	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 4 Versión: 01
---	---	-------------------------

ASIGNATURA: Algoritmos y Programación 1	CÓDIGO: ING01179
DOCENTE: Luis Fernando González Alvarán	FECHA: abril 29/2022

SEMINARIO DE ENCUENTRO NÚMERO 4

PRESEMINARIO # 4

El estudiante debe realizar la siguiente lectura y complementarla con la bibliografía o los enlaces sugeridos, buscando temas relacionados con la unidad en estudio. Posteriormente realizará en el chat de la asignatura, un aporte a las preguntas orientadoras y publicará el resultado de los ejercicios propuestos

1.DISEÑO DEL ALGORITMO.

Esta actividad de la fase de resolución del problema corresponde al diseño de una solución basada en el análisis previamente realizado, la cual se realiza a través de un algoritmo que consiste en hacer una descripción de los pasos lógicos que dan solución al problema haciendo uso de diversas estructuras previamente definidas. Los algoritmos se pueden representar de forma gráfica o no gráfica.

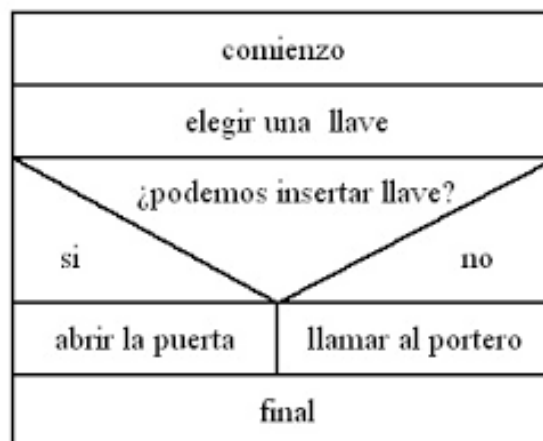


Aunque este curso se centra en el diseño de soluciones a través de pseudocódigos, a continuación, se da una breve descripción de los algoritmos gráficos.

1. Diagrama NASSI – SCHNEIDERMAN

Conocido como diagrama N-S, se considera como otra forma de representar algoritmos. Se basa en escribir las instrucciones en bloques o cuadros de texto. Este diagrama también se conoce con el nombre de diagrama de Chapin y es una combinación de la escritura en pseudocódigo y del diagrama de flujo.

En el diagrama N-S, las flechas que indican el flujo del algoritmo, se reemplazan por las cajas de texto, en las cuales se escriben las instrucciones respectivas. La sintaxis general se indica a continuación:

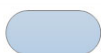

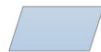




Como se puede apreciar, cada bloque contiene una instrucción, en el caso de definición de variables, se recomienda utilizar un bloque por cada tipo de datos distinto que se genere, es decir si existen varios datos enteros, su definición abarcaría un solo bloque, si por el contrario se requieren definir datos dobles, ellos ocuparían otro bloque y así sucesivamente.

2. Diagrama de flujo

El diagrama de flujo es una forma de representar una serie de instrucciones, con símbolos estandarizados por el Instituto Nacional Americano de Estándares, ANSI (por sus siglas en inglés), mediante los cuales se representan de manera gráfica un algoritmo y permiten visualizar el flujo de datos cuando se procesan y hacia donde se dirigen dichas salidas de cada proceso para ingresar en el siguiente, hasta que termine el algoritmo.

Los símbolos que se emplean en un diagrama de flujo se definen en la figura que se presenta a continuación:

Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 4 Versión: 01
---	---	-------------------------

3. Pseudocódigo

Es la herramienta de programación para la representación no gráfica de algoritmos donde las instrucciones se escriben en palabras similares al lenguaje natural, facilitando así la escritura como la lectura de programas. Aunque no existen reglas para la escritura de un pseudocódigo en español hay una serie de notación estándar que se utilizarán durante el curso y que ya está muy empleada por algunos libros.

3.1. Sintaxis General de un Pseudocódigo

Algoritmo_<nombre>

Inicio

Variables.

Entero : <lista de variables enteras>

Real : <lista de variables enteras>

Cadena : <lista de variables enteras>

Carácter : <lista de variables enteras>

Lógica : <lista de variables enteras>

} Diferentes estructuras

Fin

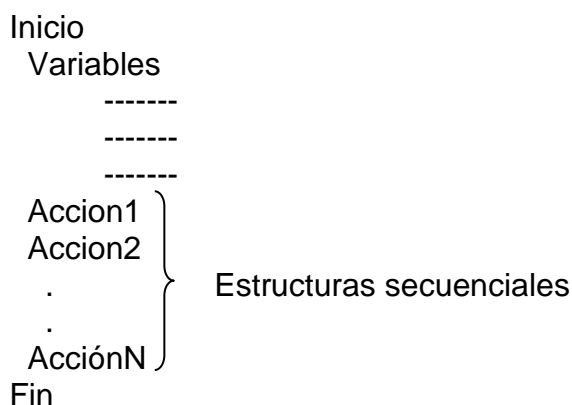
- Donde <nombre> corresponde al nombre que se le asigna a la solución (lo escoge el usuario y debe cumplir con las reglas para identificadores).
- En variables solo se enumeran los tipos de variables que hayan resultado en el análisis. Las variables, como se verá más adelante, también se pueden definir antes del inicio, pero en casos muy específicos.
- Entre inicio y fin irán las diferentes estructuras que se van trabajando, las cuales pueden ser:
 - Secuenciales
 - Selectivas o de decisión
 - Cíclicas o repetitivas

Nota. Obsérvese que en la estructura de un pseudocódigo se debe conservar una adecuada indentación, que indica la dependencia de una instrucción o un bloque de instrucciones con relación a la anterior.

Indentación. Este término significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores, para así separarlo del margen izquierdo y distinguirlo mejor del texto adyacente.

3.2. Estructuras Secuenciales

La estructura secuencial es aquella en la que una acción (instrucción) sigue a otra en secuencia. Las tareas se suceden de tal modo que la salida de una es la entrada de la siguiente y así sucesivamente hasta el fin del proceso. Una estructura secuencial se representa de la siguiente forma:



Las estructuras secuenciales se dividen en tres:

- **Lectura:** La lectura consiste en recibir desde un dispositivo de entrada (p.ej. el teclado) un valor. Esta operación se representa en un Pseudocódigo como sigue:

```

Leer (A)
Leer (B)
  
```

Donde “A” y “B” son variables previamente declaradas y que almacenarán los valores ingresados por el usuario.

- **Asignación:** La asignación consiste, en el paso de valores o resultados a una zona de la memoria. Dicha zona será reconocida con el nombre de variable que recibe el valor. La asignación se puede representar de la siguiente forma: \leftarrow
 - **Simples:** Consiste en pasar un valor constante a una variable ($a \leftarrow 15$)
 - **Contador:** Consiste en usarla como un verificador del número de veces que se realiza un proceso ($a \leftarrow a + 1$)
 - **Acumulador:** Consiste en usarla como un sumador en un proceso ($a \leftarrow a + b$)
 - **De trabajo:** Donde puede recibir el resultado de una operación matemática que involucre muchas variables ($a \leftarrow c + b * 2 / 4$).

	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 4 Versión: 01
---	---	-------------------------

- **Escritura:** Consiste en mandar por un dispositivo de salida (por ejemplo. monitor o impresora) un resultado o mensaje. Este proceso se representa en un Pseudocódigo como sigue:

Escribir (“El resultado es:”, R)

Donde “**El resultado es:**” corresponde a un mensaje que se desea que aparezca y R es una variable que contiene un valor previamente ingresado o asignado.

Ejemplo

Calcular la nota definitiva a partir de tres notas parciales que tienen igual porcentaje.

Asumiendo que ya se tiene previamente el análisis el diseño sería el siguiente:

Algoritmo_Nota

Inicio

Var

Real : Nota_1, Nota_2, Nota_3, Def

Escribir (“Ingrese la Nota uno: “)

Leer (Nota_1)

Escribir (“Ingrese la Nota dos: “)

Leer (Nota_2)

Escribir (“Ingrese la Nota tres: “)

Leer (Nota_3)

Def $\leftarrow (Nota_1 + Nota_2 + Nota_3) / 3$

Escribir (“La nota definitiva es: “, Def)


Fin

Problemas Secuenciales: Para los siguientes ejercicios realice el análisis y diseño de la solución.

1) Suponga que un individuo desea invertir su capital en un banco y desea saber cuánto dinero ganara después de un mes si el banco paga a razón de 2% mensual.

2) Un vendedor recibe un sueldo base más un 10% extra por comisión de sus ventas, el vendedor desea saber cuánto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

3) Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.

	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 4
		Versión: 01

4) Un alumno desea saber cuál será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:

55% del promedio de sus tres calificaciones parciales.

30% de la calificación del examen final.

15% de la calificación de un trabajo final.

5) Un maestro desea saber qué porcentaje de hombres y qué porcentaje de mujeres hay en un grupo de estudiantes.

6) Realizar un algoritmo que calcule la edad de una persona.

ESTRUCTURAS SELECTIVAS SIMPLES - FUNDAMENTOS

TEMAS

- ✓ Estructuras Selectivas Simples.
- ✓ Bloques de instrucciones.
- ✓ Control del flujo del programa.
- ✓ Expresiones lógicas y condiciones.

Las estructuras selectivas simples permiten la toma de decisiones dentro de un programa. Dentro de una estructura selectiva, se incluye una expresión lógica (que devuelve un valor de verdad) que será la condición que se evalúa para definir la ruta que se seguirá dentro del programa. Si la expresión lógica (condición) es verdadera, entonces, se ejecutarán todas las sentencias dentro del bloque de la estructura selectiva simple. Si la expresión lógica es falsa, no se ejecutará ninguna de estas sentencias.

SINTAXIS

```
Si condición entonces
    instrucción_A
    instrucción_B
    . . .
    Instrucción_Z
fin_si
```

EXPLICACIÓN

Se utilizan las palabras reservadas **si**, **entonces**, **fin_si**

La condición debe ser una expresión lógica válida, que al evaluarse produzca un valor de verdad, es decir, será verdadera o falsa, no existen otras posibilidades.

Instrucción_A, instrucción_B, hasta la instrucción_Z constituyen el bloque de instrucciones que se ejecutarán si el resultado de la condición es verdadero.

Como se vio en la explicación, esta estructura permite definir si las instrucciones **instrucción_A**, **instrucción_B**,... **instrucción_Z** se ejecutarán o no; dependiendo del resultado de la condición. Si el resultado de evaluar **condición** es verdadero, se ejecutarán todas las instrucciones que forman parte del bloque **entonces** de la estructura. Cuando al evaluar **condición** se obtiene un resultado falso, estas instrucciones no se ejecutarán y se pasará a la siguiente línea del algoritmo después de **fin_si**.

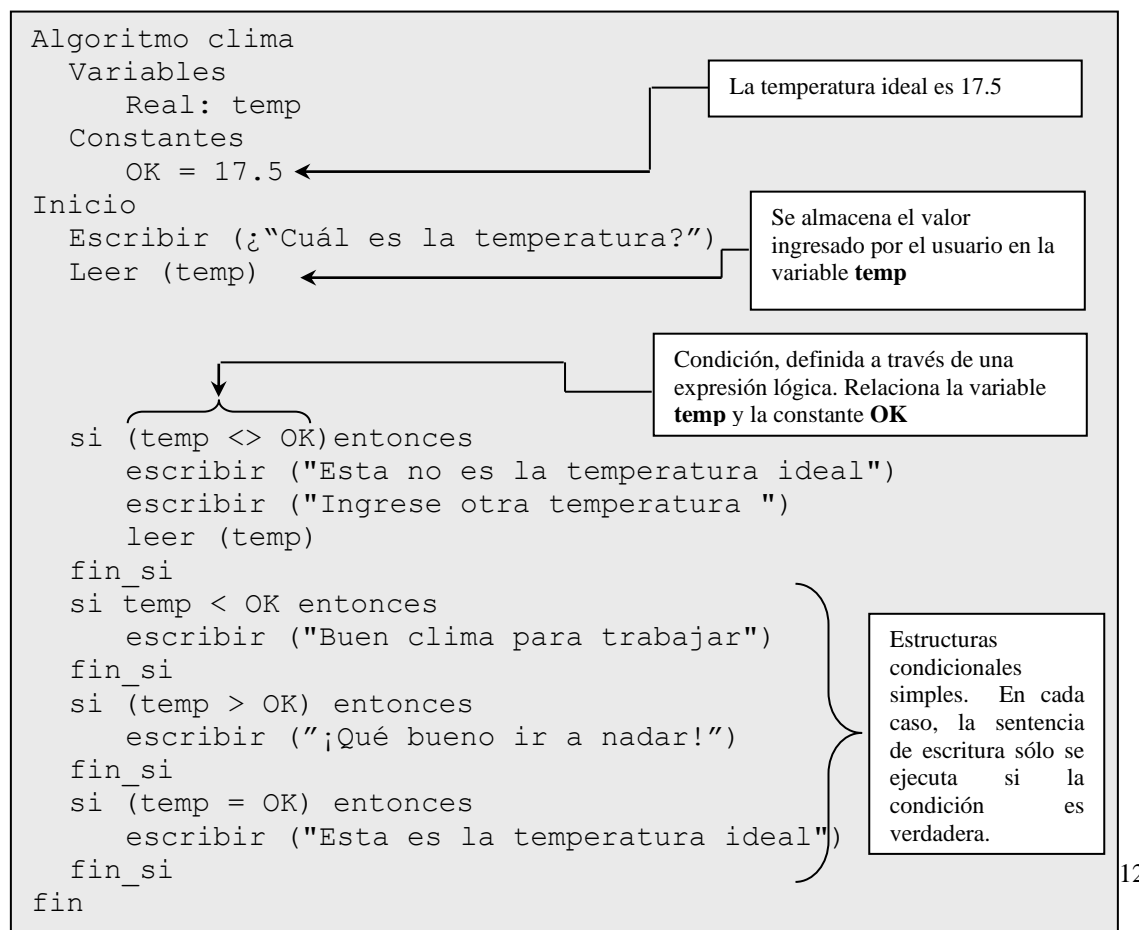
Ejemplo:

En el siguiente algoritmo, se pregunta al usuario una temperatura, y con base en su respuesta y criterios previamente establecidos, se le dan algunas indicaciones sobre lo que debe hacer.

A diferencia de los algoritmos en los que sólo se usan estructuras secuenciales, en este algoritmo algunas sentencias se ejecutan y otras no, dependiendo de los datos de entrada.

El propósito de este algoritmo es mostrar las estructuras selectivas simples y los operadores relacionales usados comúnmente.

ALGORITMO



	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 4 Versión: 01
---	---	-------------------------

Trate de seguir el algoritmo y definir qué sentencias se ejecutan y qué sentencias no se ejecutan.

SALIDA DEL ALGORITMO

¿Cuál es la temperatura?	
Entrada del Teclado:	13
	Esta no es la temperatura ideal. Ingrese otra temperatura
Entrada del Teclado:	26
	Qué bueno ir a nadar !

Descripción

Expresiones Lógicas. La expresión (temp <> OK)

Es una expresión que compara los valores de dos datos aritméticos, en éste caso, se comparan la variable temp y la constante OK, específicamente se trata de establecer si tienen valores diferentes.

Las expresiones lógicas se utilizan para establecer condiciones. Este tipo de expresiones, al evaluarse devuelven un valor de verdad, es decir, sólo puede obtenerse como resultado verdadero o falso.

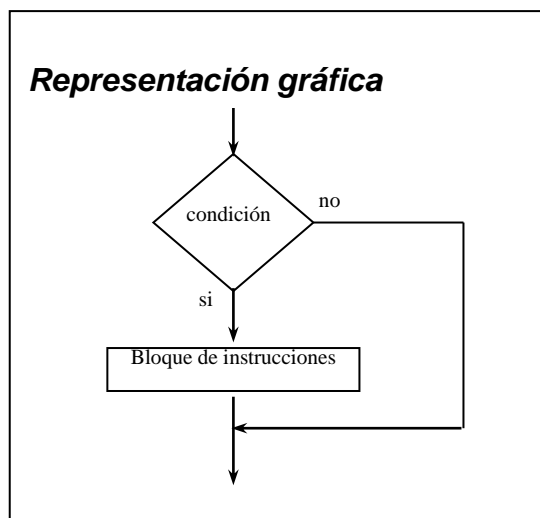
La sintaxis general de una expresión lógica es la siguiente:

operando1 operador_relacional operando2

Donde, tanto operando1 como operando2 pueden ser variables, como temp; constantes como OK; valores literales, como 30, ó cualquier expresión válida del mismo tipo del otro operando; es decir, tanto operando1 como operando2 deben ser del mismo tipo.

PREGUNTAS ORIENTADORAS

1. ¿Qué significan los diferentes operadores relacionales?
2. ¿Para qué pueden utilizarse los operadores lógicos?
3. Identifique qué bloques de instrucciones se ejecutarían si la temperatura ingresada por el usuario no es la temperatura ideal.
4. ¿Si el usuario escribe una temperatura (temp) igual a la temperatura ideal (OK), qué sentencias se ejecutarán?



Dentro de la representación gráfica (Diagrama de Flujo), las estructuras de selección se representan mediante un rombo dentro del que se escribe la expresión lógica que establece la condición. Las palabras si y no, señalan el camino que se seguirá en caso de que la condición si se cumpla o que la condición no se cumpla, respectivamente.

Ejemplo

Un hombre desea saber el nuevo saldo de su CDT en un banco, teniendo en cuenta que el capital depositado en el CDT generará un interés dependiendo del porcentaje que determine el banco, si los intereses generados son superiores a \$7.000 se reanuda el CDT de lo contrario, éste se cancela.

MODELAMIENTO

Datos de Entrada

Descripción.	Identificador
Capital depositado	cap
Porcentaje de interés	p_int

Datos de Salida

Descripción.	Identificador
Nuevo saldo del CDT	nuevo-s

Datos Auxiliares

Descripción.	Identificador
Interés generado por el CDT	Interes

ESPECIFICACIONES

Precondiciones

cap ∈ { reales ≥ 0 }
 p_int ∈ { reales ≥ 0 }
 int ∈ { reales ≥ 0 }

Postcondiciones

Nuevo_s $\in \{ \text{reales} \geq 0 \}$

Formulas

$\text{int} \leftarrow (\text{cap} * \text{p_int}) / 100$
 $\text{nuevo_s} \leftarrow \text{cap} + \text{int}$ cuando $\text{int} \text{ sea} > 7000$

Algoritmo_Interes

Variables

Real : p_int, cap, int, Nuevo-s

Inicio

Leer p_int, cap

$\text{int} \leftarrow ((\text{cap} * \text{p_int}) / 100)$

si ($\text{int} > 7000$) entonces

$\text{nuevo_s} \leftarrow (\text{cap} + \text{int})$

Imprimir (nuevo_s)

fin-si

fin

ESTRUCTURAS SELECTIVAS COMPUESTAS - FUNDAMENTOS

Otra forma de la estructura selectiva, es la estructura selectiva compuesta, en la que se incluye una cláusula opcional si_no para establecer un grupo de acciones que se ejecutarán en caso de que la condición sea falsa. Estas estructuras se conocen también como Estructuras Selectivas Dobles, ya que permiten definir entre dos grupos de instrucciones diferentes dependiendo del resultado de evaluar la condición.

SINTAXIS

Si condición entonces

InstruccionA

InstruccionB

...

InstruccionZ

si_no

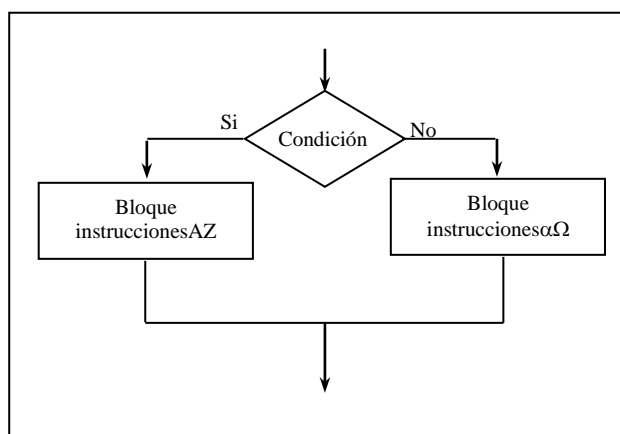
instruccion1

instruccion2

...

InstruccionN

DIAGRAMA DE FLUJO



	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 4
		Versión: 01

EXPLICACION

Se utilizan las palabras reservadas **si**, **entonces**, **si_no**, **fin_si**

La condición debe ser una expresión lógica válida que al evaluarse produzca un valor de verdad, es decir, será verdadera o falsa, no existen otras posibilidades.

InstruccionA, instruccionB,... instruccionZ constituyen el bloque de instrucciones que se ejecutarán si condición es verdadera.

La cláusula **si_no**, indica cuáles serán las instrucciones que se ejecutarán cuando condición es falsa.

Instruccion1, instruccion2... instrucciónN constituyen el bloque de instrucciones asociadas a la cláusula **si_no**.

Al utilizar este tipo de estructuras sólo se ejecutarán el grupo de instrucciones del entonces ó el grupo de instrucciones del si_no, nunca se ejecutarán ambos grupos simultáneamente.

Cuando condición es verdadera, se ejecutarán solamente las instrucciones del **entonces** y luego al flujo del programa pasa a la siguiente línea después del **fin_si**.

Por el contrario, cuando condición es falsa, se ejecutarán el grupo de instrucciones asociadas al **si_no** y posteriormente se continuará con las instrucciones después del **fin_si**

Nota: Tanto el pseudocódigo como el diagrama de flujo son formas de representación de instrucciones algorítmicas. Cualquier algoritmo debe poder expresarse utilizando cualquiera de estas notaciones.

EJEMPLO

El siguiente algoritmo muestra un mensaje indicando si una empresa es rentable o no lo es, de acuerdo con sus ingresos y egresos.

 <p>POLITÉCNICO COLOMBIANO JAIME GARZA CARDONA</p>	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 4 Versión: 01
---	---	-------------------------

```

Algoritmo rentable
  Variables
    Real: ingresos, egresos

Inicio
  Escribir ("A cuando ascienden los ingresos de la empresa?")
  Leer (ingresos)
  Escribir ("A cuando ascienden los egresos de la empresa ?")
  Leer (egresos)
  Balance ← ingresos - egresos

  Si balance > 0 entonces
    Escribir ("Esta empresa es rentable" )
    Escribir ("La empresa tuvo ganancias de $ ", balance)
  si_no
    Balance ← (-1) * balance
    Escribir ("Esta empresa no es rentable")
    Escribir ("Se produjeron pérdidas de $ ", balance)
  fin_si
Fin

```