



GUÍA DE APRENDIZAJE N° GADSI 01

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

Programa de Formación: ADSI	Código: 228106 Versión: 102			
Nombre del Proyecto: Análisis, diseño e implementación de un sistema de información de acuerdo a las necesidades del cliente	Código: 1332602			
Fase del proyecto: Implementación	Desarrollo o			
Actividad (es) del Proyecto: CONSTRUIR EL SISTEMA QUE CUMPLA CON LOS REQUISITOS DE LA SOLUCIÓN INFORMÁTICA.	Actividad (es) de Aprendizaje: <ul style="list-style-type: none">Informe técnico de diseño: Definición, objetivos, características, estructura	AMBIENTE DE FORMACIÓN - ESCENARIO (Aula, Laboratorio, taller, unidad productiva) y elementos y condiciones de seguridad industrial, salud ocupacional y medio ambiente)	MATERIALES DE FORMACIÓN DEVOLUTIVO (Herramienta - equipo) CONSUMIBLE (unidades empleadas durante el programa)	
Resultados de Aprendizaje: Desarrollar el sistema de información que cumpla con los requerimientos de la solución informática	Competencia: 22050100701- Interpretar el informe técnico de diseño, para determinar el plan de trabajo durante la fase de construcción del software, de acuerdo con las normas y protocolos establecidos en la empresa			



SERVICIO NACIONAL DE APRENDIZAJE SENA

GUÍA DE APRENDIZAJE

SISTEMA INTEGRADO DE GESTIÓN

Proceso Gestión de la Formación Profesional Integral

Procedimiento Ejecución de la Formación Profesional Integral

Versión: 02

Código: GFPI-F-019

Duración de la guía (en horas):

60 horas

2. INTRODUCCIÓN

¿Por Qué Aprender A Programar?

Los estudios sobre el mercado de trabajo muestran que la oferta de ingenieros con conocimientos en programación no está ni siquiera cerca de satisfacer la demanda. Esto es cierto tanto en los Estados Unidos como en el resto del mundo. Las empresas, sin importar su tamaño, necesitan de personas con conocimientos en programación que les ayuden a desarrollar los sistemas de información que les permitan crecer en sus mercados.

Por un lado nuestras economías están en un momento interesante. Después de años de trabajo con computadoras y tecnologías, finalmente estamos llegando al punto donde la tecnología ha permeado casi todo los aspectos de la vida diaria de los seres humanos. Por ejemplo, los bufetes de abogados están utilizando la tecnología para acelerar el proceso de descubrimiento de los casos, los inversores están utilizando la tecnología para encontrar nuevas oportunidades de inversión, y compañías como Google la están utilizando para crear vehículos que se auto-conducen. Pero a medida que la tecnología ayuda a aumentar las posibilidades de automatización en muchas industrias, existe también la posibilidad de supresión de empleos en muchos sectores de la economía.

Las empresas de todo el mundo necesitan más programadores. Las personas que desarrollan buenas habilidades en programación pueden conseguir empleos de hasta US 100,000.00 al año. En la mayoría de los países pasa un fenómeno que los economistas llaman desempleo estructural: los puestos de trabajo están disponibles, pero nuestra fuerza de trabajo no está capacitada para esos trabajos.

3. ESTRUCTURACION DIDACTICA DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial.

Camilo es un joven inquieto al momento de elegir y tomar decisiones en su vida, ahora mismo está estudiando para ser programador, él ha escuchado que a través de la programación puede resolver la cantidad de problemas a los que se enfrenta en su diario vivir, ya que se puede automatizar procesos que tal vez lleva horas, días, meses y ocasiones años al mundo real para llegar a la solución, es por ello que en la clase de Fundamentos de Programación empieza a ver algo que es la "lógica", aprendiendo que ésta en que Toda persona que pretenda construir un programa que dé solución a determinada problemática, se enfrenta a dos grandes tareas:

El **QUÉ**: acciones a realizar para poder resolver el problema. Esta tarea forma parte del trabajo de mesa previo a toda actividad de programación.



El **CÓMO**: instrucciones de las que se va a valer para escribir el código que realice las acciones determinadas en el QUÉ, las cuales están determinadas por el lenguaje de programación seleccionado.

Muchas personas confunden la Programación con la Lógica de Programación, la primera involucra el conocimiento de técnicas e instrucciones de un determinado Lenguaje a través de los cuales se hace sencillo lograr que la Computadora obtenga unos resultados mucho más rápidos que una persona. La segunda involucra, de una manera técnica y organizada, los conceptos que permiten diseñar en términos generales, la solución a problemas que pueden llegar a ser implementados a través de una computadora.

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.)

Teniendo en cuenta el caso de estudio anterior

¿Qué considera Usted es la programación?

¿Qué objetivos establece en realidad la programación?

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

LOS OPERADORES

Son elementos muy útiles al momento de programar, los operadores son símbolos que significan una instrucción específica y se clasifican en Aritméticos, Relacionales, Lógicos y De Asignación

Aritméticos: usados para efectuar operaciones matemáticas.		
Operador	Uso	Ejemplo
$-$	Restar números	Datos: $A=12-3$, Resultado: $A=9$
$+$	Sumar números	Datos: $A=12+3$, Resultado: $A=15$
$*$	Multiplicar de números	Datos: $A=12*3$, Resultado: $A=36$
$/$	Dividir de números	Datos: $A=12/3$, Resultado: $A=4$
\wedge	Elevar a una potencia un número	Datos: $A=12^3$, Resultado: $A=1728$
$--$	Quitar una unidad a un número	Datos: $A=12$; $A=--$ Resultado: $A=11$
$++$	Aumentar una unidad a un número	Datos: $A=12$; $A=++$, Resultado: $A=13$
$\%$	Obtener el módulo (mod) de una división, es decir, el residuo de dicha división	Datos: $A=12$; $B=5$; $A=A\%B$ $A=2$



Relacionales:

se utilizan para comparar dos o más valores y determinar si el resultado es falso (0) o verdadero (1).

Operador	Uso (comparación)	Ejemplo
<	Menor que	Datos: A=12; B=3 Comparación: If(A<B) A+1, Resultado: A=12; B=3
>	Mayor que	Datos: A=12; B=3 Comparación: If(A>B) A+1, Resultado: A=13; B=3
<=	Menor o igual que	Datos: A=12; B=12 Comparación: If(A<=B) B+1, Resultado: A=12; B=13
>=	Mayor o igual que	Datos: A=12; B=3 Comparación: If (A>=B) B+1, Resultado: A=12; B=4
< > ó !=	Diferente a	Datos: A=12; B=3 Comparación: If(A<>B) ó If(A!=B) A+1, Resultado: A=13; B=12
=	Igual que	Datos: A=12; B=12 Comparación: If(A=B) B+1, Resultado: A=12; B=13
+ =	Adiciona el valor de la izquierda al de la derecha	Datos: A=12; B=12 Comparación: A+=B; A=A+B Resultado: A=24



Lógicos:

arrojan un resultado verdadero (1) o falso (0) al comparar uno o más valores numéricos o bits, que pueden estar a su vez vinculados con operadores relacionales.

Operador

Uso

Ejemplo



AND, al comparar valores o expresiones, si ambos son verdaderos obtiene un "true"

Datos: $X=(0+8,1+6,2+3)$; $Y=6+2$
Comparación: $X(n)\&\&Y$,
Si se cumple "Cierto", si no "Falso"
Resultado: cuando,
• $X(1)=0+8$ y $Y=6+2$ se arrojará un "Cierto"
• $X(2)=1+6$ y $Y=6+2$ se arrojará un "Falso"
• $X(3)=2+3$ y $Y=6+2$ se arrojará un "Falso"



OR, al comparar valores o expresiones, si una es verdadera obtiene un "true", si ambas son falsas se obtiene un "false"

Datos: $X=(0+8,1+6,2+3)$; $Y=6+2$; If
Comparación: $\text{If}(X>6) \parallel \text{If}(Y>9)$,
Si se cumple "Cierto", si no "Falso"
Resultado: cuando,
• $X(1)=0+8$ (1) y $Y=6+2$ (0), se arrojará un "Cierto"
• $X(2)=1+6$ (1) y $Y=6+2$ (0) se arrojará un "Cierto"
• $X(3)=2+3$ (0) y $Y=6+2$ (0) se arrojará un "Falso"



NOT, al comparar valores o expresiones, niega la respuesta o resultado obtenido.

Datos: $X=(0+8,1+6,2+3)$; $Y=6+2$; If
Comparación: $\text{If}(X>6) \neq \text{If}(Y>9)$,
Si se cumple "Cierto", si no "Falso"
Resultado: cuando,
• $X(1)=0+8$ (1) y $Y=6+2$ (0), se arrojará un "Falso"
• $X(2)=1+6$ (1) y $Y=6+2$ (0) se arrojará un "Falso"
• $X(3)=2+3$ (0) y $Y=6+2$ (0) se arrojará un "Cierto"



Asignación

permiten atribuir un valor a una variable.

Operador

Uso

Ejemplo

=

Igualdad de valor

Datos: A=1; B=1
Expresión: A=B,
Resultado: B=1

+ =

Refiere a la suma del valor de la Izquierda más el de la derecha

Datos: A=13; B=1,
Expresión: A+=B
Resultado: A=13+1

- =

Refiere a la resta del valor de la Izquierda menos el de la derecha

Datos: A=13; B=1,
Expresión: A-=B
Resultado: A=13-1

* =

Refiere a la multiplicación del valor de la Izquierda por el de la derecha

Datos: A=13; B=1,
Expresión: A*=B
Resultado: A=13*1

/ =

Refiere a la división del valor de la Izquierda entre el de la derecha

Datos: A=13; B=1,
Expresión: A/=B
Resultado: A=13/1

^ =

Refiere a elevar el valor de la Izquierda a la potencia del de la derecha

Datos: A=13; B=1,
Expresión: A^=B
Resultado: A=13^1



TIPOS DE VARIABLES



Para realizar un programa, es necesario indicarle a la máquina el tipo de dato que almacena cada variable que se usará. De esta manera, la máquina destinará un espacio específico en la memoria de acuerdo al número de bits que ocupe. De ahí que se asignaran las siguientes variables básicas:

Variable	Tipo Variable	Uso	Bits	Ejemplo
<i>Int</i>	Numérico	Usada para hacer referencia a valores numéricos enteros.	16	<i>int</i> Edad; La variable edad puede tomar cualquier valor entero.
<i>Float</i>		Utilizada para valores numéricos con decimales.	32	<i>float</i> Promedio; La variable promedio puede tomar valores con punto decimal.
<i>Double</i>		Colocada cuando se quiere referir a una variable numérica decimal de valores muy grandes o muy pequeños.	64	<i>double</i> Pi; La variable Pi puede tomar valores con punto decimal muy largos.
<i>Char</i>	Caracteres	Empleada cuando una variable almacena letras o caracteres alfabéticos.	8	<i>char</i> Nombre; La variable nombre almacenará las letras de los nombres que le sean ingresados.
<i>Bool</i>	Booleano	Introduce valores 1("true") y 0("false").	1	<i>bool</i> rojo = <i>true</i> Mientras la variable sea roja será verdadera (1) cualquier otro color la máquina lo reconocerá como falso (0).



SERVICIO NACIONAL DE APRENDIZAJE SENA

GUÍA DE APRENDIZAJE

SISTEMA INTEGRADO DE GESTIÓN

Proceso Gestión de la Formación Profesional Integral

Procedimiento Ejecución de la Formación Profesional Integral

Versión: 02

Código: GFPI-F-019

Calificador del tipo de variable	Variable	Uso	Rango de datos
<i>Signed</i>	<i>Int</i>	Se usa cuando se pueden presentar valores numéricos negativos o positivos de forma indistinta.	-32768 a 32767
	<i>Float</i>		
	<i>Double</i>		
	<i>Char</i>		-128 a 127
	<i>Bool</i>		-1 a 0 ó 0 a 1
<i>Unsigned</i>	<i>Int</i>	Se usa cuando sólo nos importa conocer valores numéricos positivos.	0 a 65535
	<i>Float</i>		3.4E-38 a .4E+38
	<i>Double</i>		1.7E-308 a 1.7E+308
	<i>Char</i>		0 a 255
	<i>Bool</i>		0 a 1
<i>Long</i>	<i>Int</i>	Permite aumentar el número de datos en operaciones con cifras muy grandes.	-2147483648 a 2147483647
	<i>Double</i>		1.7E-308 a 1.7E+308 ó 3.4E-4932 a 1.1E+4932
<i>Short</i>	<i>Int</i>	Permite trabajar con datos en operaciones con cifras muy cortas, negativas y positivas.	-32768 a 32767
	<i>Unsigned Short Int</i>	Permite trabajar con datos en operaciones con cifras muy cortas positivas.	0 a 65535



Al declarar alguna variable, sin especificar si se trata de un *signed* o *unsigned*, la máquina automáticamente lo asocia a una variable *signed*, lo que podría causar errores posteriores en la lógica de programación.



Otros tipos de variable	Variable	Uso	Ejemplo
<i>Arrays</i>	Matrices	Sirven para almacenar diversos elementos de un mismo tipo, se debe especificar el número de elementos entre corchetes al declararse; para que se puedan realizar operaciones con éstos. Cuando se llame a una operación un elemento de la matriz primero se citará a la fila y después la columna.	<pre>int mat[3][3]={3,1,8}, {7,4,5}, {6,9,2}; mat[2][3]=5 Int vec[5]={10,20,30,40,50}; vec[0]=10 vec[1]=20 vec[2]=30 vec[3]=40 vec[4]=50</pre>
<i>Strings</i>	Cadenas	Son cadenas caracteres o de elementos char, donde su último carácter debe agregarse manualmente o bien dejar el espacio considerado y es nulo "\0".	<pre>char str[5]="Nano", donde al str[0]="N"; str[1]="a"; str[2]="n"; str[3]="o"; str[4]="\0"</pre>



Ten presente que para cada lenguaje de programación es necesario que consultes los tipos de datos que aplican, ya que no son los mismos.





Resolución de problemas con algoritmos

Para solucionar un problema mediante un algoritmo es necesario seguir un orden, uno de los primeros pasos es el diseño previo de un algoritmo, con la resolución de problemas se puede seguir este orden de la siguiente manera:

- a. **Analizar el problema:** es analizar la situación que se está presentando y organizar en un orden lógico cada uno de los pasos para así resolver el problema.
- b. **Diseñar el algoritmo:** en este paso se describe la secuencia ordenada de pasos que conduce a la solución del problema citado (diagrama de flujo o pseudocódigo).
- c. **Expresar el algoritmo:** el algoritmo se debe expresar como un programa en un lenguaje de programación adecuado. (Fase de codificación.)
- d. **Ejecución y validación:** se pone en ejecución el programa realizado en un computador.

ANALIZAR EL PROBLEMA

DISEÑAR EL ALGORITMO

EXPRESAR EL ALGORITMO

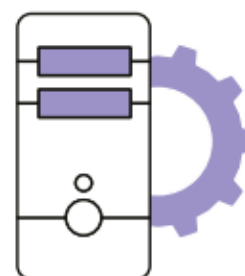
EJECUCIÓN Y EVALUACIÓN

*Pasos para resolver algoritmos.



Análisis del problema

- **Reformular el problema:** si el problema que se ha planteado no se encuentra bien formulado, redactado o no se entiende al momento de leerse, debe reformularlo de tal manera que usted lo entienda y tenga claro la situación problema a solucionar.
- **Resultados Esperados:** debe especificar, describir y/o escribir los resultados que espera, por ejemplo (cuál es el producto final que se quiere tener para dar solución al problema, cual es la información a la que se necesita llegar, que se espera del problema citado)
- **Datos disponibles:** identificar la información disponible se resuelve haciéndose las siguientes preguntas:
¿qué información es importante o relevante para solucionar el problema?
¿cuáles son los datos de entrada?,
¿cuál es la incógnita?, ¿qué información me falta para resolver el problema?.
- **Restricciones:** determina las condiciones que plantea el problema para lograr el resultado, lo que está permitido, lo prohibido.
- **Procesos necesarios:** en esta fase debe definir los procesos para poder convertir la información disponible, en resultados esperados que den solución al problema ya que se determinan los procesos que se necesitan, las formulas a utilizar y el orden de lo que se debe realizar.





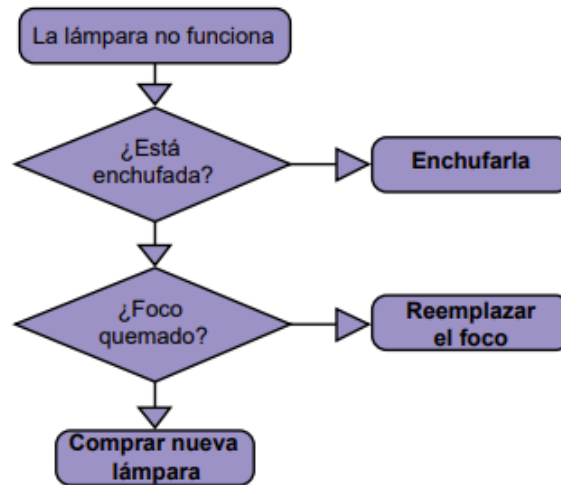
Análisis de Ejercicios

Ejemplo 1. La lámpara

Fuente: <https://goo.gl/images/pHAOQY>

Como se puede observar en el ejercicio anterior existe un análisis para determinar si la lámpara funciona o no funciona dependiendo la situación y así tomar una decisión con respecto al problema presentado.

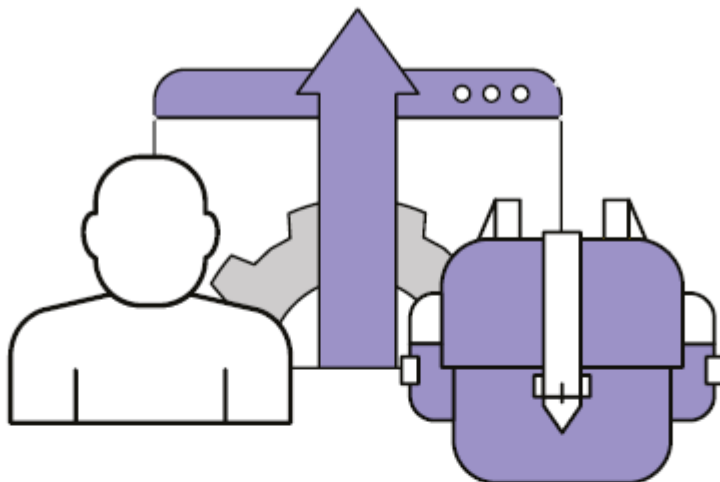
En esto es lo que básicamente consiste un algoritmo, en determinar un orden lógico y una descripción breve para lograr tomar una decisión o lograr un objetivo.



*Simbología del algortimo de la lámpara.



Ejemplo 2



José David, es un muchacho que desea comprar una maleta de \$105.000. El recibe ingresos de diferentes fuentes: en la casa le dan \$6.000 para sus gastos semanales durante 4 semanas, por atender una tienda, tres veces recibió \$12.000. También su hermano lava la piscina una vez al mes por \$13.000 y cuida la tienda por \$10.000. ¿José David tiene ahorrado el dinero suficiente para comprar la patineta o aún le falta?

- **Formular el problema:** ya se encuentra claramente planteado, pero si no entiende la redacción puede redactarlo a su manera, hasta es posible realizar un resumen de lo planteado.
- **Resultados esperados:** saber si José David tiene o no tiene ahorrado el dinero para comprar su maleta, la cual cuesta \$105.000 pesos.
- **Datos disponibles:** los ingresos de José David \$6.000 pesos por 4 semanas + 12.000 pesos por 3, los datos irrelevantes serían: los \$13.000 y \$10.000 pesos que ganó el hermano ya que no aportan información para la solución de este problema y se pueden omitir.
- **Restricciones:** no se encuentra ninguna.
- **Procesos:** calcular el valor ahorrado por José David para saber si le alcanza para comprar la patineta.
- $\text{Valor Ahorrado} = 24.000 + 36.000 = 60.000$ Es decir no le alcanza para la maleta.



Ejemplo 3

Se necesita calcular el área de un triángulo rectángulo cuya Base mide 3 cm, la Altura 4 cm y la Hipotenusa 5 cm.

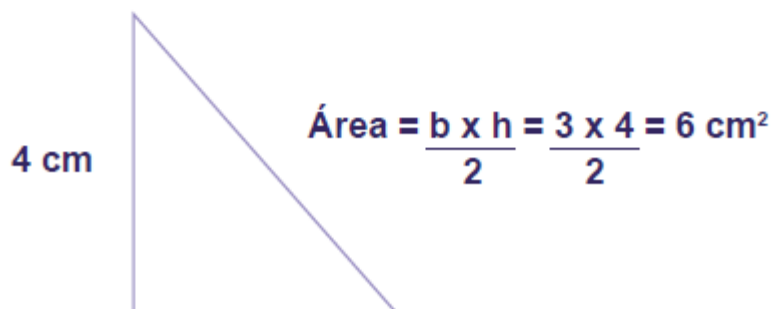
- Formular el problema: ya se encuentra claramente planteado, pero si no entiende la redacción puede redactarlo a su manera, hasta es posible realizar un resumen de lo planteado.
- Resultados esperados: el área de un triángulo rectángulo.
- Datos disponibles: Base, Altura, Hipotenusa, tipo de triángulo.

La incógnita es el área y todos los valores son constantes. El valor de la hipotenusa se puede omitir.

El aprendiz debe preguntarse si sus conocimientos actuales de matemáticas le permiten resolver este problema; de no ser así, debe plantear una estrategia para obtener los conocimientos requeridos.

Determinar las restricciones: utilizar las medidas dadas, y saber que se debe aplicar la fórmula del área de un triángulo rectángulo.

- Procesos necesarios: guardar en dos variables los valores de Base y Altura; Guardar en una constante el divisor 2; aplicar la fórmula $\text{área} = \text{base} \times \text{altura} / 2$; comunicar el resultado (área).





Diseño de un algoritmo.

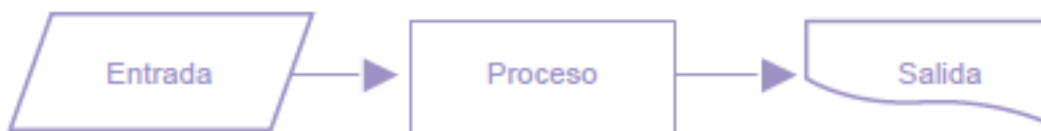
El diseño de un algoritmo se puede realizar mediante un diagrama de flujo o mediante pseudocódigo. Los algoritmos tienen las siguientes características.

Características de los algoritmos

Un algoritmo debe:

- Ser preciso e indicar el orden de realización de cada paso.
- Estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Ser finito. Si se sigue un algoritmo, se debe terminar en algún momento.

Un algoritmo debe contener como mínimo las siguientes partes:



*Partes de un algoritmo.

Por ejemplo para realizar una receta de comida por medio de un algoritmo, cada parte podrá estar determinada así:

Entrada: insumos y elementos de trabajo, cocineros.

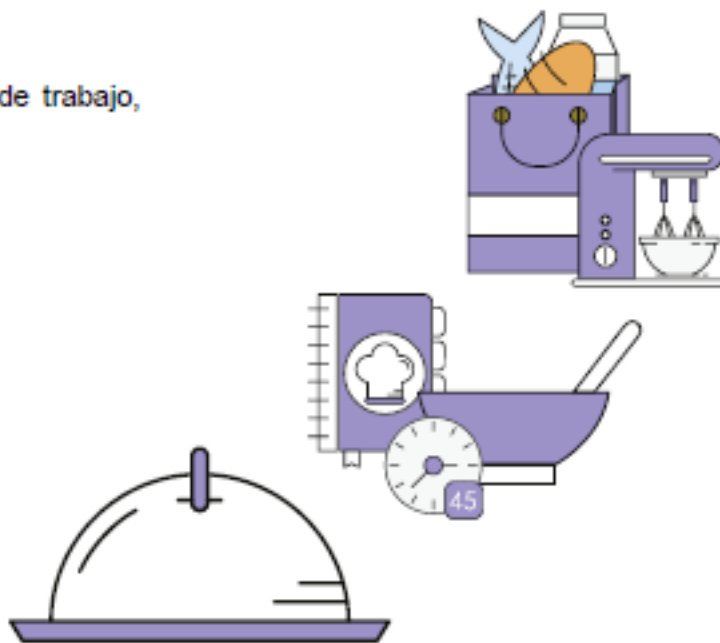
Entrada de un algoritmo.

Proceso: elaboración del plato

Proceso Algorítmico

Salida: preparación finalizada.

Salida del producto.





Diseño de un algoritmo mediante diagrama de flujo

Los algoritmos pueden representarse de varias maneras, entre ellas está la representación de diagramas de flujo. La representación en diagramas de flujo tiene como objetivo seguir paso a paso la solución de un problema mediante símbolos.

Definición Diagrama de flujo.

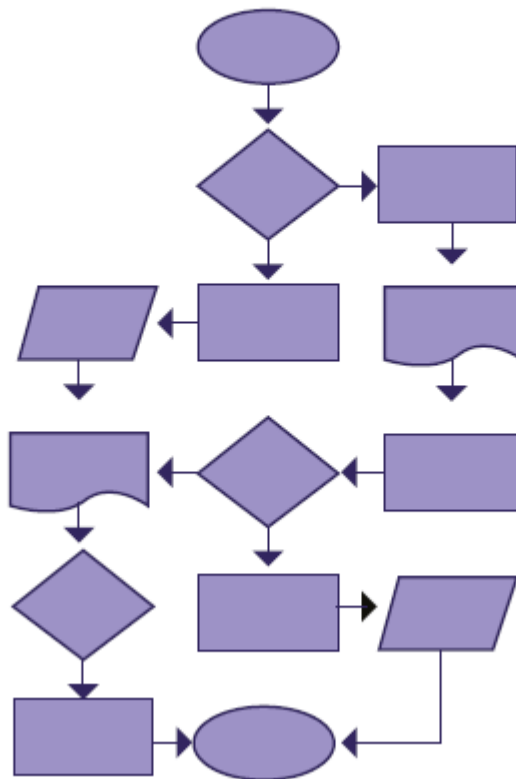
Un diagrama de flujo es un conjunto secuencial de figuras geométricas estándar conectadas lógicamente entre sí para dar solución a un problema específico, cada figura tiene un significado propio.

La secuencia lógica se da por medio de flechas llamadas líneas de flujo que indican el flujo lógico del algoritmo. Al ser un diagrama gráfico facilita la visión de la ejecución del algoritmo.

La simbología utilizada en estos diagramas ha sido estandarizada por las organizaciones ANSI (American National Institute) y por ISO (International Standard Organization).

SÍMBOLO	SIGNIFICADO	EXPLICACIÓN
	Paso de tipo operación	Representa cualquier tarea del proceso que lleve implícita una acción física o intelectual (excepto las de inspección o almacenaje).
	Paso de inspección	Se corresponde con tareas de verificación del trabajo realizado en determinada actividad del proceso. Sus acciones más comunes son: clasificar, observar, supervisar, auditar, probar, revisar, verificar, entre otras.
	Paso de decisión	Representa cualquier punto de decisión. Siempre tendrá al menos dos salidas.
	Paso de almacenaje	Se corresponde con una etapa del proceso que sitúa un producto, información o servicio en una zona de conservación (archivo, almacén o refrigerador) o posición (cola) para utilizarlo o proporcionar el servicio más adelante.
	Paso de demora	Corresponde a actividades que implican un retraso o pausa en el flujo del proceso.
	Línea de flujo	Muestra la dirección y sentido del flujo del proceso y representa el progreso de los pasos en la secuencia.
	Documento	Se utiliza con el objetivo de especificar los documentos confeccionados, corregidos o consultados en cada etapa.
	Conector de tareas	Se utiliza el caso de que el diagrama no se pueda hacer en una sola hoja.

Existen herramientas software que permiten realizar los gráficos mencionados anteriormente, una de las más utilizadas es DFD, día, entre otros, los cuales son útiles y de libre acceso en la web.



Reglas para la elaboración de un diagrama de flujo:

- Los diagramas se deben realizar de arriba hacia abajo y de izquierda hacia derecha.
- Los símbolos de inicio y final deben aparecer solo una vez
- La ejecución de un programa siempre empieza en la parte superior del programa.
- La dirección del flujo se debe representar por medio de flechas.
- Todas las líneas de flujo deben llegar a un símbolo o a otra línea.
- Se deben inicializar las variables que se utilicen o permitir la asignación de valores mediante la consulta a un usuario.



Diseño de un algoritmo mediante pseudocódigo

La representación de pseudocódigo sigue paso a paso la solución de un problema con lenguaje natural, pero recuerde que debe realizar la metodología resolución de problemas para poder iniciar su algoritmo con pseudocódigo. Podría recordarse la imagen de resolución de problemas explicada anteriormente.

Paso 1: inicio

Paso 2: los procesos que se van realizar....

Paso 3: si hay más pasos se debe continuar...

Paso n: fin



Ejemplo1:

Se retoma el ejemplo de José David para llevarlo en forma de algoritmo: José David, se encuentra ahorrando para comprar la maleta que vale 105.000 pesos. En su casa le han dado para sus gastos 24.000 pesos durante 4 semanas. Por atender la tienda recibió \$36.000 pesos. Su hermano Juan Antonio ganó 23.000 pesos por lavar la piscina y cuidar la tienda. ¿José David tiene ahorrado el dinero suficiente para comprar la maleta o aún le falta?

Paso 1: inicio

Paso 2: asignarle el valor del dinero ahorrado por José David a la Variable ValorAhorrado = 24.000+36.000

Paso 3: mostrar el ValorAhorrado, indicando si le alcanza o no para la maleta.

Paso 4: fin.

Ejemplo 2

Se retoma el ejemplo anterior también por continuidad.

Se necesita calcular el área de un triángulo rectángulo cuya Base mide 3 cm, la Altura 4 cm y la Hipotenusa 5 cm.

Paso 1: inicio

Paso 2: indicar que la variable Base tiene un valor de 3 cm

Paso 3: indicar que la variable Altura es de 4 cm

Paso 4: calcular el área $a = (Bases * Altura) / 2$

Paso 5: mostrar el área

Paso 6: fin.



Ejemplo 3

También se pueden resolver problemas cotidianos: por medio de pseudocódigo escriba un algoritmo para poder pasarse los semáforos, una vez usted se encuentra esperando pasar como peatonal.

Paso 1: inicio

Paso 2: ver el color del semáforo

Paso 3: si el semáforo esta en rojo : hay que detenerse

Paso 4: mostrar mensaje hay que detenerse.

Paso 5: si el semáforo esta en amarillo: alistarse para pasar

Paso 6: mostrar mensaje alistarse.

Paso 7: si el semáforo esta en verde: pasarse la calle.

Paso 8: mostrar mensaje puede pasarse y la persona se pasa la calle.

Paso 9: fin.

Variables.

Cuando se inicia con el análisis del problema se identifican los datos iniciales, estos datos se estructuran o se definen como variables; en ellas se pueden almacenar valores y son nombradas con identificadores, es decir nombres para poder identificarlas dentro del algoritmo. Por ejemplo, si en el problema de calcular el área de un triángulo tengo como datos iniciales la base y la altura, estas dos anteriores serian llamadas variables y sus nombres serán "BASE" y "ALTURA".

Un aspecto importante de las variables es que pueden cambiar su valor durante la ejecución del algoritmo.

Se debe tener en cuenta que una variable puede ser declarada, asignada o solicitada mediante un algoritmo para poder ser utilizada.

Declaración de una variable

Para utilizar una variable tanto en pseudocódigo como en diagrama de flujo es necesario siempre declararla, es decir indicarle al algoritmo que va a utilizar una variable por ejemplo Nom_Persona , quiere decir que usted necesita una variable que almacene los nombres de las personas. Ejemplo Nom_Persona=Nombre de las personas.



Asignación de valor a una variable

Luego de declararla puede usted necesitar que esa variable inicie por defecto con un valor por ejemplo Saldo = 60000, quiere decir que utilizará la variable saldo pero que esta inicializada con un valor de 60000, puede inicializar una variable según el valor que necesite para solucionar el problema.

Asignarle un valor a una variable, constante, acumulador o contador, expresiones complejas o simples, por ejemplo.

Variable = expresión

Nom_Persona = Rita

Saldo = 4000

Area = (Base*altura)

Solicitar una variable

Cuando se necesita que el valor de la variable pueda ser ingresado por un usuario al computador, entonces debe solicitarse este valor de la variable.

Constantes

Almacenan datos al igual que las variables, pero su gran particularidad es que sus datos no cambian durante la ejecución del algoritmo, es decir siempre el valor de la constante va ser el mismo.

Las constantes se deben declarar e inicializar.

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO		
<p>Declarar una constante e inicializarla</p> <table border="1"><tr><td>Máximo</td><td>100</td></tr></table>	Máximo	100	<p>Declarar una constante e inicializarla</p> <p>Constante Máximo 100</p>
Máximo	100		

Ejemplo de constante.

Contadores

Como su palabra lo dice permiten contar, para poder utilizar un contador es necesario inicializarlo en un valor y luego incrementar su valor de una manera constante para permitir realizar el conteo.



Es una variable cuyo valor se incrementa o decrementa en una cantidad constante cada vez que se produce un determinado suceso o acción. Los contadores se utilizan con la finalidad de contar sucesos o acciones internas de un bucle; deben realizar una operación de inicialización y posteriormente las sucesivas de incremento o decremento del mismo. La inicialización consiste en asignarle al contador un valor. Se situará antes y fuera del bucle.

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
<p>Declarar una constante e inicializarla</p> <pre>graph TD A[cont 0]</pre>	<p>Declarar una constante e inicializarla</p> <p>cont = 0</p>
<p>Utilizar contador</p> <pre>graph TD A[cont cont]</pre>	<p>Utilizar un contador</p> <p>cont = cont + 1</p>

Representación:

<nombre del contador> = <nombre del contador> + <valor constante>

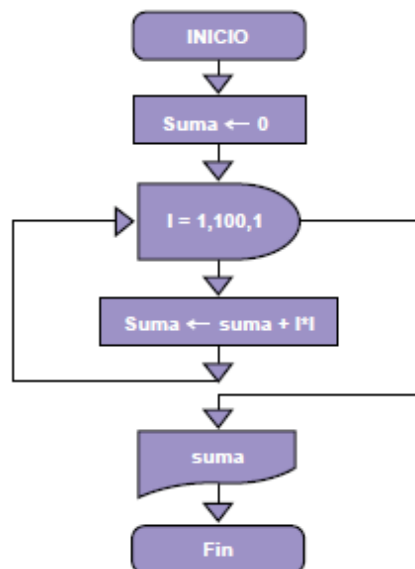
Si en vez de incremento es decremento se coloca un menos en lugar del más.

Ejemplo: $i = i + 1$

Acumuladores

Como su palabra lo dice permiten acumular el valor de una variable, para poder utilizar un acumulador es necesario inicializarlo en un valor y luego iniciar con la acumulación del valor.

El anterior algoritmo está representado mediante un Diagrama de Flujo y lo que hace es inicializar un valor en 1 hasta 100, hasta imprimir la suma de cada dígito acumulándolo hasta 100. En este caso sería $1+2+3+4+5+...+100$.





Identificadores

Los identificadores son nombres que se dan a las variables, constantes, acumuladores y contadores para así poder diferenciarlos. Para asignar los nombres se debe tener en cuenta lo siguiente:

- Los nombres pueden estar formados por una combinación de letras y números (saldoMes, salario, fecha2, baseTriángulo, etc).
- El primer carácter de un nombre debe ser una letra.
- La mayoría de los lenguajes de programación diferencian las mayúsculas de las minúsculas.
- Los nombres deben ser nemotécnicos, con solo leerlos se puede entender lo que contienen. Deben ser muy descriptivos; no utilizar abreviaturas, a menos que se justifique plenamente.
- No utilizar caracteres reservados (% , + , / , > , etc).
- No utilizar palabras reservadas por los lenguajes de programación.
- Para cumplir con convenciones ampliamente utilizadas (Jiménez, 2002), los nombres de procedimientos, variables y constantes deben empezar con minúscula. Ejemplo, fecha, suma, etc. Si es un nombre compuesto por varias palabras, cada una de las palabras (con excepción de la primera) debe empezar con mayúscula. Ejemplo: fechaInicial, baseTriángulo, etc.

Correspondencia de pseudocódigo a diagrama de flujo.

Consiste en llevar un algoritmo a un diagrama de flujo.

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
<pre>graph TD; Inicio([Inicio]) --> DIV[DIV = 2]; DIV --> BASE[/BASE/]; BASE --> ALTURA[/ALTURA/]; ALTURA --> AREA1[ÁREA = BASE * ALTURA]; AREA1 --> AREA2[ÁREA = ÁREA / DIV]; AREA2 --> AREA3{ÁREA}; AREA3 --> Final([Final]);</pre>	<p>Paso 1: inicio.</p> <p>Paso 2: asignar el número 2 a la constante DIV.</p> <p>Paso 3: saber la base del triángulo y guardarlo en la variable "BASE".</p> <p>Paso 4: saber la altura del triángulo y guardarla en la variable "ALTURA".</p> <p>Paso 5: guardar en la variable "AREA" el valor de BASE * ALTURA</p> <p>Paso 6: guardar en la variable "AREA" la división entre el "AREA" sobre la variable "DIV".</p> <p>Paso 7: mostrar el valor de área.</p> <p>Paso 8: fin.</p>



Estructuras de decisión

Son utilizadas para tomar decisiones lógicas, llamadas también estructuras selectivas o alternativas, en ellas se evalúa una condición y en función del resultado de la misma se realiza una opción u otra.

Ejemplo:

DIAGRAMA DE FLUJO	PSEUDOCÓDIGO
	<pre>Inicio Si condición entonces Fin - si Fin condición</pre>

Las estructuras de decisión pueden ser:

Simple.

Ejecuta una determinada acción cuando se cumple una determinada condición (llamada si-entonces). La selección si-entonces evalúa la condición y si la condición es verdadera, entonces ejecuta la acción; si la condición es falsa entonces no hace nada. Ejemplo:

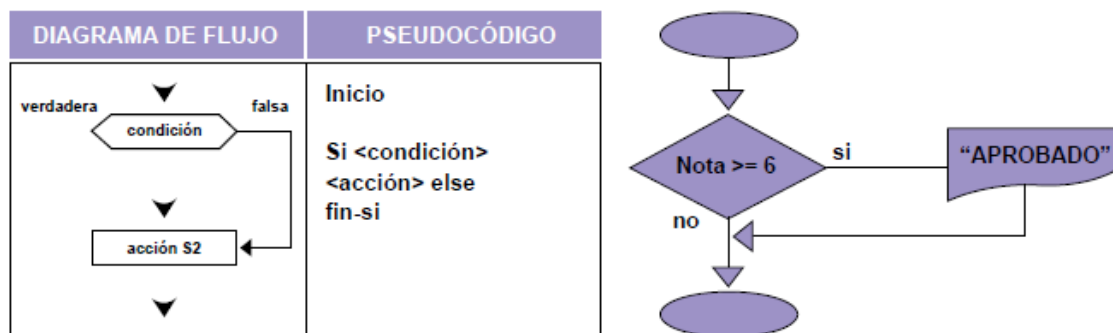


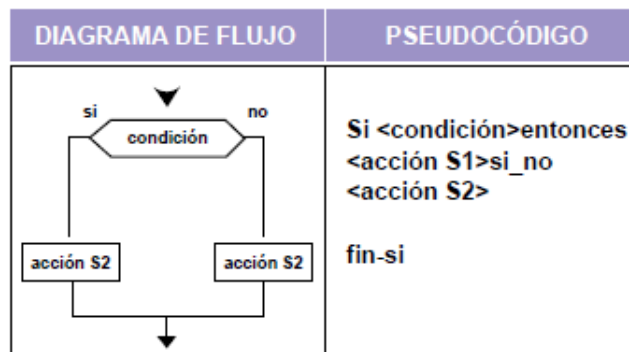
Figura 15. Estructura simple de un condicional.

En el ejemplo anterior existe una estructura de decisión simple dado que si la Nota es superior a 6 es considerado el aprendiz como un logro aprobado.

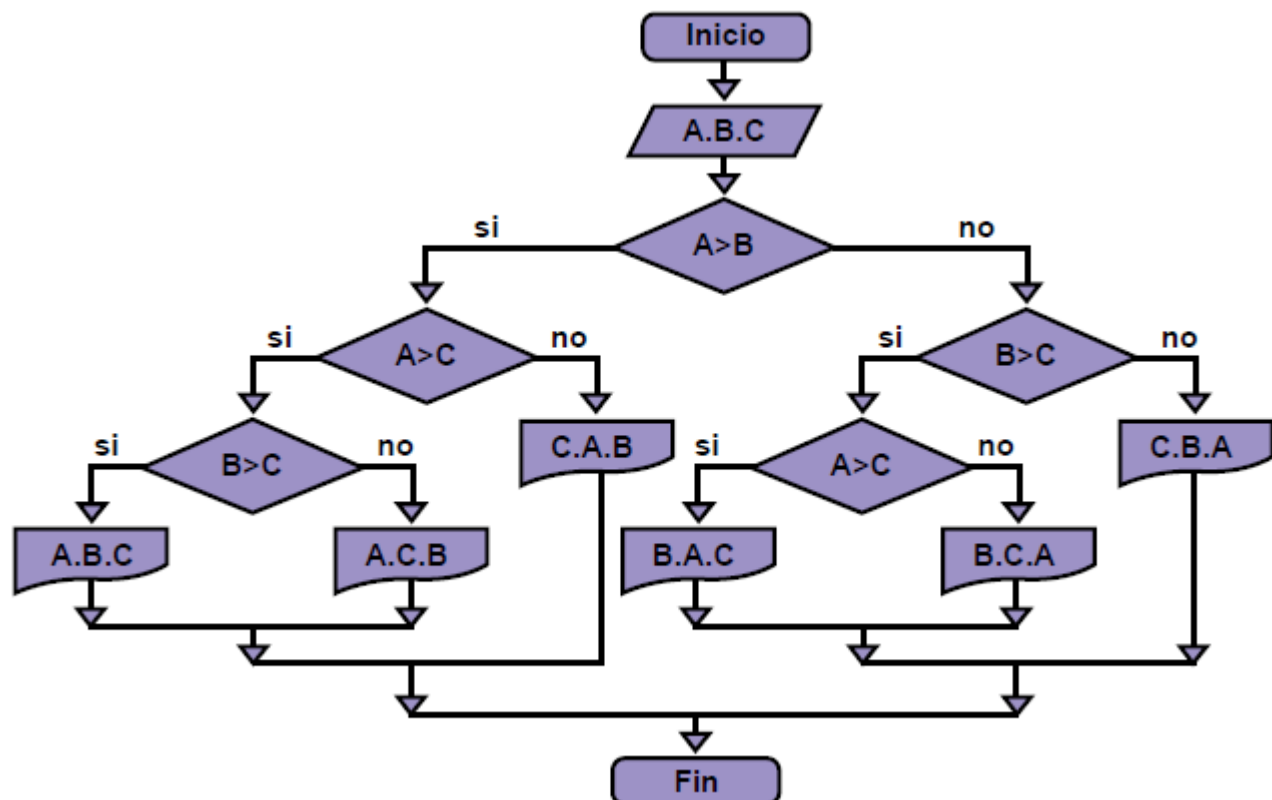


Dobles:

La estructura anterior es muy limitada y normalmente se necesitará una estructura que permita elegir entre dos opciones o alternativas posibles, en función del cumplimiento o no de una determinada condición. **Ejemplo:**



En el siguiente ejemplo dados 3 números definir cuál de ellos es mayor e imprimirlo.





Estructura para

En muchas ocasiones se conoce de antemano el número de veces que se desean ejecutar las acciones de un bucle. Esta estructura permite ejecutar una o varias instrucciones un determinado de veces finita, fija.

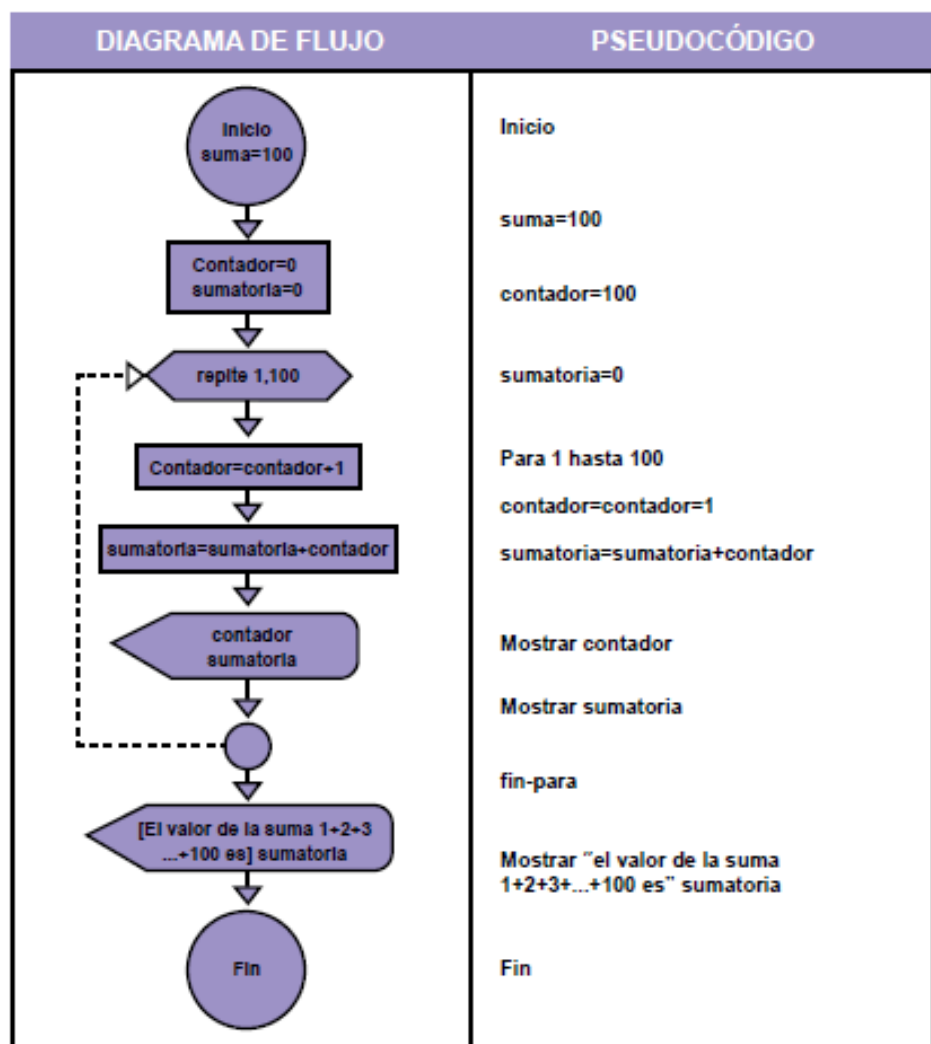


Figura Diagrama con estructura cíclica y/o repetitiva

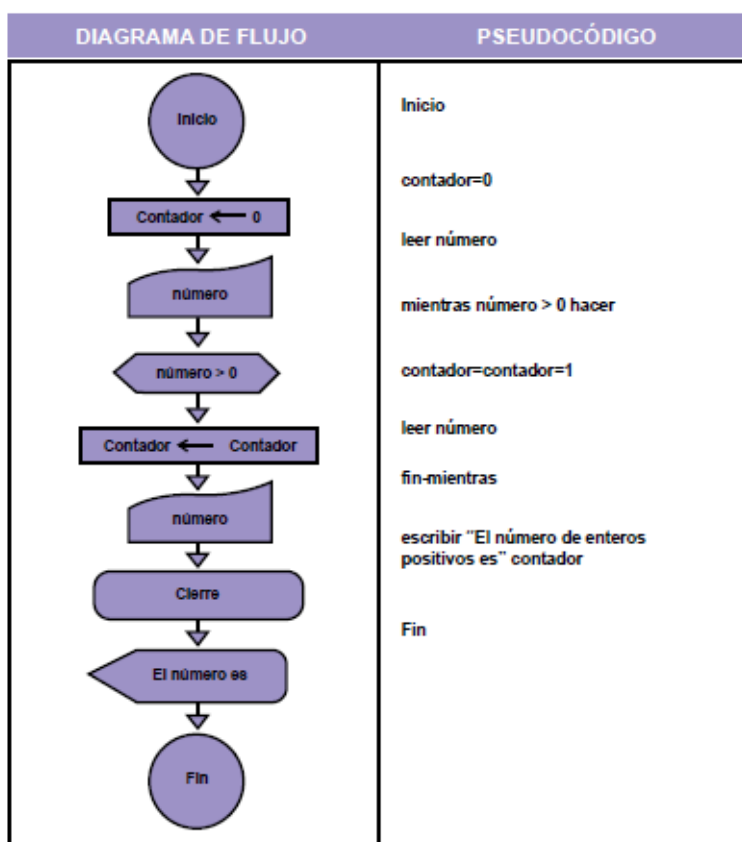


Estructura mientras

Se llama Mientras a la estructura algorítmica que se ejecuta mientras la condición evaluada resulte verdadera. Se evalúa la expresión booleana y, si es cierta, se ejecuta la instrucción especificada, llamada el cuerpo del bucle. Entonces se vuelve a evaluar la expresión booleana, y si todavía es cierta se ejecuta de nuevo el cuerpo. Este proceso de evaluación de la expresión booleana y ejecución del cuerpo se repite mientras la expresión sea cierta.

Cuando se ejecuta la instrucción mientras, la primera cosa que sucede es que se evalúa la condición (una expresión booleana). Si se evalúa falsa, ninguna acción se toma y el programa prosigue en la siguiente instrucción del bucle. Si la expresión booleana es verdadera, entonces se ejecuta el cuerpo del bucle, después de lo cual se evalúa de nuevo la expresión booleana. Este proceso se repite una y otra vez mientras la expresión booleana (condición) sea verdadera.

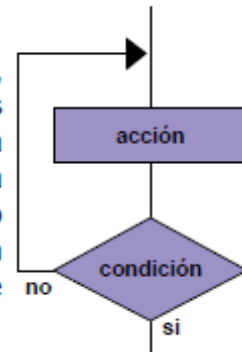
Figura Estructura con la función Mientras.





Estructura Repita

La estructura cíclica REPITA, al igual que la estructura cíclica mientras, se ejecuta un número indeterminado de veces, estas dos estructuras tienen un comportamiento similar, presentando su principal diferencia en el lugar de estructura donde se evalúa la condición, dado que la estructura MIENTRAS evalúa la condición del ciclo al inicio del mismo y la estructura REPITA lo hace al final del mismo, de este modo, en la estructura cíclica REPITA, el programador garantiza que el ciclo se ejecuta al menos una vez. (Véase Figura).



4 Actividades de transferencia del conocimiento.

DESARROLLAR EL SIGUIENTE TALLER USANDO SÓLO EL CICLO PARA. Los ejercicios los debe realizar en la Herramienta DFD y PSeint

1. Calcular el promedio de un alumno que tiene 7 calificaciones en la materia de Diseño Estructurado de Algoritmos. Se le debe indicar al usuario que la nota debe ser entre 0 y 10.
2. Leer 10 números y si el número es Positivo mostrar en pantalla “El número XX es Positivo”, si el número es Cero o Negativo mostrar nada
3. Leer 20 números e imprimir cuantos son positivos, cuantos negativos y cuantos cero.
4. Suponga que se tiene un conjunto de calificaciones de 5 Notas obtenidas. Realizar un algoritmo para calcular la calificación promedio de las 5 notas y la calificación más baja de las 5 notas. Debe mostrar al final en Pantalla el Promedio de las 5 Notas fue XXX y la nota más baja fue XXX. Debe indicarle al usuario por pantalla que la nota debe ser entre 1 y 5.
5. Calcular e imprimir la tabla de multiplicar de un número cualquiera. Imprimir el multiplicando, el multiplicador y el producto. Se deben mostrar por EJEMPLO: $2 \times 2 = 4$
6. Una persona debe realizar un muestreo con 8 personas para determinar el promedio de peso de los niños, jóvenes, adultos y viejos que existen en su zona habitacional. Se determinan las categorías con base en la siguiente, tabla:

Categoría	Edad
Niños	0-12
Jóvenes	13-29
Adultos	30-59
Viejos	60 en adelante

Se debe solicitar la edad y el peso de cada persona y calcular y mostrar el promedio por categoría.

7. Al cerrar un expendio de naranjas, 5 clientes recibirán un 15% de descuento si compran más de 10 kilos. Determinar cuánto pagara cada cliente y cuanto percibirá la tienda por esas compras.

**SERVICIO NACIONAL DE APRENDIZAJE SENA****GUÍA DE APRENDIZAJE****SISTEMA INTEGRADO DE GESTIÓN**

Proceso Gestión de la Formación Profesional Integral

Procedimiento Ejecución de la Formación Profesional Integral

Versión: 02

Código: GFPI-F-019

8. Realizar un programa que pida 10 números y diga cuantos son pares. **Para sacar los número primos se debe usar la función mod 2**

9. Calcular el número mayor y el número de menor de una lista de 7 números.

10. Realice un programa que imprima los múltiplos del 5 entre 0 y 100. **Usar función mod 5**

11. Realizar un programa que permita efectuar la venta de 4 productos, donde para cada producto debe calcular el subtotal, IVA (19%) y total, visualizando estos resultados, al terminar la venta de los 4 productos visualizar el total de IVA generado por los 4 productos, y el total de la venta.

EJEMPLO Escriba el precio del Producto: 100

Subtotal: 100

IVA: 19

TOTAL: 119

Luego debe decir El total del IVA por concepto de los 4 productos Fue: XXXXX

El Valor total de los 4 Productos fue: XXXX

12. Utilizando el ejercicio anterior, adicionalmente se pide efectuar un descuento por venta acorde a la siguiente tabla:

Subtotal	Descuento
Menor de 50.000	3%
Entre 50.000 y 100000	4%
Entre 100000 y 500000	6%
Mayor de 500000	7%

13. Realizar un programa que muestre los pares que hay entre 1 y N.

14. Dado un grupo de 5 Números (Diferentes a Cero), realice un algoritmo que permita determinar y dar el número mayor y menos de la lista.

15. La Empresa Aguas de Manizales Posee dos tipos de clientes: Residencial y Empresarial, realice un algoritmo que permita procesar “n” cantidad de clientes teniendo como datos por C/U. de ellos:

- Código del Cliente



SERVICIO NACIONAL DE APRENDIZAJE SENA

GUÍA DE APRENDIZAJE

SISTEMA INTEGRADO DE GESTIÓN

Proceso Gestión de la Formación Profesional Integral

Procedimiento Ejecución de la Formación Profesional Integral

Versión: 02

Código: GFPI-F-019

- Tipo de Cliente (residencial o Empresarial)

- Lectura Actual de consumo en Mts³

- Lectura Anterior en Mts³

La Lectura Actual SIEMPRE debe ser Mayor a la Lectura Anterior

Se debe Calcular y dar como salida la cantidad de Mts³ y el neto a pagar por cada cliente.

El consumo se cobran a razón de:

- Residencial 8500 por mt³.
- Empresarial 10800 por mt³.

16. Una Ferretería vende dos tipos de Cables, Cable Tipo A (\$20000 MT) y Cable Tipo B (\$35000 MT); resuelva el problema en DFD y PSeint que teniendo como datos por cada cliente su nombre, tipo de cable a comprar y cantidad de metros requeridos, calcule y de cómo salida el nombre y el neto a pagar por cada cliente.

Se deben tomar las siguientes consideraciones:

- El número de compradores es N Usuarios
- La empresa da una rebaja del 10% por cada compra que exceda de los 100 MT de cable de cualquier tipo.

16. Una Empresa de Reproducción Digital desea realizar un algoritmo que permita calcular y dar como salida el pago que tiene que hacer cada cliente por concepto de copias a \$200 pesos, tomando en cuenta que se tienen los siguientes datos: (Realizar con 5 clientes)

* Tipo de Copia (Carta, Oficio)

* Cantidad de Copias

Además el programa se debe hacer en DFD y PSeint y debe mostrar:

* Total de Copias tipo Carta y Tipo Oficio procesadas por la empresa.

* Costo de total de las copias en carta y en oficio

4.1 Actividades de evaluación.

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
---------------------------	-------------------------	---------------------------------------

**SERVICIO NACIONAL DE APRENDIZAJE SENA****GUÍA DE APRENDIZAJE****SISTEMA INTEGRADO DE GESTIÓN**

Proceso Gestión de la Formación Profesional Integral

Procedimiento Ejecución de la Formación Profesional Integral

Versión: 02

Código: GFPI-F-019

**Evidencias de
Conocimiento :**

Ejercicios

- Elabora el plan de trabajo, de acuerdo con la interpretación del informe técnico de diseño, según normas y protocolos de la empresa.

CuestionarioEsta guía se evaluará con el instrumento
Instrumento DE Evaluación IEADSI01



4. RECURSOS PARA EL APRENDIZAJE

ACTIVIDADES DEL PROYECTO	DURACIÓN (Horas)	Materiales de formación devolutivos: (Equipos/Herramientas)		Materiales de formación (consumibles)		Talento Humano (Instructores)		AMBIENTES DE APRENDIZAJE TIPIFICADOS
		Descripción	Cantidad	Descripción	Cantidad	Especialidad	Cantidad	ESCENARIO (Aula, Laboratorio, taller, unidad productiva) y elementos y condiciones de seguridad industrial, salud ocupacional y medio ambiente
Análisis e interpretación de la información recolectada en el trabajo de campo	60 horas	Computadores personales	30	Hojas de block Marcador	200 2	Análisis y desarrollo de sistemas de información	1	Sala de informática



5. GLOSARIO DE TERMINOS

LÓGICA: Es una secuencia de operaciones realizadas por el hardware o por el software.

ALGORITMO: Conjunto de sentencias / instrucciones en lenguaje nativo, los cuales expresan la lógica de un programa.

BIT:(dígito binario) un dígito simple de un numero binario (1 ó 0)

BYTE: Grupo de bits adyacentes operados como una unidad, (grupos de 8 bits).

DIAGRAMA DE FLUJO: Es la representación gráfica de una secuencia de instrucciones de un programa que ejecuta un computador para obtener un resultado determinado.

CÓDIGO FUENTE: Programa en su forma original, tal y como fue escrito por el programador, el código fuente no es ejecutable directamente por el computador, debe convertirse en lenguaje de maquina mediante compiladores, ensambladores o intérpretes.

CAMPO: Es el espacio en la memoria que sirve para almacenar temporalmente un dato durante el proceso, Su contenido varia durante la ejecución del programa.

CAMPO NUMÉRICO, el que solo puede almacenar valores (dígitos).

CAMPO ALFANUMERICO, el que puede almacenar cualquier carácter (dígito, letra, símbolo especial).

COMPILADOR: Programa de computadora que produce un programa en lenguaje de máquina, de un programa fuente que generalmente esta escrito por el programador en un lenguaje de alto nivel.

VARIABLE: En programación es una estructura que contiene datos y recibe un nombre único dado por el programador, mantiene los datos asignados a ella hasta que un nuevo valor se le asigne o hasta que el programa termine.

CONSTANTE: Valor o conjunto de caracteres que permanecen invariables durante la ejecución del programa.

DATO: El termino que usamos para describir las señales con las cuales trabaja la computadora es dato; Aunque las palabras dato e información muchas veces son usada indistintamente, si existe una diferencia importante entre ellas. En un sentido estricto, los datos son las señales individuales en bruto y sin ningún significado que manipulan las computadoras para producir información.

PROGRAMA: Es una colección de instrucciones que indican a la computadora que debe hacer. Un programa se denomina software, por lo tanto , programa, software e instrucción son sinónimos.



SERVICIO NACIONAL DE APRENDIZAJE SENA

GUÍA DE APRENDIZAJE

SISTEMA INTEGRADO DE GESTIÓN

Proceso Gestión de la Formación Profesional Integral

Procedimiento Ejecución de la Formación Profesional Integral

Versión: 02

Código: GFPI-F-019

6. REFERENTES BIBLIOGRÁFICOS

Fundamentos de Programación. MacGraw Hill - 2006

http://books.google.es/books/about/L%C3%B3gica_de_programaci%C3%B3n.html?id=Z_n5lbyJfrQC

7. CONTROL DEL DOCUMENTO (ELABORADA POR)

Elaborado por:

Andrés Julián Hoyos Caicedo

Ingeniero de Sistemas y Telecomunicaciones

Instructora, Centro de Procesos Industriales, SENA – Regional Caldas.