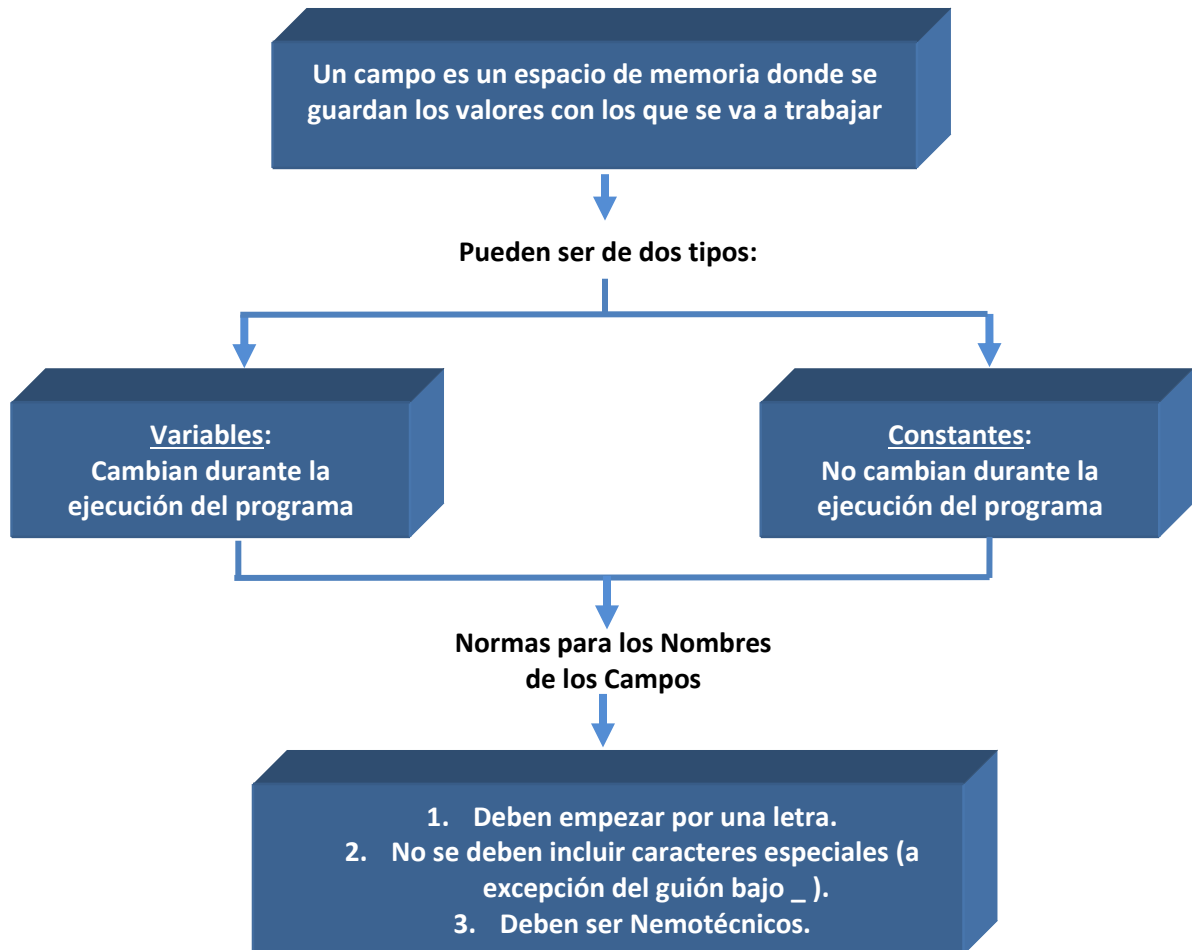
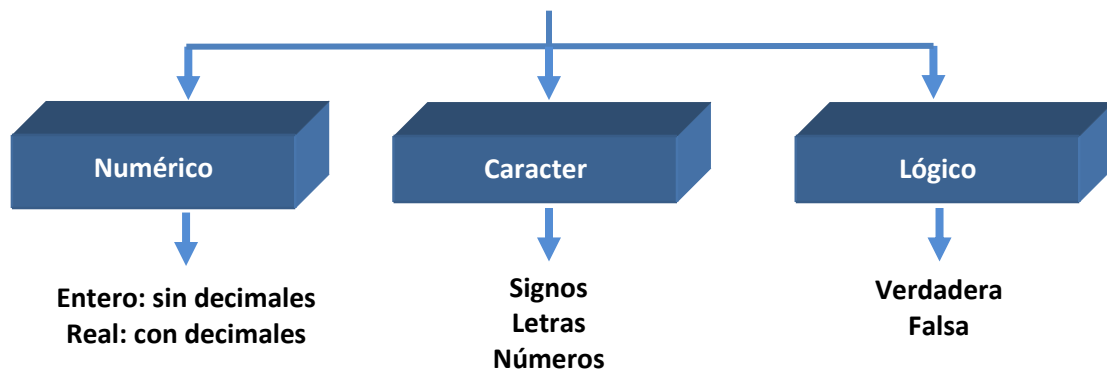




TIPOS DE CAMPOS



TIPOS DE CAMPOS E INFORMACIÓN





ESTRUCTURA SECUENCIALES

Este tipo de estructura, permite representar las instrucciones que se ejecutan una tras otra en un algoritmo. Se basan fundamentalmente en tres tipos de instrucciones:

1. INSTRUCCIONES DE ENTRADA DE DATOS:

- ▶ Son las que indican cuáles son los valores necesarios para resolver el problema.
- ▶ Estos valores ingresan de un medio externo y son llevados a las direcciones de memoria donde se encuentran las variables.
- ▶ Por ejemplo, si se desea ingresar dos números para sumarlos, su representación sería:

Lea: numero1, numero2

2. INSTRUCCIONES DE ASIGNACIÓN:

- ▶ Consiste en asignarle a una variable, el resultado de una expresión.
- ▶ Generalmente, estas instrucciones hacen parte del proceso de un algoritmo.
- ▶ Por ejemplo, si se desea guardar en una variable, la suma de dos números, su representación sería:

suma ← numero1 + numero2

- ▶ **Importante:** Tanto la variable (lado izquierdo) como la expresión (lado derecho), deben ser siempre del mismo tipo de dato.

3. INSTRUCCIONES DE SALIDA DE DATOS:

- ▶ Son las que extraen los valores de la memoria de las variables hacia un medio de salida.
- ▶ En esta instrucción se pueden mostrar los resultados con mensajes, entre comillas dobles y separados por coma de las variables.
- ▶ Por ejemplo, si se desea ver el valor del resultado de una suma, su representación sería:

Escriba: "El resultado de la suma es", suma



EJEMPLO 1:

Con base en lo anterior, se realizará el primer algoritmo, teniendo en cuenta también, los primeros 4 pasos de la Clase Conceptos Básicos.

1. Problema:

Haga un algoritmo, que calcule la suma de dos números ingresados por el usuario.

2. Análisis:

Tener en cuenta:

- ▶ Lo que está después de // se toma como comentario, es decir que es algo de guía para los programadores, pero que no influye para nada en el algoritmo que se está haciendo.
- ▶ Se debe recordar siempre, que el Análisis debe hacerse en términos de variables, constantes y asignaciones.

Datos de Entrada: //qué se tiene o qué se necesita para resolver el problema
 numero1, numero2

Datos de Salida: //el resultado que pide el problema
 suma

Proceso: //qué se hace con lo que se tiene, para calcular el resultado
 suma $\leftarrow$ numero1 + numero2

3. Algoritmo:

De manera general, acorde a lo anterior, en una estructura secuencial, que es lo básico para un algoritmo, se tienen los siguientes pasos:

1. Inicio
2. Declaración de variables y constantes
3. Solicitud de Datos (para ello, se utiliza el lea y el escriba juntos, porque si se utiliza el lea solo, el usuario no va a saber qué es lo que debe digitar)
4. Cálculos (el proceso como tal que se hizo en el análisis, en términos de asignaciones)
5. Mostrar los resultados
6. Fin

Con base en lo anterior, el algoritmo para el problema dado, sería el siguiente:



Inicio // Paso 1

Real: numero1, numero2, suma // Paso 2

Escriba: "Digite dos números para sumarlos" //Paso 3

Lea: numero1, numero2

suma $\leftarrow$ numero1 + numero2 //Paso 4

//Aquí se usan las expresiones, en las instrucciones de asignación.

Escriba: "El resultado de la suma es =", suma //Paso 5

Fin //Paso 6

- ▶ Al hacer el algoritmo, se debe tener en cuenta también, que se debe manejar una indentación tanto horizontal (lo que se conoce como sangría) y también una vertical, es decir que estén separados los pasos. Esto es muy importante cuando los algoritmos que se realizan son muy extensos.

4. Prueba de Escritorio:

Se debe recordar que la prueba de escritorio, es el seguimiento de los pasos del algoritmo para comprobar que este funcione y no tenga errores, algo así como una compilación manual. Con base en ello, una posible prueba de escritorio (dado que dependerá de los valores que se asignen para probar), sería la siguiente:

Inicio

numero1 //Aquí el compilador reserva los espacios de memoria,

numero2 //teniendo en cuenta el tamaño del tipo de datos

suma

Digite dos números para sumarlos //Sería lo que sale en pantalla

numero1 $\leftarrow$ 10 //El lea, al tomar los valores del usuario, se convierte en asignación

numero2 $\leftarrow$ 20

suma $\leftarrow$ 10 + 20 //Se reemplazan los valores y se desarrolla la expresión

suma $\leftarrow$ 30

El resultado de la suma es = 30

Fin



EJEMPLO 2:

Hacer un algoritmo que sume los números 5 y 3 y muestre el resultado.

Nota: Este ejercicio es parecido al anterior pero dado que se dan los valores en el problema, se toman **como constantes y no como variables** y por tanto, no se le piden los valores al usuario.

A continuación, se muestra como se deben desarrollar, los ejercicios que se les formulen como problemas.

Análisis:

Datos de Entrada: numero1 $\leftarrow$ 5, numero2 $\leftarrow$ 3
Datos de Salida: suma
Proceso: suma $\leftarrow$ numero1 + numero2

Algoritmo:

Inicio

Entero: numero1 $\leftarrow$ 5, numero2 $\leftarrow$ 3, suma

suma $\leftarrow$ numero1 + numero2

Escriba: "El resultado de la suma de ", numero1, "+", numero2, " = ", suma
//Este sería un ejemplo de un escriba mostrando varios valores

Fin

Prueba de Escritorio:

Inicio

numero1 $\leftarrow$ 5

numero2 $\leftarrow$ 3

suma

suma $\leftarrow$ 5 + 3

suma $\leftarrow$ 8

El resultado de la suma de 5+3 = 8

Fin



EJEMPLO 3:

Hacer un algoritmo que indique el número de meses transcurridos entre dos años cualesquiera.

Análisis:

Datos de Entrada: anio1, anio2

Datos de Salida: numeromeses

Proceso: numeromeses $\leftarrow$ (anio1 - anio2) * 12

Algoritmo:

Inicio

Entero: anio1, anio2, numeromeses

Escriba: "Digite dos años, siendo el primer año mayor"

Lea: anio1, anio2

numeromeses $\leftarrow$ (anio1 - anio2) * 12

Escriba: "El número de meses entre ", anio1, ", y ", anio2, ", es ", numeromeses

Fin

Prueba de Escritorio:

Inicio

anio1

anio2

numeromeses

Digite dos años, siendo el primer año mayor

anio1 $\leftarrow$ 2020

anio2 $\leftarrow$ 2010

numeromeses $\leftarrow$ (2020 - 2010) * 12

numeromeses $\leftarrow$ 10 * 12

numeromeses $\leftarrow$ 120

El número de meses entre 2020 y 2010 es 120

Fin