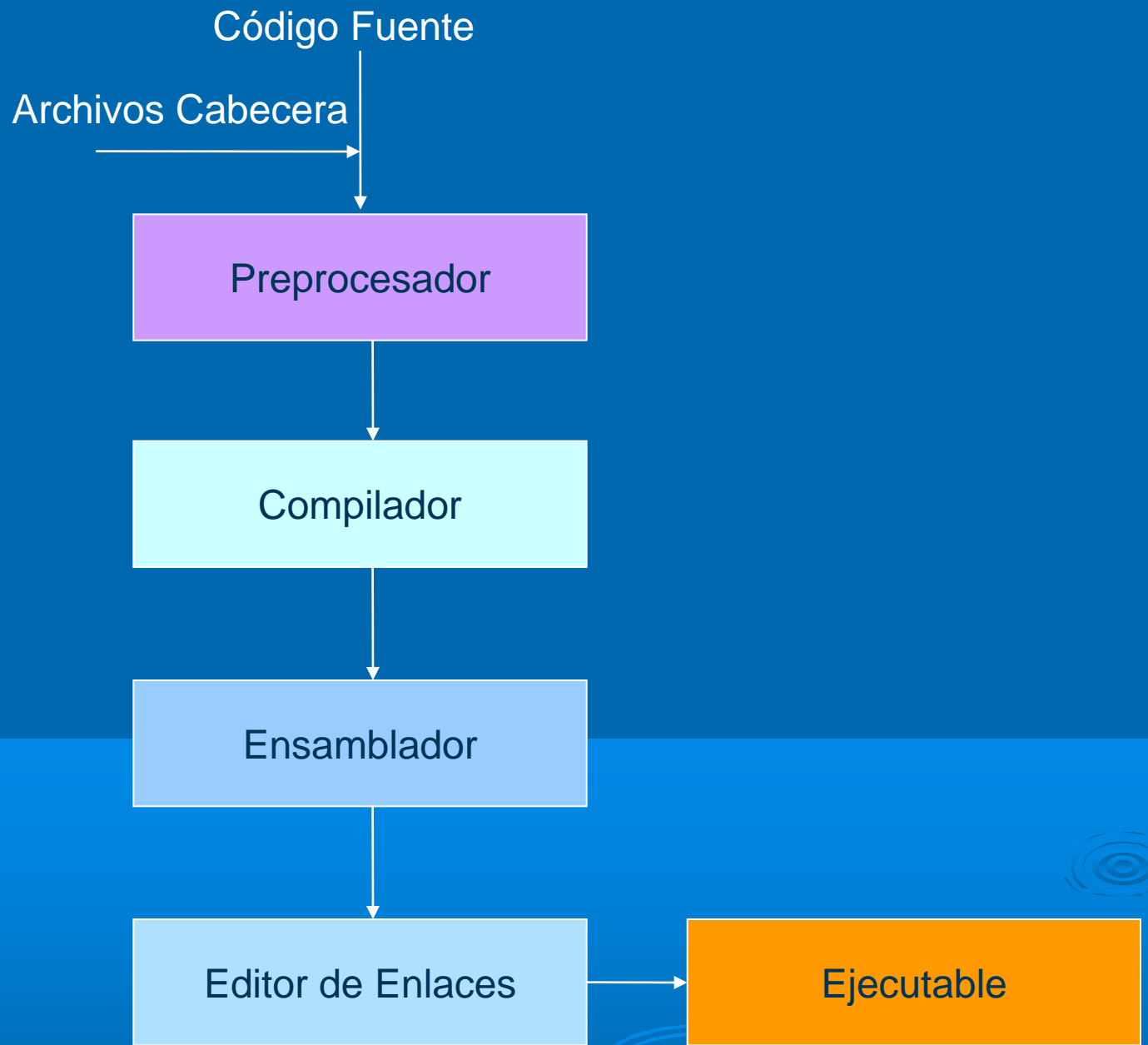
Contenido

- Ciclo de Desarrollo de un programa
- Formas de Representación Algorítmica:
Diagramas de flujo y pseudocódigo
- Estructuras de Control
 - Estructuras de secuencia
 - Estructuras de control condicionales
 - Simples: if
 - Dobles: if-else
 - Múltiples: switch

Ciclo de Desarrollo de un programa Archivos



Ciclo de Desarrollo de un programa



Diagramas de Flujo

- Es una forma de representación algorítmica que usa una esquematización gráfica.
- Muestra gráficamente los pasos a seguir para alcanzar la solución al problema.
- Todo esquema de representación algorítmica debe ser independiente del lenguaje.
- El diagrama de flujo representa la solución del problema. El programa representa la implementación

Diagramas de Flujo

Símbolos utilizados



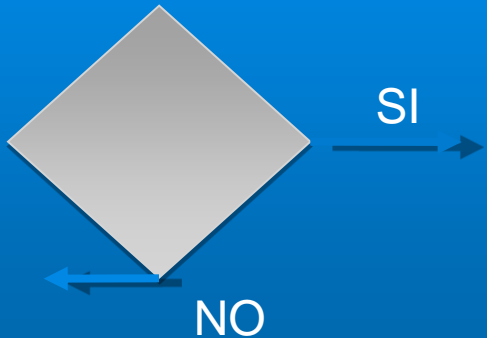
Marca el inicio y fin del diagrama



Se usa para introducir datos de entrada



Representa un proceso.
Asignaciones, operaciones aritméticas



Se utiliza para representar una decisión

Diagramas de Flujo

Símbolos utilizados



Se utiliza para representar una decisión múltiple: el switch



Representa la impresión de un resultado



Expresan la dirección del flujo del diagrama



Expresa conexión dentro de una misma página



Representar conexión entre páginas diferentes



Expresa un módulo de un subproblema que hay que resolver antes de continuar el DF.

Diagramas de Flujo

Ejemplo. Calcular el área de un triángulo

INICIO

BAS, ALT

*/*Lectura de datos*/*

AREA ← BAS*ALT/2

*/*Cálculo del Área*/*

AREA

*/*Escritura del resultado*/*

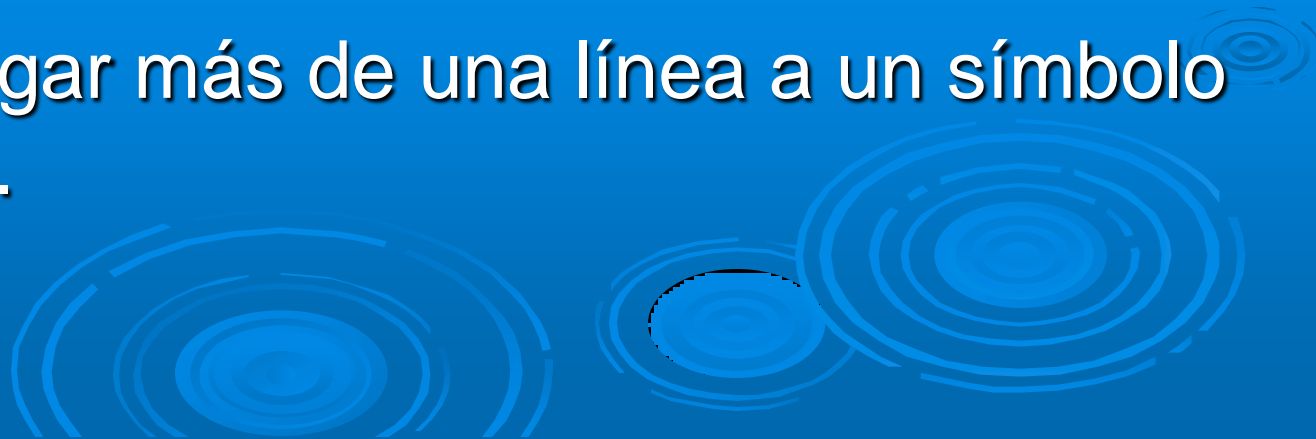
FIN

Diagramas de Flujo

- Todo diagrama de flujo debe tener inicio y fin.
- Las líneas utilizadas para indicar la dirección del flujo del diagrama, deben ser rectas: verticales u horizontales
- Todas las líneas utilizadas para indicar dirección del diagrama, deben estar conectadas.
- Debe construirse de arriba hacia abajo y de izquierda a derecha.
- La notación utilizada debe ser independiente del lenguaje de programación.

Diagramas de Flujo

- Si la construcción del diagrama de flujo requiere más de una hoja se deben utilizar los conectores adecuados y enumerar las páginas correspondientes.
- Al resolver un problema complejo es conveniente colocar comentarios que expresen o ayuden a entender la solución.
- No puede llegar más de una línea a un símbolo determinado.



Seudocódigo

- No se trata de un lenguaje de programación para las computadoras.
- Es un lenguaje artificial e informal que ayuda a desarrollar algoritmos.
- Se hacen con el objeto de ayudar al programador “a pensar” un programa, antes de intentar escribirlo en un lenguaje de programación.
- El pseudocódigo consiste sólo de caracteres, de modo que se puede hacer en papel o con algún editor de texto.

Seudocódigo

Ejemplo. Calcular el área de un triángulo

Inicio

Var BAS, ALT: Real /*datos de entrada*/

AREA: Real /*dato de salida */

LEER BAS

LEER ALT

$AREA \leftarrow BAS * ALT / 2$

ESCRIBIR AREA

Fin

Estructuras de Control

- Por lo regular, en un programa las instrucciones son ejecutadas una a una secuencialmente. Sin embargo, existen algunas instrucciones que le permiten al programador especificar que el enunciado siguiente a ejecutar, puede ser otro diferente al que le sigue en la secuencia. Esto es conocido como transferencia de control.
- Los investigadores Bohm y Jacopini demostraron que los programas podían ser escritos sin ninguna instrucción *goto*.

Estructuras de Control

- El trabajo de Bohm y Jacopini demostró que todos los programas podían ser escritos en términos de sólo tres estructuras de control:
 - Estructura de secuencia
 - Estructura de selección
 - Estructura de repetición



Estructura de Secuencia

Una estructura de secuencia es aquella en la que una acción (instrucción) sigue a otra en secuencia.

Acción 1



```
graph TD; A[Acción 1] --> B[Acción 2]; B --> C[Acción 3];
```

Acción 2

Acción 3

Pseudocódigo

var

Inicio

<acción1>

<acción2>

fin

Estructura de Secuencia

Una estructura de secuencia es aquella en la que una acción (instrucción) sigue a otra en secuencia.

Acción 1



```
graph TD; A[Acción 1] --> B[Acción 2]; B --> C[Acción 3];
```

Acción 2

Acción 3

Pseudocódigo

var

Inicio

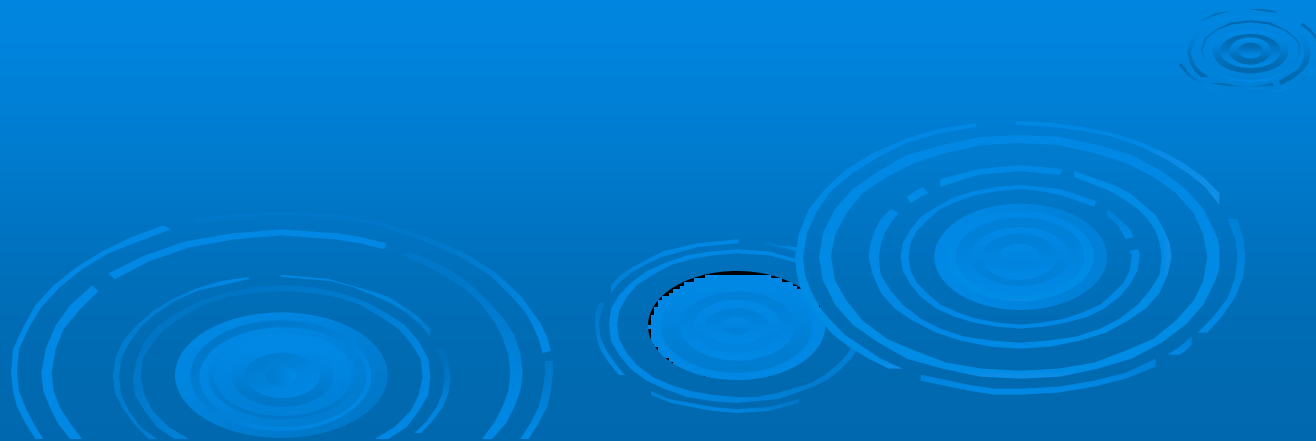
<acción1>

<acción2>

fin

Estructuras de Control Condicionales

- En las estructuras de control condicionales se evalúa una condición y en función del resultado de la misma se realiza una opción u otra.
- Las condiciones se especifican usando expresiones lógicas.
- Estas estructuras pueden ser:
 - **Simple**
 - **Dobles**
 - **Múltiples**



Estructuras de control simple

Seudocódigo: if

·

·

acción m

si <condición> entonces

<acción 1>

<acción 2>

·

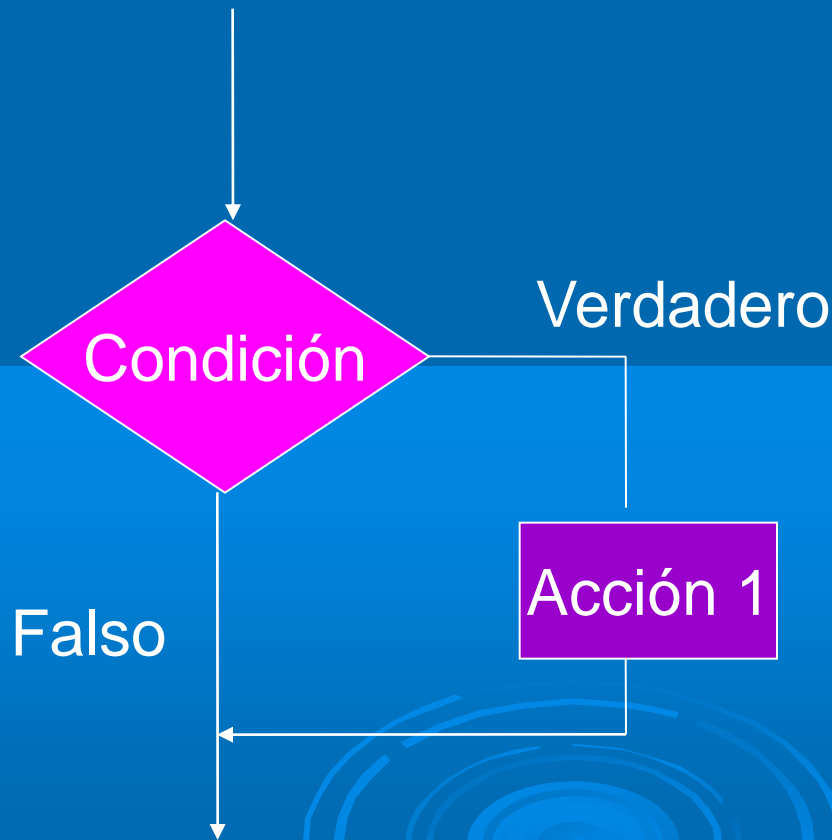
·

<acción n>

fin_si

Estructuras de control simple

Diagrama de flujo: if

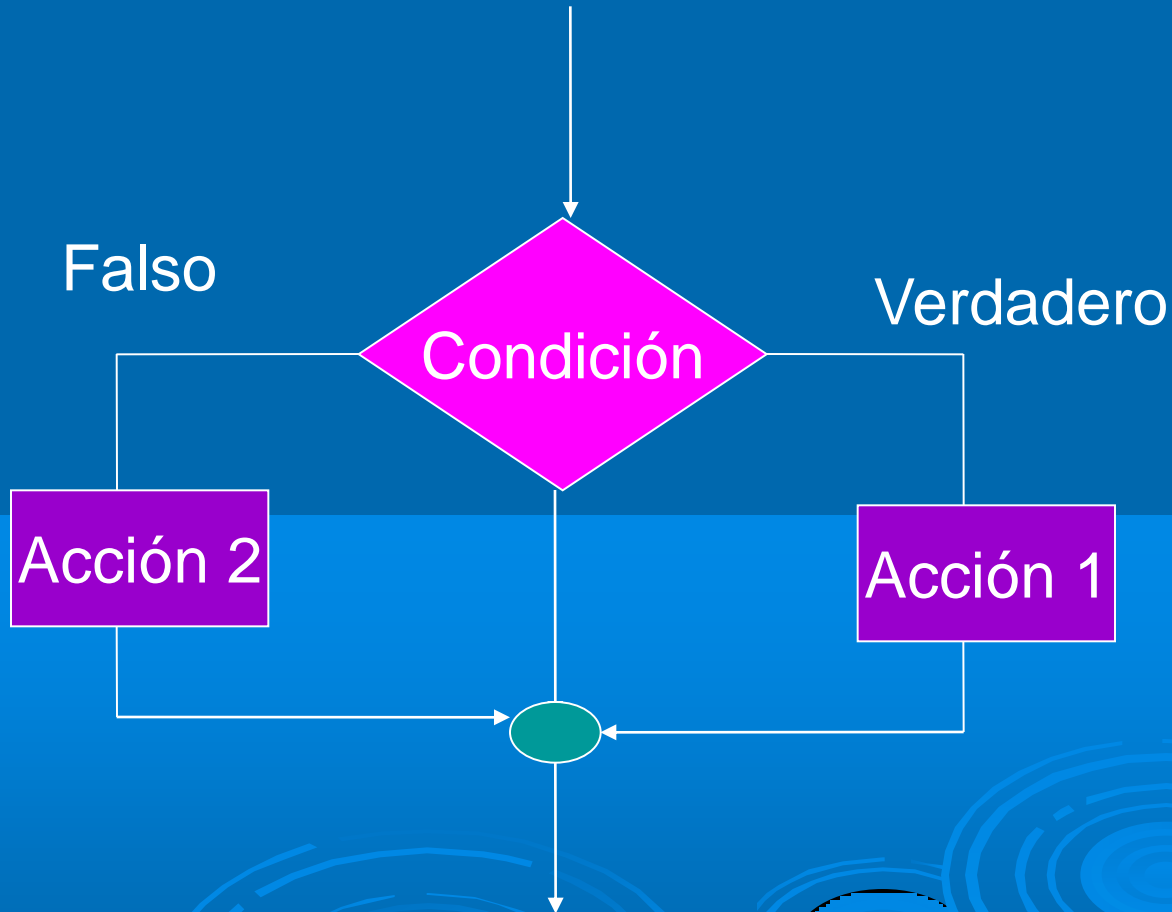


Estructuras de control doble

- **Seudocódigo: if else**
acción m
si <condición> entonces
 <acción 1.1>
 <acción 1.2>
 .
 .
 <acción 1.n>
si_no
 <acción 2.1>
 <acción 2.2>
 .
 .
 <acción 2.n>
fin_si

Estructuras de control doble

Diagrama de flujo: if else



Estructuras de Control Múltiples

Seudocódigo: switch

```
·  
acción m  
caso de E hacer  
e1: <acción 1.1>  
e2: <acción 2.1>  
·  
·  
en <acción n.1>  
si_no  
    <acción m.1>  
    <acción m.2>  
·  
·  
    <acción m.n>  
fin_caso
```

Estructuras de Control Múltiples

Diagrama de flujo: switch

