












# ESTRUCTURAS BASICAS

# DIAGRAMA DE FLUJO:

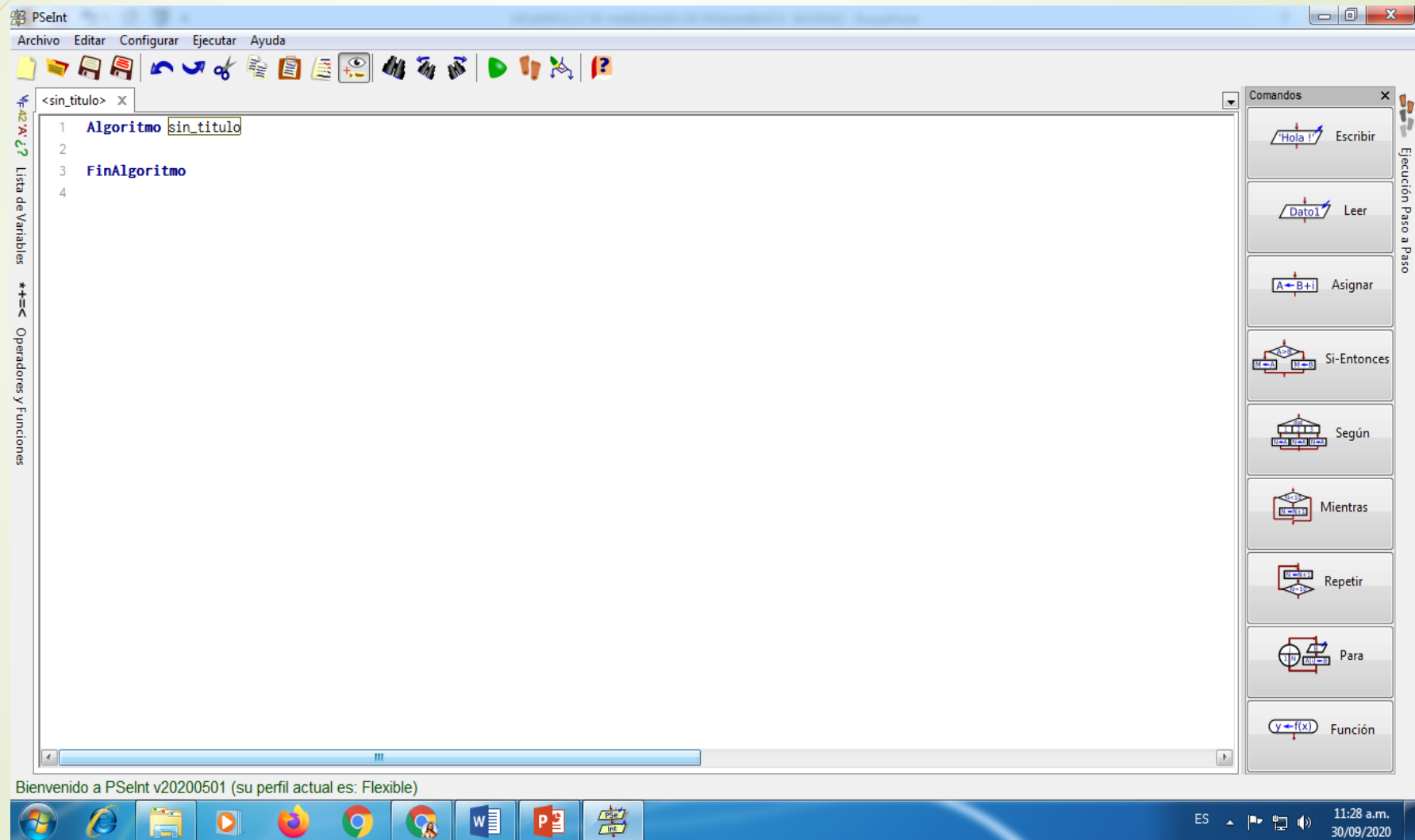
## Diagramas de flujo

Símbolo	Función
	<b>Terminal.</b> Representa el comienzo o el fin de un programa.
	<b>Entrada / Salida.</b> Indica una introducción de datos desde un dispositivo externo (por defecto, el teclado) o una salida de datos hacia algún dispositivo externo (por defecto, la pantalla)
	<b>Proceso.</b> Representa cualquier operación que se lleve a cabo con los datos del problema.
	<b>Condición.</b> Señala una bifurcación del flujo de instrucciones. La bifurcación está siempre controlada por una operación relacional llamada condición, cuyo resultado puede ser "verdadero" o "falso" (o también "sí" o "no"), dependiendo del valor de los datos de la expresión condicional. En función del resultado de dicha expresión, el flujo de ejecución continúa por una u otra rama (pero nunca por las dos a la vez)
	<b>Condición múltiple.</b> Sirve para indicar una bifurcación del flujo en varias ramas, no sólo en una. En este caso, la condición no puede ser booleana, sino entera.
	<b>Conector.</b> Para enlazar un fragmento del diagrama de flujo con otro fragmento situado en la misma página. Se usa cuando el diagrama es muy grande y no puede dibujarse entero de arriba a abajo.
	<b>Conector.</b> Como el anterior, pero para conectar un fragmento del diagrama con otro fragmento situado en una página diferente.
	<b>Dirección del flujo.</b> Indica el orden de ejecución de los pasos del algoritmo.
	<b>Subrutina.</b> Llamada a un subproceso o módulo independiente (ver apartado de "Programación Modular")

# CONCEPTOS BASICOS DE PROGRAMACIÓN

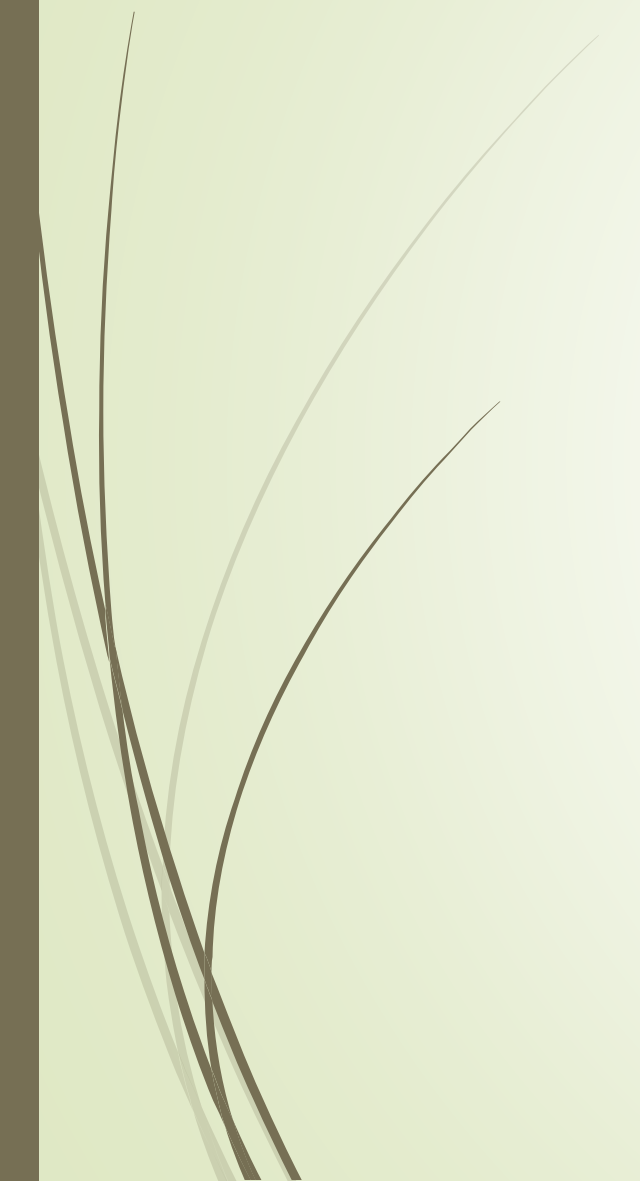
- ▶ **LENGUAJE DE PROGRAMACIÓN:** lenguajes artificiales que posibilitan codificar instrucciones que serán interpretadas y ejecutadas por la computadora u otro dispositivo. / debería ser capaz de ejecutar un algoritmo
- ▶ **Lenguaje informático:** Cualquier lenguaje que le dé instrucciones a una computadora
- ▶ **TIPOS DE LENGUAJE:** compilado, interpretado, intermedio
  - ▶ Compilado: convierte el código a binarios que lee el sistema operativo
  - ▶ Interpretado: requiere de un programa que lea la instrucción del código en tiempo real y lo ejecute.
  - ▶ Intermedio: se compila el código fuente a un lenguaje intermedio y este último se ejecuta en una máquina virtual.
- ▶ **LENGUAJES:**
  - ▶ JAVA
  - ▶ PHYTON
  - ▶ C++

# EXPLICACIÓN DEL ENTORNO GRAFICO PSEINT






# CONCEPTOS BASICOS DE PROGRAMACIÓN

- CONSTANTES
  - VARIABLES
  - CONTADORES
  - ACUMULADORES
  - IDENTIFICADORES
- 



# CONSTANTES

- **Constantes:** son aquellos cuyo valor no cambia durante la ejecución del algoritmo- pi, el valor del IVA, tu fecha de nacimiento, tu número de tarjeta de identidad.
- 





# VARIABLES

- **Variables:** elementos cuyo valor puede cambiar durante la ejecución del algoritmo- Nombre, edad, sexo, sueldo, etc.



# CONTADORES

- Un contador 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.
- Contador = contador + constante
- La inicialización consiste en asignarle al contador un valor con el que arrancará el proceso de conteo. Este proceso deberá situarse "antes y fuera del bucle"
- **Bucle:** Conjunto o serie de instrucciones de un programa cuya ejecución se repite hasta que se cumple una determinada condición de salida.



# CONTADORES

## ESTRUCTURAS DE CONTROL ITERATIVAS Acumuladores y contadores

Es muy importante que inicialicemos siempre nuestros contadores

**Veces = 0**

**total = 33**

El valor con el que se inicie el contador, dependerá del problema a resolver.

Veces =	Veces +	1
1	0	1
2	1	1
3	2	1

total =	Total -	3
30	33	3
27	30	3
24	27	3

El valor inicial está marcado en azul y el valor final está en color naranja



# ACUMULADORES

## ► Acumuladores:

Un acumulador es una variable que suma sobre sí misma un conjunto de valores, para de esta manera tener la suma de todos ellos en una sola variable. La diferencia entre un contador y un acumulador es que mientras el primero va aumentando de uno en uno, el acumulador va aumentando en una cantidad variable.

► Sintaxis: `Acumulador = Acumulador + Variable;`

# ACUMULADORES

## ESTRUCTURAS DE CONTROL ITERATIVAS Acumuladores y contadores

Es muy importante que inicialicemos siempre nuestros acumuladores

**nomina = 0**

**total = 350**

El valor con el que se inicie el acumulador, dependerá del problema a resolver.

nomina =	nomina +	sueldo	total =	Total -	deuda
1200	0	1200	300	350	50
1700	1200	500	225	300	75
3200	1700	1500	175	225	150

El valor inicial está marcado en azul y el valor final está en color naranja







# IDENTIFICADORES

- ▶ Los identificadores son nombres que se dan a los elementos utilizados para resolver un problema y poder diferenciar unos de otros. Al asignar nombres (identificadores) a variables, constantes y procedimientos se deben tener en cuenta algunas reglas, como las siguientes:
  - ▶ 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.
  - ▶ Los nombres deben ser nemotécnicos, con solo leerlos se puede entender lo que contienen.
  - ▶ Es conveniente utilizar una sola palabra para nombrar páginas, controles, variables, etc.



# FUNDAMENTOS DE PROGRAMACIÓN PSEINT

- DESCARGAR PROGRAMA
- EXPLICACIÓN DEL ENTORNO GRAFICO PSEINT
- EJEMPLOS DE APLICACIÓN



# DESCARGAR PROGRAMA



FLEXIBLE







sin\_titulo> SUMA A+B.psc x

```

1 Algoritmo sin_titulo
2     Escribir "Ingrese el primer numero:"
3     Leer A
4     Escribir "Ingrese el segundo numero:"
5     Leer B
6     C ← A+B
7     Escribir "El resultado es: ",C
8 FinProceso
9
10
11

```



Algoritmo sin\_titulo

'Ingrese el primer num...

A

'Ingrese el segundo nu...

B

$C \leftarrow A+B$

'El resultado es: ',C

FinAlgoritmo