

Lógica de programación I

Juan Pablo Restrepo Uribe

Ing. Biomedico

MSc. Automatización y Control Industrial

jprestrepo@correo.iue.edu.co

2023-2

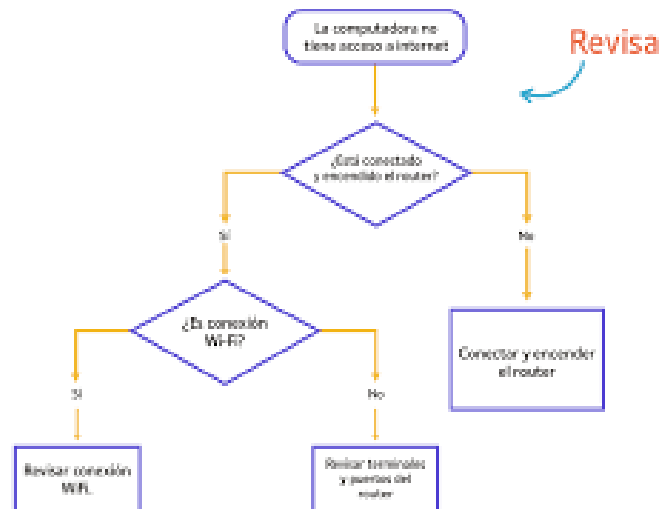
Institución Universitaria de Envigado

Unidad 1: Definiciones Básicas

- Formas de representar algoritmos
 - Pseudocódigo
 - Diagrama flujo
 - Diagrama de bloques (Nassi-Shneiderman).

Formas de representar algoritmos

Las principales maneras de representación de un algoritmo son a través de código fuente haciendo uso de un lenguaje de programación, a través de pseudocódigo, a través de un diagrama de flujo o a través de un sistema formal.



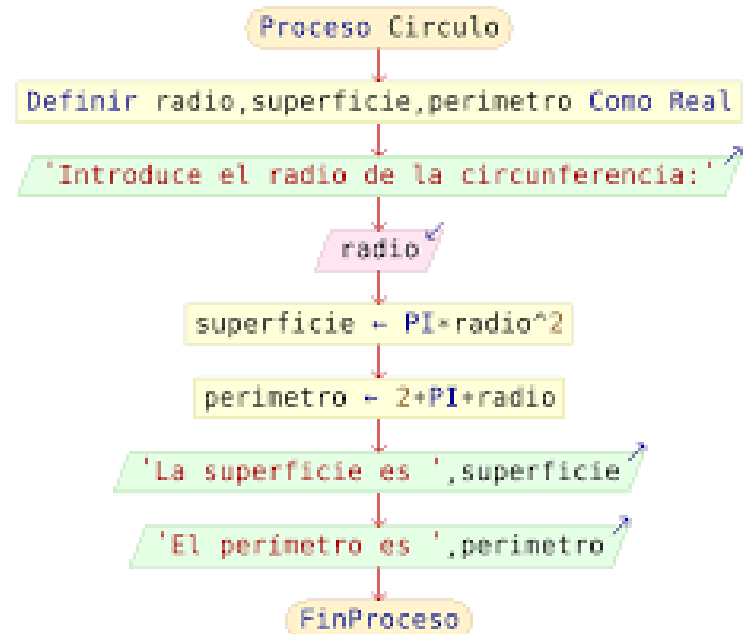
Formas de representar algoritmos: Pseudocódigo

El pseudocódigo de un programa informático representa la descripción del algoritmo en un lenguaje de alto nivel, sin que éste sea capaz de ser procesado por una máquina. Se considera uno de los pasos previos a la escritura del código fuente.

```
Procedimiento Ordenar (L)
//Comentario: L = (L1, L2, ..., Ln) es una lista con n elementos//
k ← 0;
Repetir
    intercambio ← Falso;
    k ← k + 1;
    Para i ← 1 Hasta n - k Con Paso 1 Hacer
        Si Li > Li+1 Entonces
            intercambiar (Li, Li+1)
            intercambio ← Verdadero;
        Fin Si
    Fin Para
Hasta Que intercambio = Falso;
Fin Procedimiento
```

Formas de representar algoritmos: Diagrama de flujo

Un diagrama de flujo representa de manera gráfica un algoritmo, teniendo en cuenta todos los posibles resultados y condiciones dentro de dicho gráfico. De manera general, se utiliza el lenguaje UML, el cual, es un lenguaje a través del que se representa el flujo de trabajo y que hace uso de una serie de símbolos de manera universal.

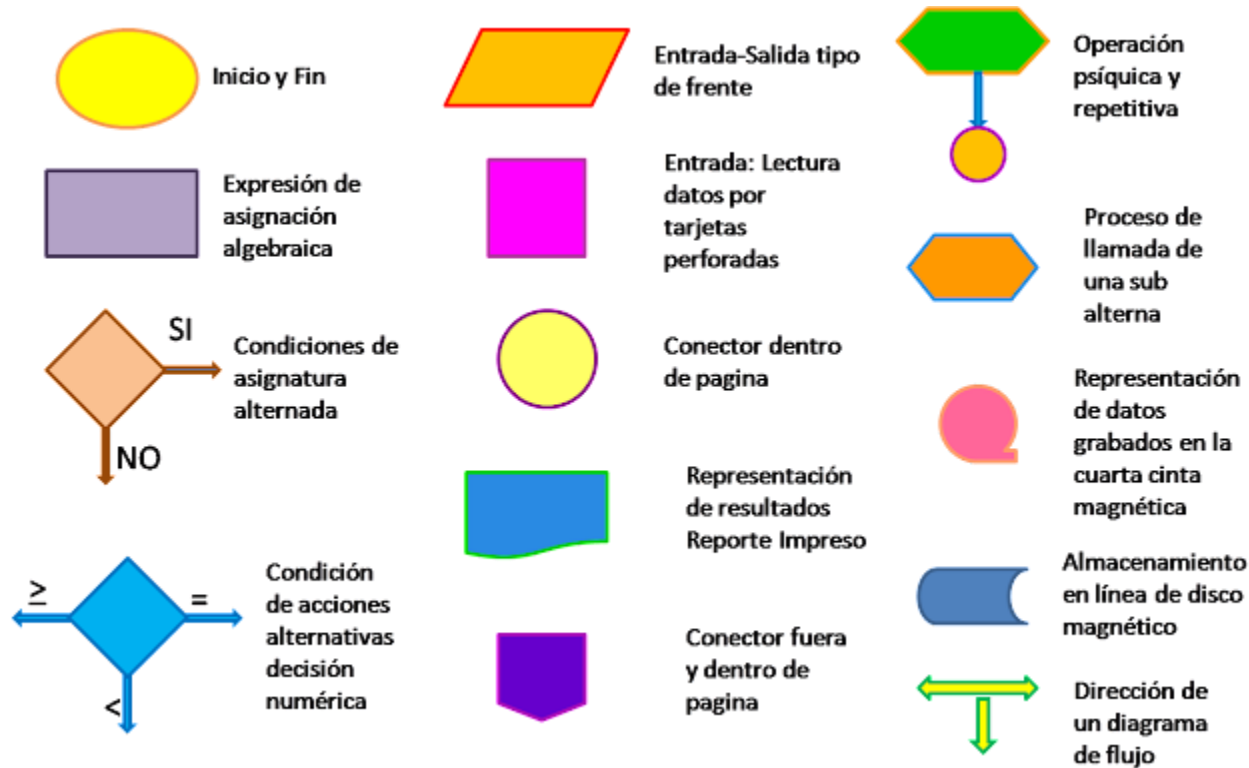


Formas de representar algoritmos

Principalmente, un diagrama de flujos está formado por diferentes elementos que nos permiten dar forma a la idea del algoritmo:

- **Línea o flechas del flujo:** Indica la instrucción que se va a realizar, o la dirección del flujo del proceso.
- **Inicio y final:** Es la forma en la cual se indica el “inicio del proceso” y “final del proceso”. Su icono suele ser un rectángulo con las esquinas redondeadas.
- **Asignación/ definición:** Permite asignar un valor o definir una variable, son los procesos o actividades que se necesitan para resolver el problema. En este caso, suele estar representado con un rectángulo.
- **Datos de entrada o de salida:** Representa la impresión de datos en la salida. Se representa con un recuadro con las esquinas inclinadas y una flecha hacia fuera.
- **Decisión:** Indica que des del punto que nos encontramos, puede haber más de un camino para seguir, según la condición dada. En este caso se usa un rombo.

Formas de representar algoritmos



```

504         if not hasattr(self, '_headers_buffer'):
505             self._headers_buffer = []
506         self._headers_buffer.append(("Xs Xd Xs\r\n" %
507                                     (self.protocol_version, code, message)).encode(
508                                         'latin-1', 'strict'))
509
510     def send_header(self, keyword, value):
511         """Send a MIME header to the headers buffer."""
512         if self.request_version != 'HTTP/0.9':
513             if not hasattr(self, '_headers_buffer'):
514                 self._headers_buffer = []
515             self._headers_buffer.append(
516                 ("%s: %s\r\n" % (keyword, value)).encode('latin-1', 'strict'))
517
518         if keyword.lower() == 'connection':
519             if value.lower() == 'close':
520                 self.close_connection = True
521             elif value.lower() == 'keep-alive':
522                 self.close_connection = False
523

```


Formas de representar algoritmos: Estructuras de control

En la redacción de pseudocódigo se utilizan tres tipos de estructuras de control: las secuenciales, las selectivas y las iterativas.

Estructuras secuenciales

Las instrucciones se siguen en una secuencia fija que normalmente viene dada por el número de renglón. Es decir que las instrucciones se ejecutan de arriba hacia abajo.

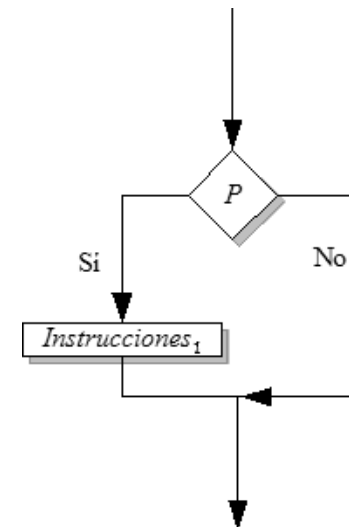
```
instrucción1;  
instrucción2;  
instrucción3;  
...  
instrucciónn;
```

Formas de representar algoritmos: Estructuras de control

Estructuras selectivas

Las instrucciones selectivas representan instrucciones que pueden o no ejecutarse, según el cumplimiento de una condición.

```
Si condición Entonces
    instrucciones;
Fin Si
```



Formas de representar algoritmos: Estructuras de control

Selectiva doble (alternativa)

La instrucción alternativa realiza una instrucción de dos posibles, según el cumplimiento de una condición.

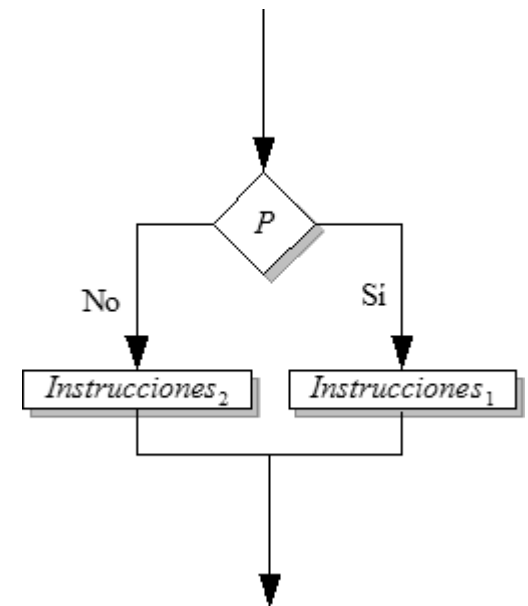
Si condición Entonces

instrucciones₁;

Si no Entonces

instrucciones₂;

Fin Si



Formas de representar algoritmos

Selectiva múltiple

También es común el uso de una selección múltiple que equivaldría a anidar varias funciones de selección.

```
Si condición1 Entonces  
    instrucciones1;  
Si no si condición2 Entonces  
    instrucciones2;  
Si no si condición3 Entonces  
    instrucciones3;  
...  
Si no Entonces  
    instruccionesn;  
Fin Si
```

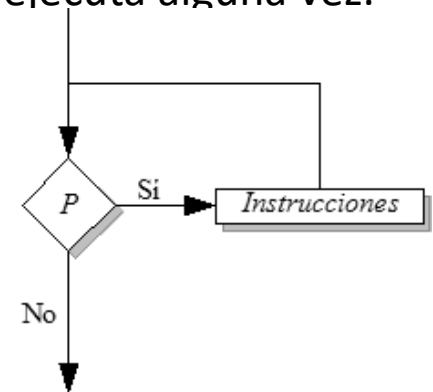
Formas de representar algoritmos: Estructuras iterativas

Las instrucciones iterativas representan la ejecución de instrucciones en más de una vez.

Bucle mientras

El bucle se repite mientras la condición sea cierta, si al llegar por primera vez al bucle mientras la condición es falsa, el cuerpo del bucle no se ejecuta alguna vez.

```
Mientras condición Hacer  
    instrucciones;  
Fin Mientras
```



Formas de representar algoritmos

Bucle para

Una estructura de control muy común es el ciclo FOR, la cual se usa cuando se desea iterar un número conocido de veces, empleando como índice una variable que se incrementa (o decrementa):

```
Para  $i \leftarrow x$  Hasta  $n$  Con Paso  $z$  Hacer  
    instrucciones;  
Fin Para
```