

PROGRAMACION I

Unidad 1

Pasos para realizar la resolución de problema usando un Lenguaje de Programación

Pasos	Etapas	Descripción
1	Análisis del problema	Conducen al diseño detallado por medio un código escrito en forma de un algoritmo.
2	Especificación del Problema	Definición precisa del problema.
3	Diseño de algoritmo	Elegir una solución y escribir el algoritmo.
4	Codificación	Se implementa el algoritmo en un código escrito en un lenguaje de programación. Refleja las ideas desarrolladas en las etapas de análisis y diseño
5	Compilación y ejecución	Traduce el programa fuente a programa en código de maquina y lo ejecuta.
6	Verificación	Verificar que el programa cumple con la especificación.
7	Depuración	Buscar errores ,corregirlos o eliminarlos.
8	Documentación	Son comentarios, etiquetas de texto, que facilitan la comprensión del programa

CONCEPTO DE ALGORITMO

Un algoritmo es un conjunto preciso, finito y ordenado de instrucciones o reglas bien definidas y no ambiguas que describe, paso a paso, la secuencia de pasos a seguir para resolver un problema o realizar una tarea específica.

- Es una secuencia de operaciones que transforma la entrada de datos en la salida deseada.
- El diseño del *algoritmo*, debe ser independiente de la computadora que resuelve el problema.

Aquí hay algunos puntos clave sobre los algoritmos:

1.Precisión: Los algoritmos deben ser precisos y sin ambigüedades. Cada paso debe estar claramente definido y no dejar lugar a interpretaciones.

2.Finitud: Un algoritmo debe ser finito, lo que significa que debe tener un número finito de pasos. No puede haber bucles infinitos o instrucciones indefinidas.

3.Orden: Los pasos de un algoritmo deben estar organizados de manera secuencial y ordenada, de modo que cada paso se ejecute después del anterior y antes del siguiente.

Aquí hay algunos puntos clave sobre los algoritmos:

1.Determinismo: Un algoritmo debe ser determinista, lo que significa que para una entrada dada, siempre producirá la misma salida.

2.Eficiencia: Se espera que un algoritmo sea eficiente en términos de tiempo y recursos utilizados. Se busca optimizar la cantidad de operaciones necesarias para alcanzar el resultado deseado.

3.Generalidad: Los algoritmos deben ser lo más generales posible, es decir, deben ser aplicables a una amplia gama de problemas similares.

PROCESO

Es un conjunto de acciones organizados en el tiempo y concebido como activo.

- Describir un proceso es interpretar un acontecimiento como un conjunto de acciones más elementales cuyo efecto acumulativo es el mismo que el producido por el acontecimiento completo.

ACCION

Es un acontecimiento producido por un actor ejecutante.

- Tiene un lugar durante un período de tiempo finito y produce un resultado bien definido y previsto. Es decir que existe un instante de inicio de la acción(t_0) y un instante de fin de la acción (t_1).

ACCION

Es una actividad que comienza con dato y termina con un resultado.

- Para que sea posible reconocer el resultado es preciso que el sistema observado durante este lapso de tiempo este provisto de indicadores que tomen valores diferentes. Estos indicadores se denominan **INFORMACIÓN**.
- Todo lo que se puede detectar en los estados intermedios se denomina también **INFORMACIÓN**.

Ejemplo:

Un panadero mezcla harina, levadura y agua y obtiene una masa que luego hornea para obtener pan:

Ei: harina, levadura y agua

Acción 1: Mezclar los ingredientes.

Acción 2: Amasar

Acción 3: Hornear

E_f: pan

Estado intermedio: masa

- El conjunto de los valores de los distintos indicadores en un instante dado (t) del desarrollo del acontecimiento se denomina **ESTADO** en el instante t .
- El **RESULTADO** es el estado del sistema en el instante t_1 .
- Si se conoce el estado del sistema en el instante t_0 , entonces se puede determinar cual será el estado de sistema en t_1 .
- El estado del sistema en t_0 define los datos de la acción.

Ejemplo:

Un panadero mezcla harina, levadura y agua y obtiene una masa que luego hornea para obtener pan:

Acción principal: Obtener pan

Ei: harina, levadura y agua

Ei1: Ingredientes mezclados

Acción 1: Mezclar los ingredientes.

Acción 2: Amasar

Ei2: masa

Acción 3: Hornear

Ef: pan

Actividad:

- Identificar el estado inicial y final
- Objeto/s sobre el cual se realiza la acción
- Identificar la acción principal
- Identificar las acciones necesarias para alcanzar el estado final.
- De ser posible identificar los estados intermedios

Enunciado:

Colgar un cuadro en la pared para lo cual se cuenta con martillo y clavo.

Acción principal: Colgar el cuadro en la pared

Objetos: pared, cuadro

Ei: cuadro, pared, martillo y clavo

Acción 1: clavar el clavo en la pared.

Ei1: clavo clavado en la pared, cuadro sin colgar, martillo

Acción 2: Colgar el cuadro

Ef: cuadro colgado, martillo

AMBIENTE

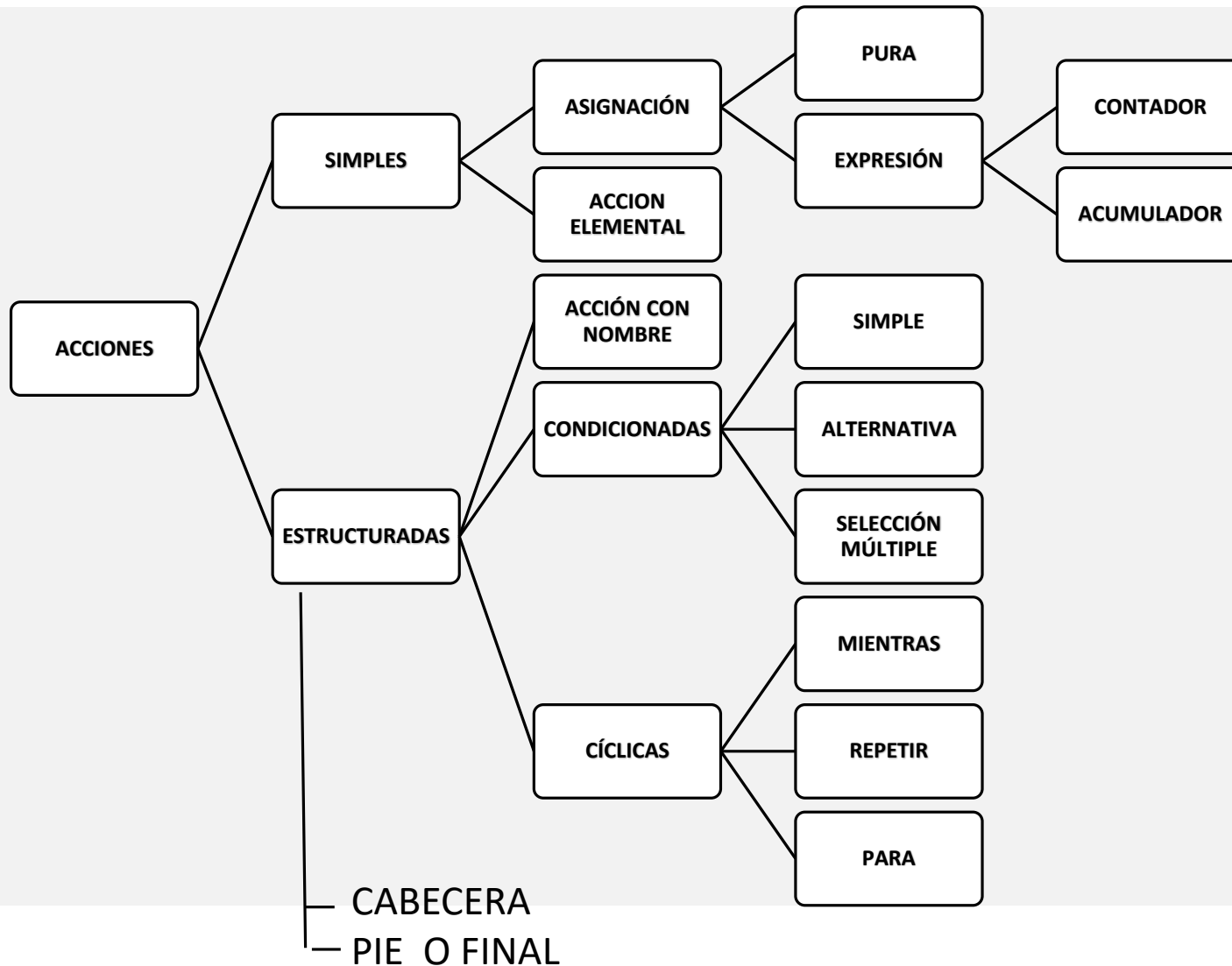
Es el conjunto de todos los recursos necesario para la ejecución de un trabajo.

- Una ACCIÓN en un evento que modifica el ambiente. Una acción transforma en ambiente desde un estado E_i (estado inicia) a otro estado E_f (estado final).
- Una acción es primitiva si su enunciado es suficiente para que se pueda ejecutar sin información adicional.
- Una acción no primitiva debe ser descompuesta en acciones primitivas.

Tipos de Datos

NOMBRE	CONJUNTO DE VALORES	OPERACIONES
Enteros	Negativos y positivos sin decimal	Sumar, restar, dividir, multiplicar, residuo
Reales	Negativos y positivos, con decimal	Sumar, restar, dividir, multiplicar
Lógicos	Verdadero o Falso(1 o 0)	And, Or, Not
Caracteres	Letras, números, especiales, juntos forman una cadena	..

ESTRUCTURA DE LAS ACCIONES



Constantes

Las constantes son valores que no pueden cambiar en la ejecución del programa.

Variables

Las variables son valores que se pueden modificar durante la ejecución de un programa.

ASIGNACIÓN

Es la acción que da el valor de una expresión a una variable-.

RECEPTOR:=EMISOR

- El valor que se asigna a la variable debe ser del mismo tipo que la variable.

Es la reunión de datos (constantes y variables) relacionados mediante operadores.

- Las expresiones están formadas por operandos y operadores.
- La expresión puede consistir en una entidad simple, como una constante, una variable, o una referencia a una función. También puede consistir en alguna combinación de tales entidades interconectadas por uno o más operadores.

Operadores

Un operador es un símbolo que indica alguna operación sobre uno o varios objetos del lenguaje, a los que se denomina operandos.

Los operadores, junto con los operandos, forman expresiones.

Atendiendo al tipo de operación que realizan, se clasifican en :

- Aritméticos**
- Relacionales**
- Lógicos**

Operadores Aritméticos

Tipo de Operador	OPERADOR	DESCRIPCIÓN
UNARIOS	-	Negación
BINARIOS	-	Resta
	+	Suma
	*	Producto
	/	División
	MOD	Resto de división entera
	DIV	División entera sin parte fraccionaria

Operadores Relacionales

Se usan para expresar condiciones y describir una relación entre dos valores.

resultado de una expresión relacional sólo puede ser *verdadero* o *falso*, lo que en C se identifica con los valores distinto de cero y cero, respectivamente.

	<i>OPERADOR</i>	<i>DESCRIPCIÓN</i>
<i>BINARIOS</i>	<i>></i>	<i>Mayor que</i>
	<i>>=</i>	<i>Mayor o igual que</i>
	<i><</i>	<i>Menor que</i>
	<i><=</i>	<i>Menor o igual que</i>
	<i>=</i>	<i>Igual que</i>
	<i><></i>	<i>Diferente que</i>

Operadores Lógicos

Actúan sobre expresiones booleanas, es decir, sobre valores *verdadero* o *falso* generados por expresiones como las explicadas en el caso anterior.

	<i>OPERADOR</i>	<i>DESCRIPCIÓN</i>
<i>UNARIOS</i>	-	<i>not</i>
<i>BINARIOS</i>	\wedge	<i>Y (and)</i>
	\vee	<i>O (or)</i>

El resultado de una operación lógica viene dado por su tabla de verdad

TABLA DE VERDAD

Una **TABLA DE VERDAD** es una tabla que muestra el valor de verdad de una proposición compuesta, para cada combinación de verdad que se pueda asignar.

Negación

A	\neg	A
V		F
F		V

Conjunción

A	B	\wedge	B
V	V	V	
V	F	F	
F	V	F	
F	F	F	

Disyunción

A	B	\vee	B
V	V	V	
V	F	V	
F	V	V	
F	F	F	

- Las tablas de verdad son una herramienta que nos ayudan a determinar cuáles son las condiciones necesarias para que sea verdadero o válido un enunciado propuesto.
- Estas se usan como respaldo para tomar decisiones a partir de un resultado dado.

PRECEDENCIA DE OPERADORES

Se refiere a la reglas que establecen el orden en que se deben ejecutar las operaciones dentro de una expresión.

En otras palabras, cuando tenemos una expresión que tiene múltiples operadores, no todos tienen la misma prioridad. Si encontramos una expresión como:

$$a = 2 + 3 * 5$$

¿Se calcula primero el $=$, el $+$ o el $*$?

En general, la precedencia de operadores en programación está estrechamente relacionada con las reglas de cálculo matemáticas. De hecho, en muchos casos se emplean las mismas regla, como que las operaciones de multiplicación y división se realizan antes que las de suma y resta.

TABLA DE PRECEDENCIA

- Paréntesis: Los operadores dentro de paréntesis se evalúan primero. Los paréntesis se utilizan para alterar la precedencia y forzar el orden deseado de evaluación.
- Exponenciación: De existir, los operadores de exponenciación se evalúan después de los paréntesis. Este operador se utiliza para elevar un número a una potencia.
- Multiplicación, división y módulo: Estos operadores tienen una precedencia superior a la de la suma y la resta. Se evalúan de izquierda a derecha, a menos que se utilicen paréntesis para alterar el orden.
- Suma y resta: Estos operadores se evalúan después de la multiplicación, la división y el módulo. Al igual que los operadores anteriores, se evalúan de izquierda a derecha a menos que se utilicen paréntesis.
- Operadores de comparación: Los operadores de comparación (como menor que $<$, mayor que $>$) tienen una precedencia menor que los operadores aritméticos.
- Operadores de equidad: Los operadores de equidad (igual a y distinto a) tienen una precedencia inferior a los de comparación.
- Operador lógico \wedge (AND): A continuación tendríamos los operadores lógicos. Generalmente el operador lógico AND tiene mayor precedencia.
- Operador lógico \vee (OR): Finalmente, terminamos nuestro listado genérico con el operador lógico OR, que tiene precedencia inferior a todos los anteriores.

ESQUEMA

Es una descripción o representación mental reducida a un proceso.

- La actividad de esquematización se apoya en el conocimiento de técnicas y herramientas que permiten analizar lo que ocurre (abstracción) y de notaciones que permiten comunicar a los demás el resultado de tales análisis..

En esta etapa se realizará una representación gráfica de la secuencia. Estas representaciones son las herramientas pueden ser:

- *diagramas de flujo,*
- *pseudocódigos*

Diagrama de Flujo

Representa la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema.

- Se les llama diagramas de flujo porque los símbolos utilizados se conectan por medio de flechas para indicar la secuencia de operación
- Su correcta construcción es sumamente importante porque, a partir del mismo se escribe un programa en algún Lenguaje de Programación.
- Si el Diagrama de Flujo está completo y correcto, el paso del mismo a un Lenguaje de Programación es relativamente simple y directo.

Diagrama de Flujo - Símbolos



Este se utiliza para representar el inicio o el fin de un algoritmo. También puede representar una parada o una interrupción programada que sea necesaria realizar en un programa.



Este se utiliza para un proceso determinado, es el que se utiliza comúnmente para representar una instrucción, o cualquier tipo de operación que origine un cambio de valor.



Este símbolo es utilizado para representar una entrada o salida de información, que sea procesada o registrada por medio de un periférico.



Este es utilizado para la toma de decisiones, ramificaciones, para la indicación de operaciones lógicas o de comparación entre datos.

Diagrama de Flujo - Símbolos



Este es utilizado para enlazar dos partes cualesquiera de un diagrama a través de un conector de salida y un conector de entrada. Esta forma un enlace en la misma página del diagrama.



Este es utilizado para enlazar dos partes de un diagrama pero que no se encuentren en la misma página.



Este es utilizado para indicar la secuencia del diagrama de flujo, es decir, para indicar el sentido de las operaciones dentro del mismo.



Este es utilizado para representar la salida de información por medio de la impresora.



Este es utilizado para representar la salida o para mostrar la información por medio del monitor o la pantalla.

Inicio

Leer las medidas A, B
y C.

Calcular el volumen.

Escribir el volumen
obtenido.

Fin.

Inicio



A, B, C



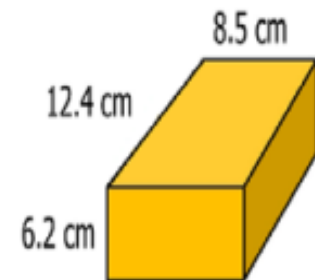
Volumen = A*B*C



Volumen



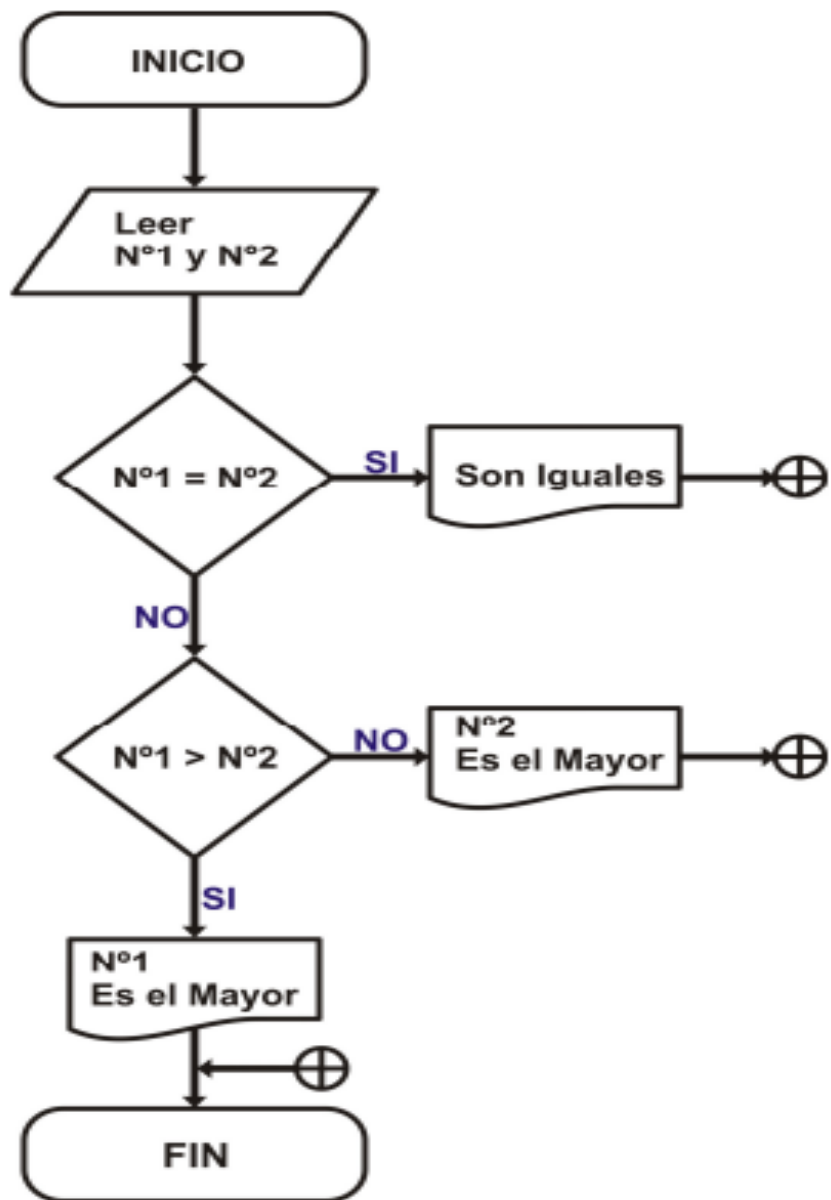
Fin



Volumen = ?

Resolver el siguiente problema

¿Cómo es posible determinar si 2 números son iguales o diferentes y si son diferentes determinar cual de los 2 es el mayor?



Lenguaje Estructurado

El lenguaje estructurado es un lenguaje natural limitado en palabras y construcciones, lo que le da más precisión y claridad, evitando ambigüedades (el lenguaje natural humano carece de precisión y es muy ambiguo). No es oral, sólo escrito.

Los lenguajes estructurados NO deben confundirse con los lenguajes de programación estructurados ni la programación estructurada que es un paradigma de programación.

Los pseudocódigos son lenguajes estructurados que permiten describir algoritmos (no necesariamente de programación).

El pseudocódigo es un conjunto estructurado de sentencias que se utiliza para representar un algoritmo.

En pseudocódigo se describen los algoritmos utilizando una mezcla de lenguaje común, con instrucciones de programación, palabras claves, etc.

El **OBJETIVO** es que el programador se **centre en la solución lógica del algoritmo** y no en la implementación en un lenguaje de programación concreto (con las posibles complicaciones en las reglas sintácticas), o en otras palabras, sólo ayudan a "pensar" un programa antes de escribirlo en un lenguaje de programación formal.

El pseudocódigo es una herramienta muy útil para los programadores ya que les permite diseñar algoritmos de una manera más clara y sencilla. A través de él, se pueden planificar las distintas etapas del proceso de programación, definiendo las tareas que deben llevarse a cabo en cada una de ellas, así como las variables y operaciones necesarias para llevarlas a cabo.

- El pseudocódigo es una descripción informal de alto nivel del principio de funcionamiento de un programa informático u otro algoritmo.
- No está escrito en ningún lenguaje de programación específico, sino que utiliza un lenguaje natural y símbolos fácilmente comprensibles para el ser humano.

Una frase en lenguaje estructurado puede consistir en una ecuación algebraica, por ejemplo:

$$x = (a + b) * 6$$

También puede ser una sencilla frase imperativa que consista en un verbo y un objeto. Nótese que esta frase no tiene el punto y coma que termina una instrucción en muchos lenguajes de programación; puede o no terminar con un punto (",").

Además, las frases que describen los cálculos deben ser verbos orientados a la acción como CALCULAR, ORDENAR, RESTAR, etc., por lo que se pudo haber escrito el ejemplo anterior así:

$$\text{CALCULAR } x = (a + b) * 6$$

Pseudocódigo de un algoritmo que calcule la media de tres números:

Inicio

Leer (n1)

Leer (n2)

Leer (n3)

$\text{suma} = n1 + n2 + n3$

$\text{media} = \text{suma} / 3$

escribir (media)

Fin

Inicio

Leer (n1)

Leer (n2)

Leer (n3)

Calcular $\text{suma} = n1 + n2 + n3$

Calcular $\text{media} = \text{suma} / 3$

escribir (media)

Fin

El orden en el que se realizan las operaciones es importante: no puede calcularse la media sin antes haber leído los números.

Acciones Estructuradas

Acción con nombre

Es la incorporación de un nombre al conjunto de acciones que se tienen que ejecutar.

Acción nombre-acción es
Ambiente

var 1: Tipo de dato;

var 2: Tipo de dato;

.....

var n: tipo de dato;

Algoritmo

Acción 1;

Acción 2;

.....

Acción n;

facción

Inicio

Escribir: "¿Cual es tu edad?"

Leer Edad

Escribir "Tu edad es" + Edad

Fin

**Acción Edad es
Ambiente**

Edad:entero;

Algoritmo

Escribir: "¿Cual es tu edad?";

Leer (Edad);

Escribir (Edad);

facción

Consigna

Calcular la suma de dos números que se ingresan por pantalla.

Acción Sumar es

Ambiente

n1,n2,suma:entero

Algoritmo

Escribir: "Ingrese n1";

Leer (n1);

Escribir: "Ingrese n2";

Leer (n2);

Calcular suma=n1+n2;

Escribir ("El resultado de la operación es," suma);

facción

Consigna

Calcular la suma de dos números que se ingresan por pantalla.

Consigna

Calcular la suma de dos números que se ingresan por pantalla.

Inicio

Escribir: "Ingrese dos números"

Leer n_1

Leer n_2

Escribir "La suma es" , n_1+n_2

Fin

Acción Suma es
Ambiente

n1,n2,sum:entero

Algoritmo

Escribir: “Ingrese n1”;

Leer (n1);

Escribir: “Ingrese n2”;

Leer (n2);

Calcular $sum = n1 + n2$;

Escribir (“El resultado de la operación es”, sum);

facción

Consigna

Intercambiar el valor de dos variables

Acción Suma es Ambiente

n1,n2,n3:entero

Algoritmo

Escribir: “Ingrese n1”;

Leer (n1);

Escribir: “Ingrese n2”;

Leer (n2);

n3=n1;

n1=n2;

n2=n3;

Escribir(n1);

Escribir(n2);

Escribir(n3);

facción

Consigna

Calcular el promedio de tres valores que se ingresan por teclado

Acción Suma es Ambiente

**n1,n2,n3:entero
prom:real;**

Algoritmo

**Escribir: "Ingrese n1";
Leer (n1);
Escribir: "Ingrese n2";
Leer (n2);
Escribir: "Ingrese n3";
Leer (n3);
prom=(n1+n2+n3)/3;
Escribir(prom);**

facción