

# **Variables, Tipos de Datos y Acciones elementales**

Prof. Hilda Contreras

Programación 1

[hildac.programacion1@gmail.com](mailto:hildac.programacion1@gmail.com)

# Algoritmos

**Es la descripción de un esquema expresado con la ayuda de un repertorio finito y bien comprendido de acciones elementales que ejecuta una máquina para resolver un problema**

Conjunto finitos de pasos que resuelve un problema dado.

# Algoritmos

Se pueden expresar en:

- **Lenguaje natural:** nuestro lenguaje pero es ambiguo y poco preciso
- **Seudo código:** notación formal, lenguaje artificial semejante a un lenguaje de programación que *no se compila ni ejecuta*
- **Diagrama:** diagrama de flujo o diagrama de actividades de UML (Lenguaje de Modelado Unificado)

# Seudo código o notación algorítmica

Características:

- Fácil de leer y aprender
- Refleje de forma directa el análisis de la solución
- Tiene poder de expresión: con poco elementos se pueden expresar muchas cosas
- Tiene traducción a cualquier Lenguaje de programación

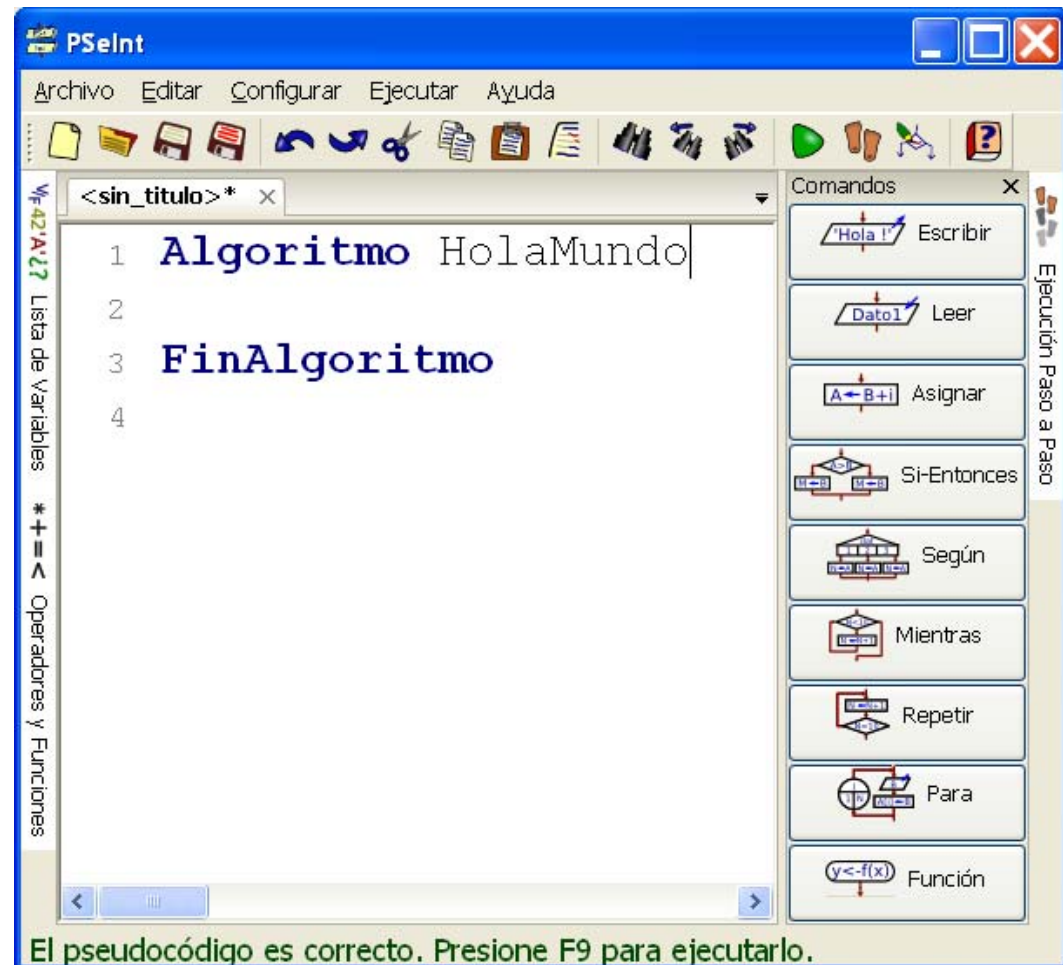
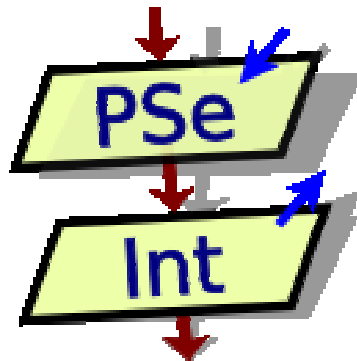
# Notación algorítmica

Necesita expresar:

- Conjuntos de valores que toma la información
- Operaciones sobre los valores
- Acciones elementales
- Composición de acciones

# Notación algorítmica – Herramienta PSeInt

PSeInt



# Contenido del Tema

1. Descripción de los Valores
2. Tipos de datos
3. Acciones elementales

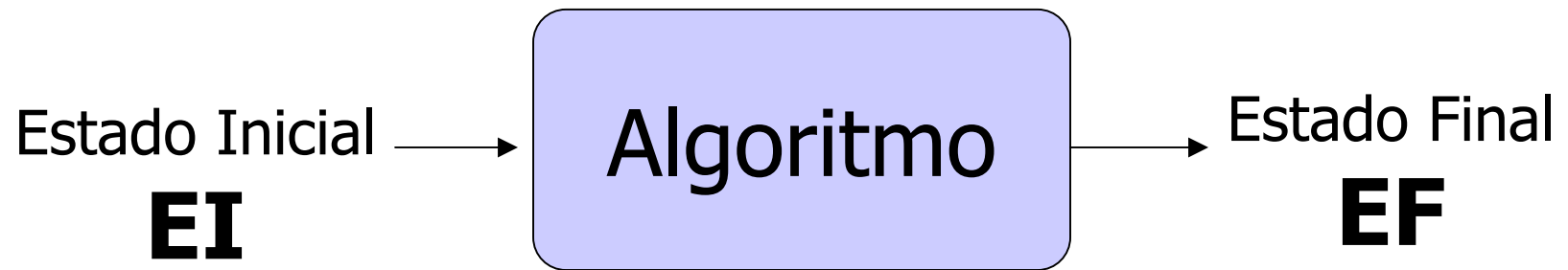
# Descripción de los Valores

Todo problema esta especificado en términos de la información inicial y lo que se desea obtener como resultado

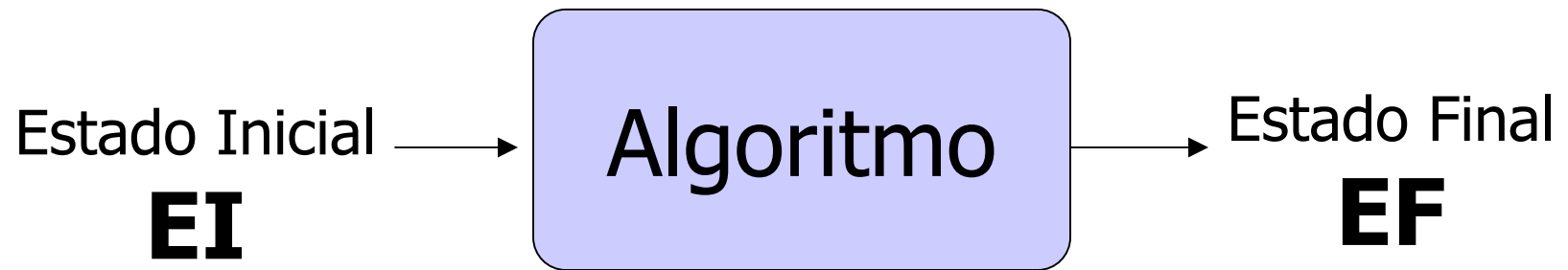
→ Es necesario que el valor de la información de entrada y salida se represente en el algoritmo



# Descripción de los Valores



# Descripción de los Valores



**ENTRADA → PROCESO → SALIDA**

# Descripción de los Valores

El conjunto de valores que puede tomar la información son de 2 tipos:

- **Valores NO estructurados o elementales:** no pueden dividirse en componentes
- **Valores estructurados:** formado por componentes y por lo tanto pueden descomponerse en otros valores

# Valores NO estructurados

- **Enteros:** números positivos y negativos sin parte decimal
- **Reales:** números positivos y negativos que puede tener parte decimal
- **Lógicos o booleanos:** valores de verdad de las proposiciones lógicas
- **Caracteres:** cualquier símbolos que puede escribirse en el teclado

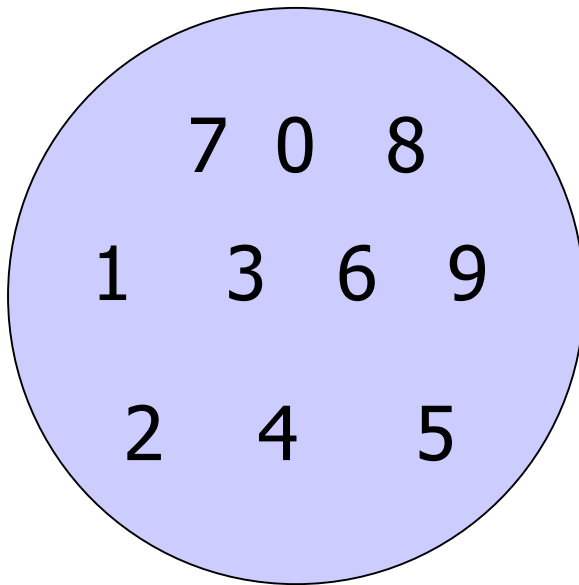
# Valores

Se expresan:

- **Constantes:** denotación de un valor particular del conjunto
- **Nombres o identificadores (Variables):** denotación de un valor genérico del conjunto
- **Expresiones:** denotación de valores a partir de operaciones entre constantes y o nombres

# Valores

CONJUNTO



Variable A  
A denota al 6

Expresiones:

$A + 3$

$A > 9$

$A * 3 - 12$

# Tipo de dato

**Tipo de dato = Valores + Operaciones**

Es el conjunto de valores que una variable puede tomar y las operaciones que pueden aplicarse sobre ellos.

# Valores no estructurados

<b>Tipo</b>	<b>Constantes</b>	<b>Nombres</b>	<b>Expresiones</b>
<b>ENTERO</b>	-1, 7, 0	I, J	$(I + 5) * J$
<b>REAL</b>	3.14, -2.3, 8	H	$5.5 * 3.14 + H$
<b>LOGICO</b>	Verdadero, falso	S, A, B	$S = 3.14, A \text{ o } B$
<b>CARACTER</b>	'A', 'n', '\$'	M	'L'.M



# Acciones elementales

- a) Asociación de un valor con un nombre (nominación de la información)
- b) Entrada simple (copia de un valor externo)
- c) Salida simple (copia **sobre** un dispositivo externo)

# Asociación Valor - Nombre

**Asignación:** asociación que permite modificar el estado de los objetos que conforman el algoritmo para obtener los resultados.

Operador de asignación: <-

Por ejemplo: **A** <- 3

Cambia el valor del nombre que aparece a la izquierda. A toma el valor de 3 o el valor 3 se le da al nombre A

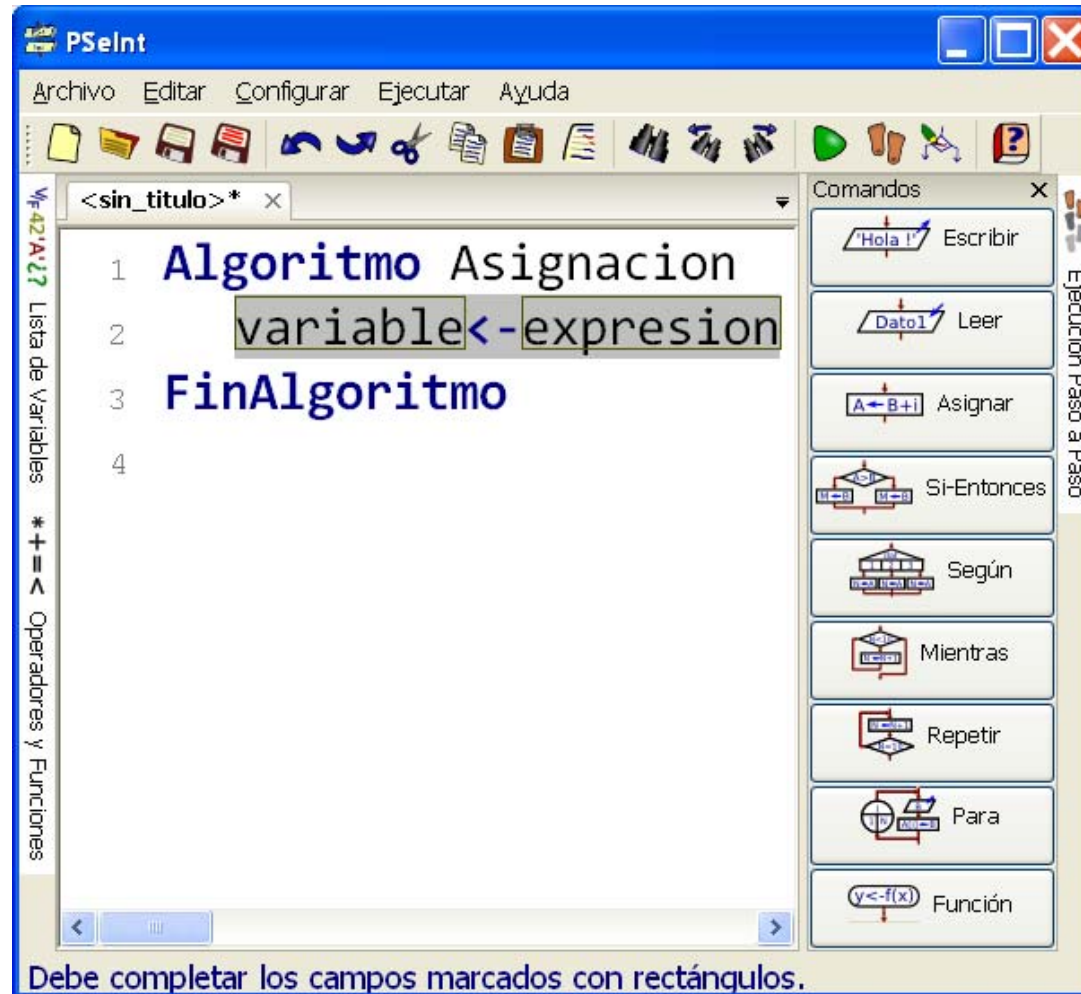
# Asociación, Asignación

Ejemplos:

- **A** <- 3
- **P** <- 3.1415
- **x** <- **y** (X toma el valor nominado por Y)
- **P** <- **P** + 1 (P toma su valor + 1)
- **C** <- **P** = 3.1415 (comparación se asigna el valor lógico)

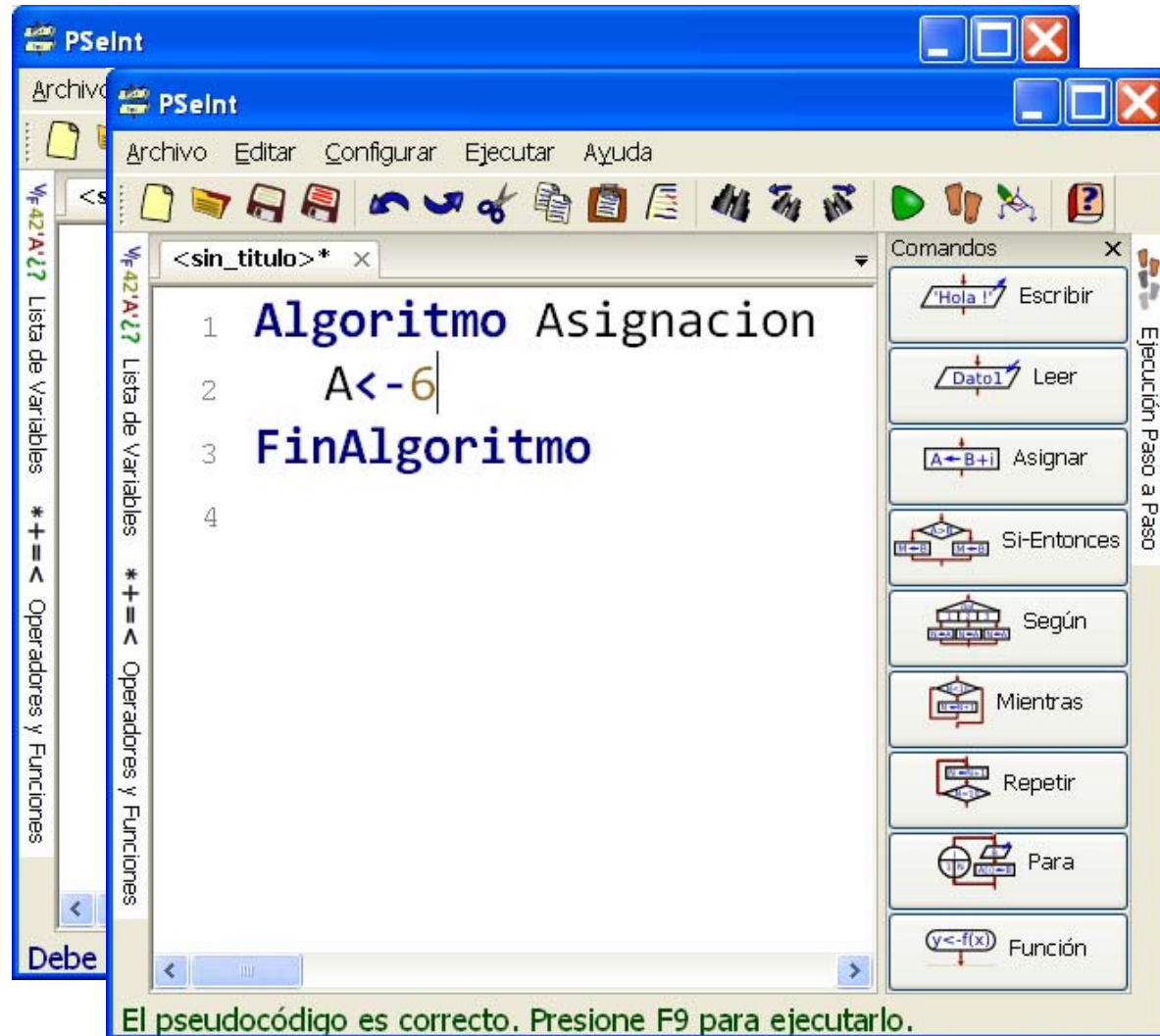
# Asociación, Asignación

PSeInt



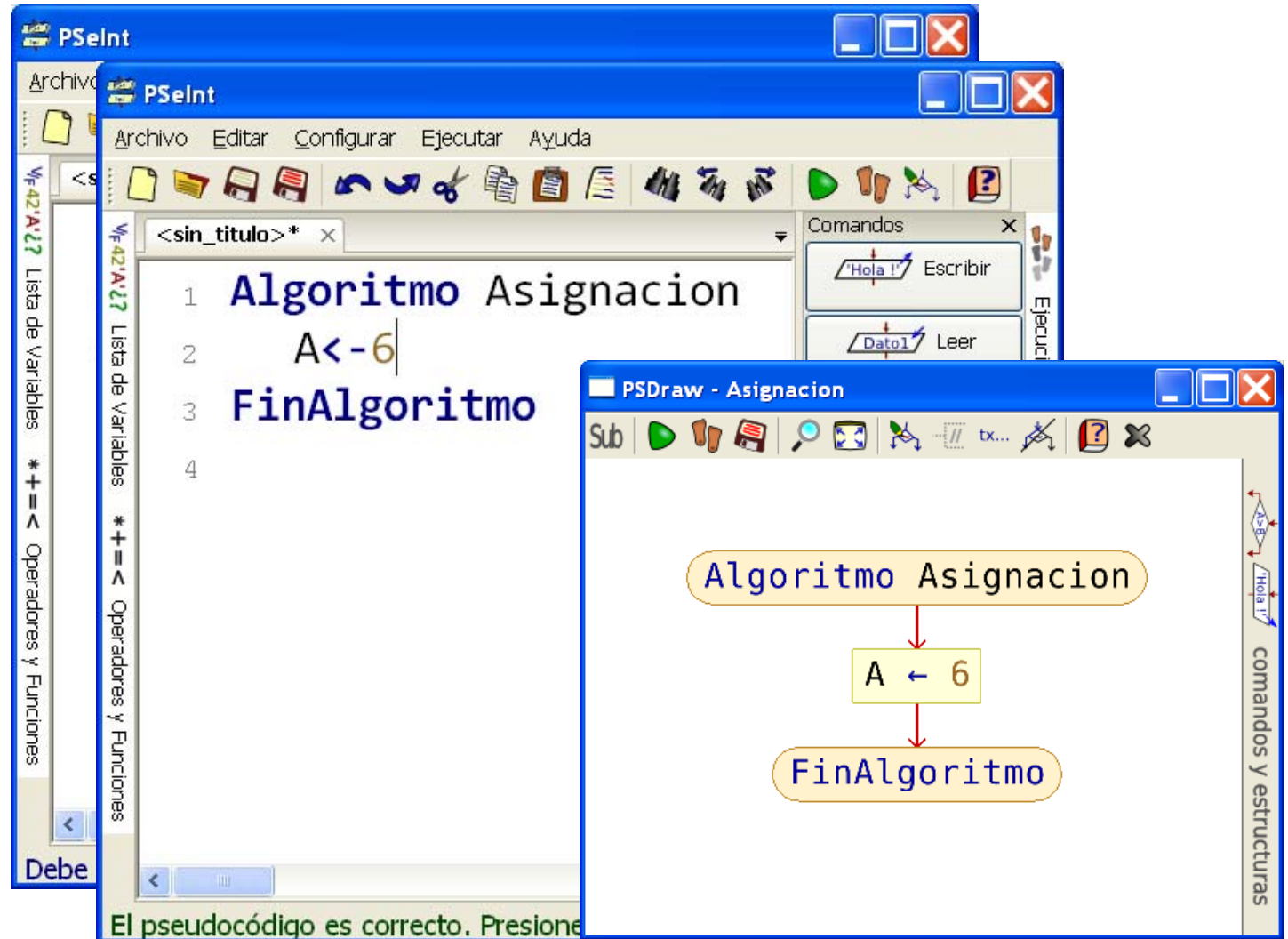
# Asociación, Asignación

PSeInt



# Asociación, Asignación

PSeInt



# Entrada simple

Lectura: permite nominar información que se encuentra en un dispositivo externo. Genera cambio de estado del nombre

Ejemplo:       **Leer A**

Se trata de una asignación que toma la información de un dispositivo externo y se asigna al nombre **A**

# Entrada simple

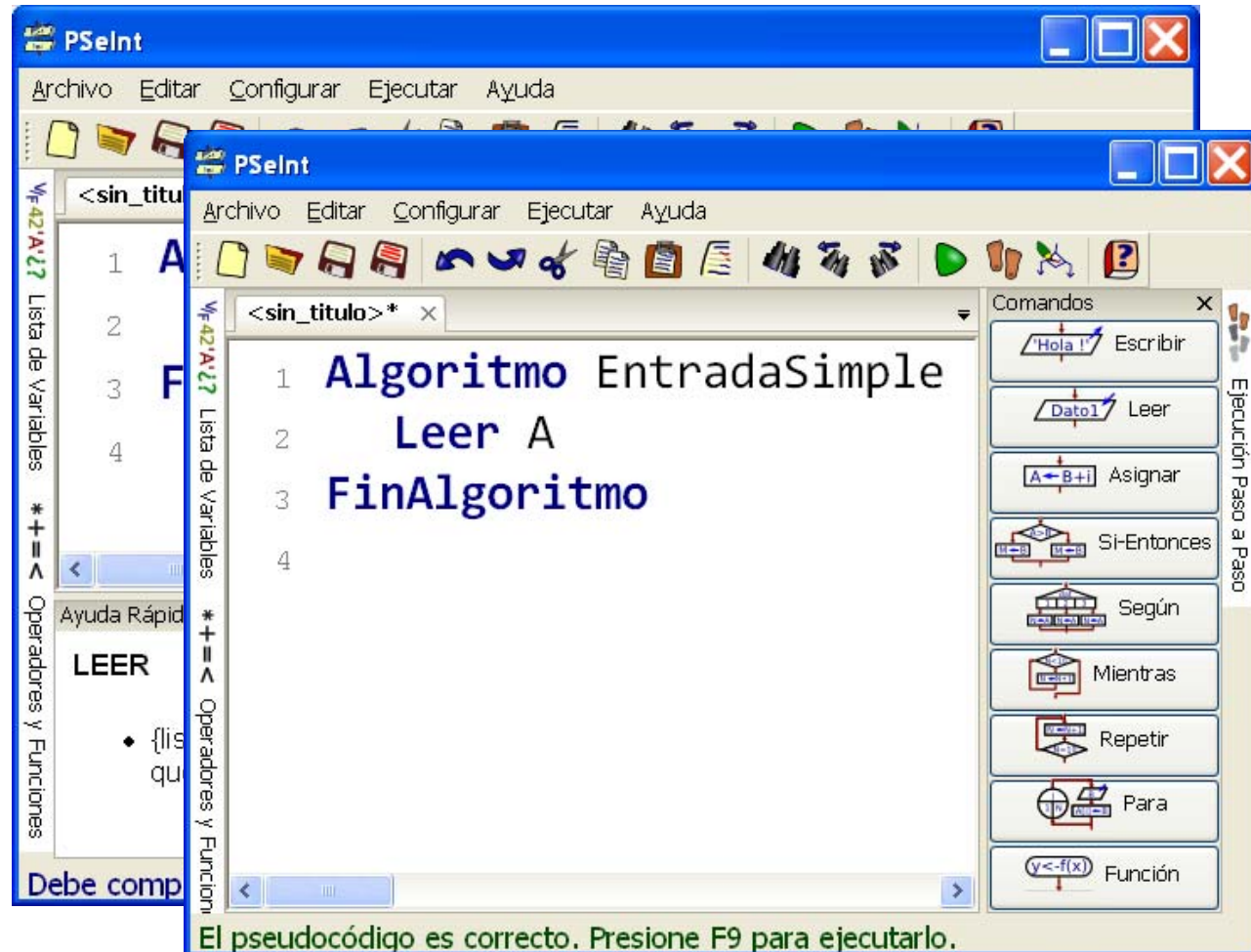
PSeInt



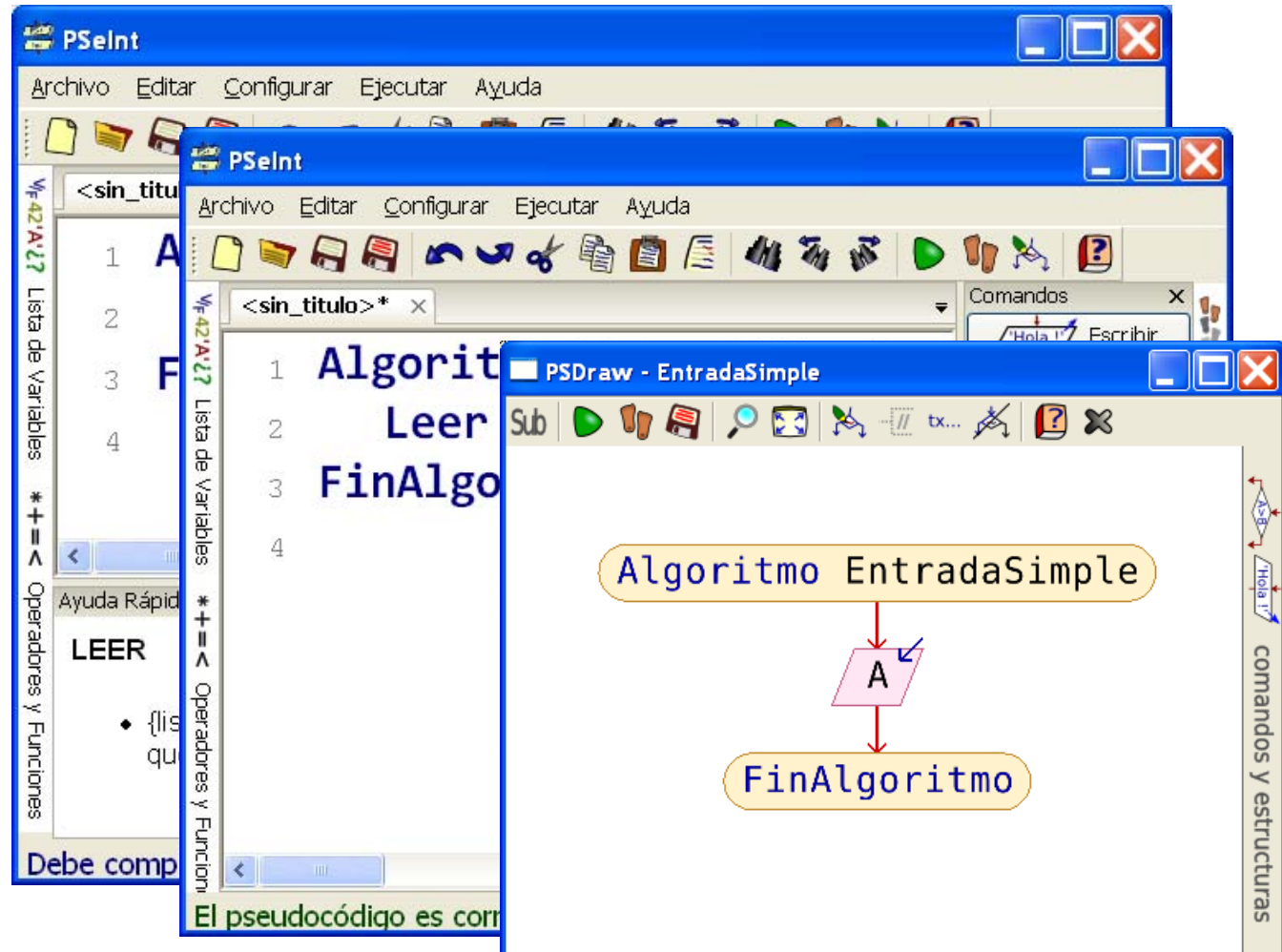


# Entrada simple

PSeInt



# PSeInt



# Salida simple

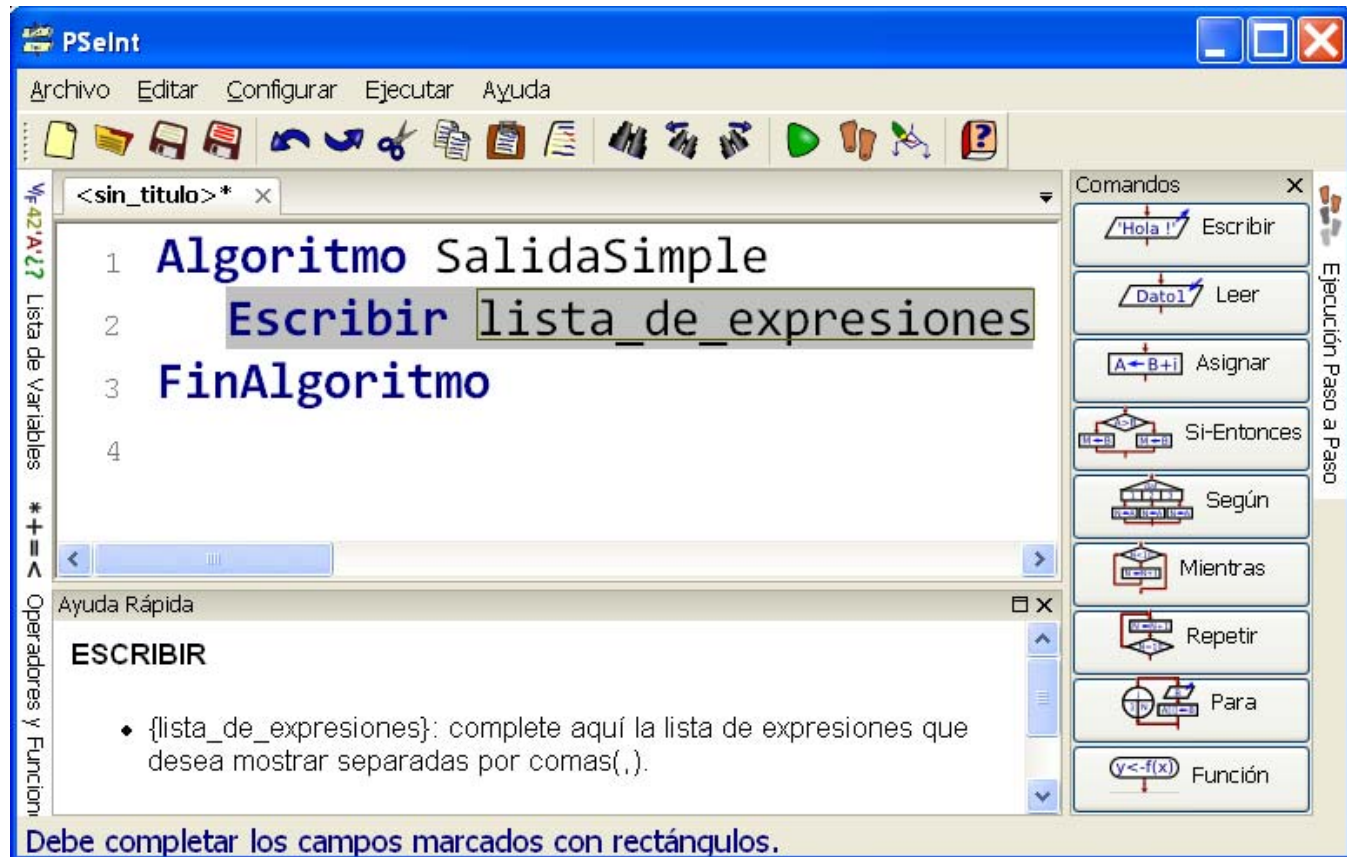
Escribir o Imprimir: acción que permite mostrar los resultados una vez obtenidos

Ejemplo:       **Escribir A**

Indica que sobre algún dispositivo externo se esta copiando el valor del nombre **A**

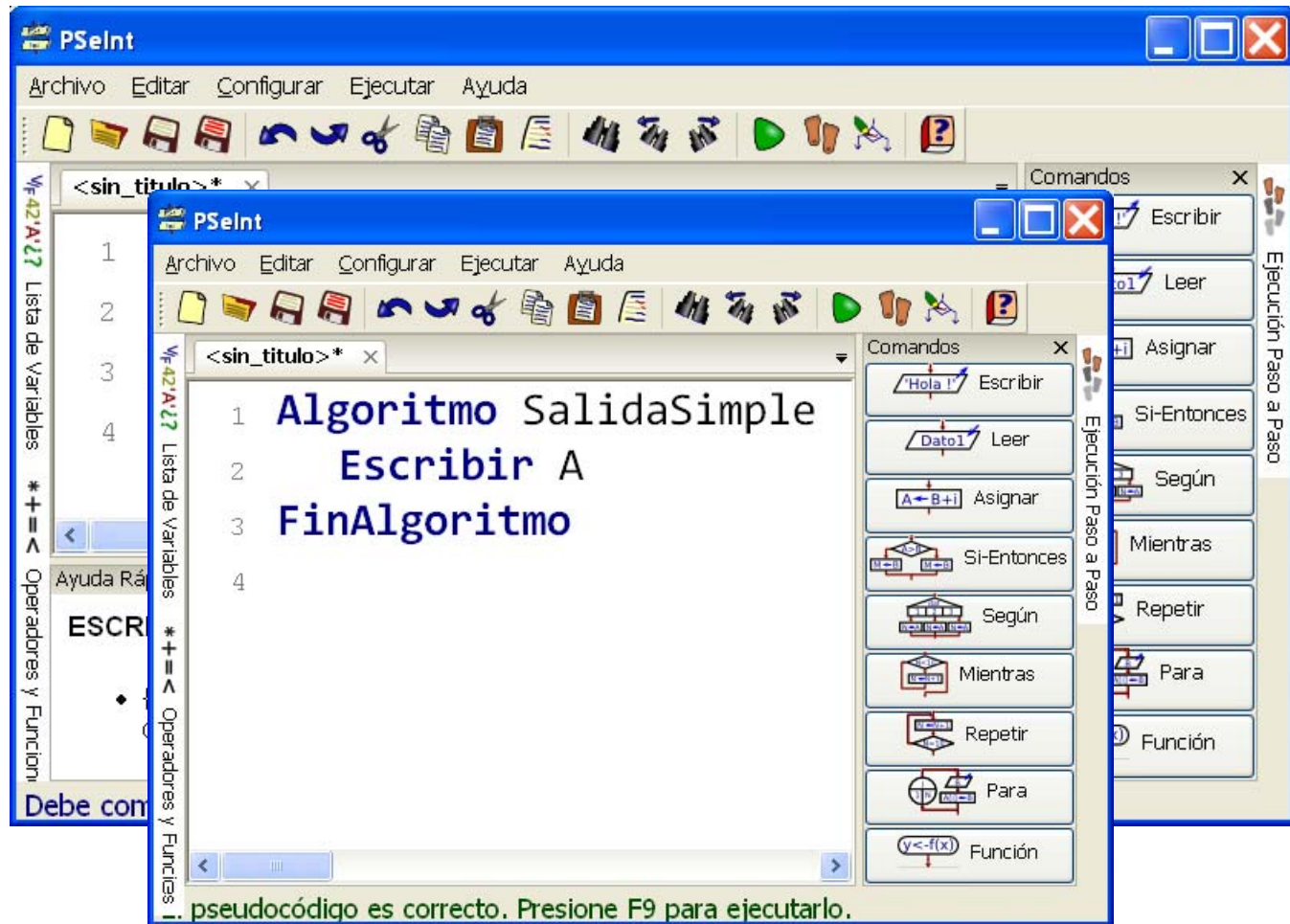
# Entrada simple

PSeInt



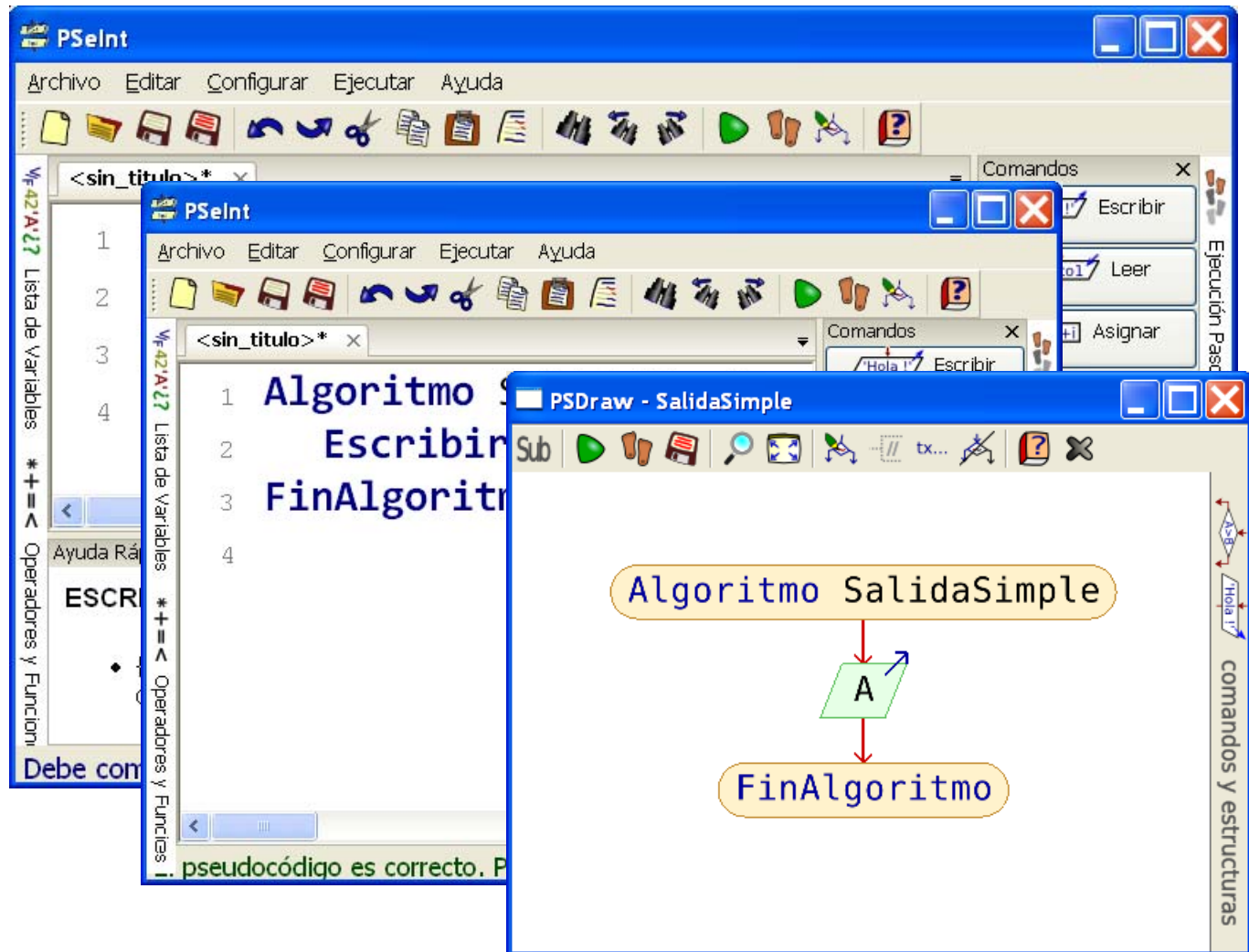
# Entrada simple

PSeInt



# Entrada simple

PSeInt



# TAREA

Consulte en Internet como se realizan las acciones elementales de la asignación, salida simple y entrada simple en:

- Lenguaje de programación C
- Lenguaje de programación que quiera aprender

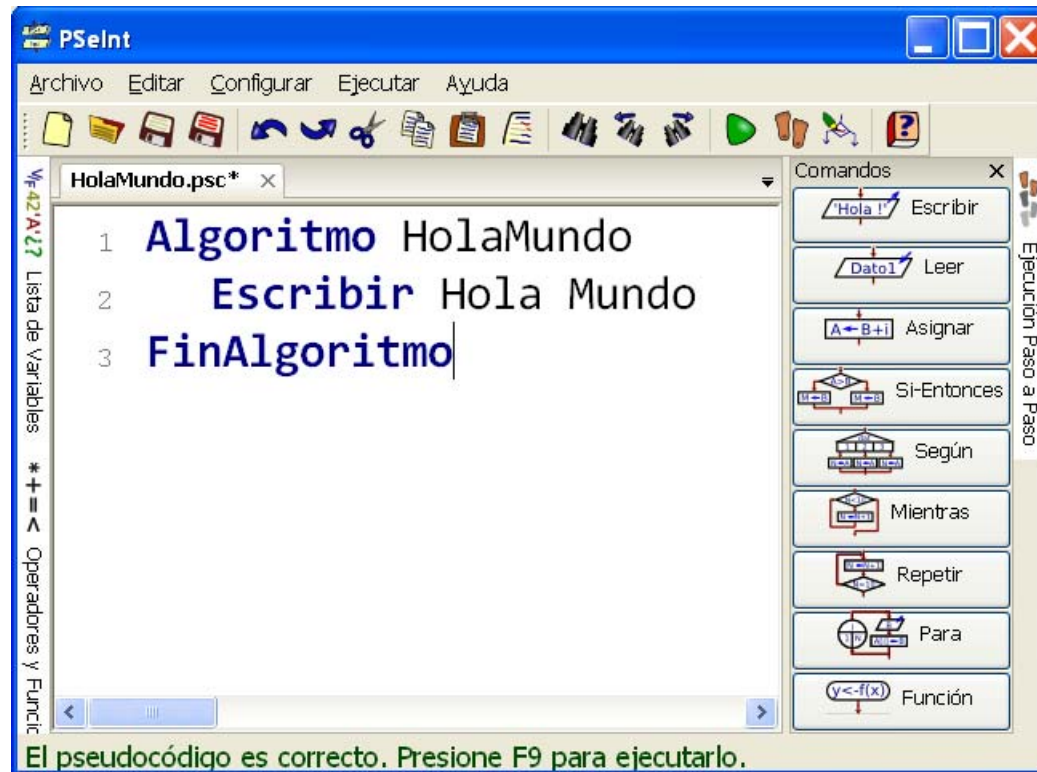
Escriba su primer programa HolaMundo en Lenguaje C, que escriba en pantalla "Hola Mundo".

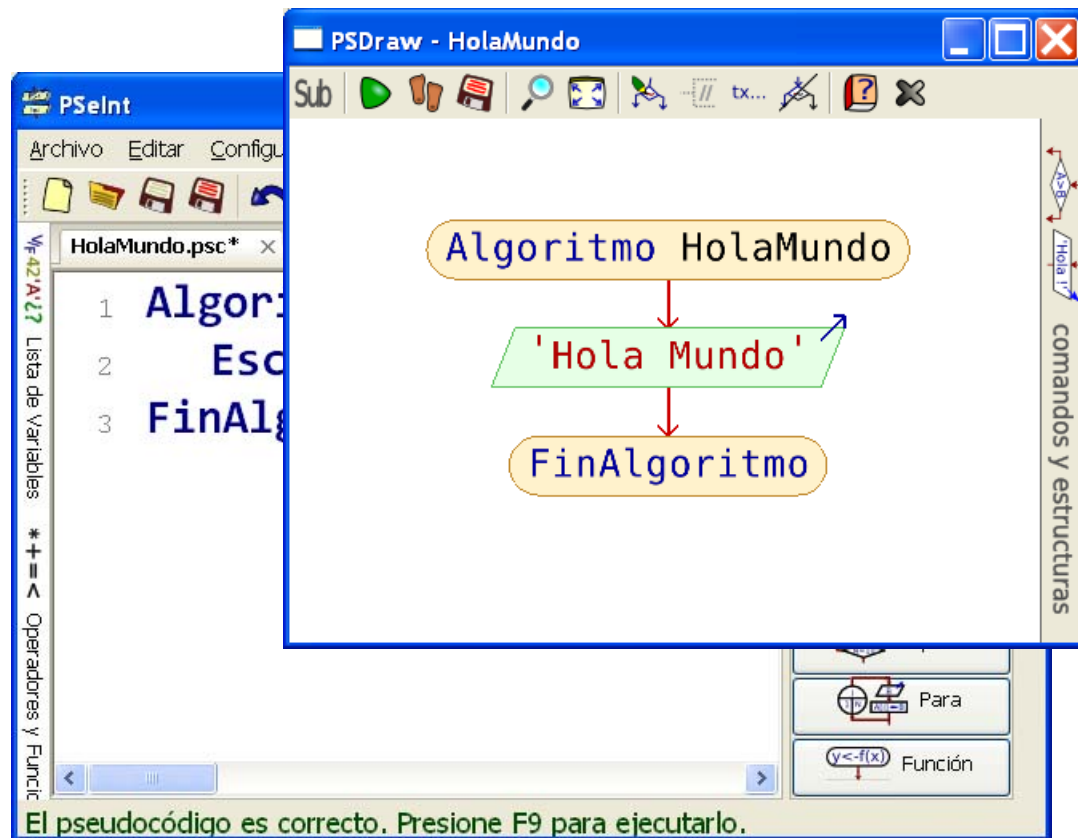


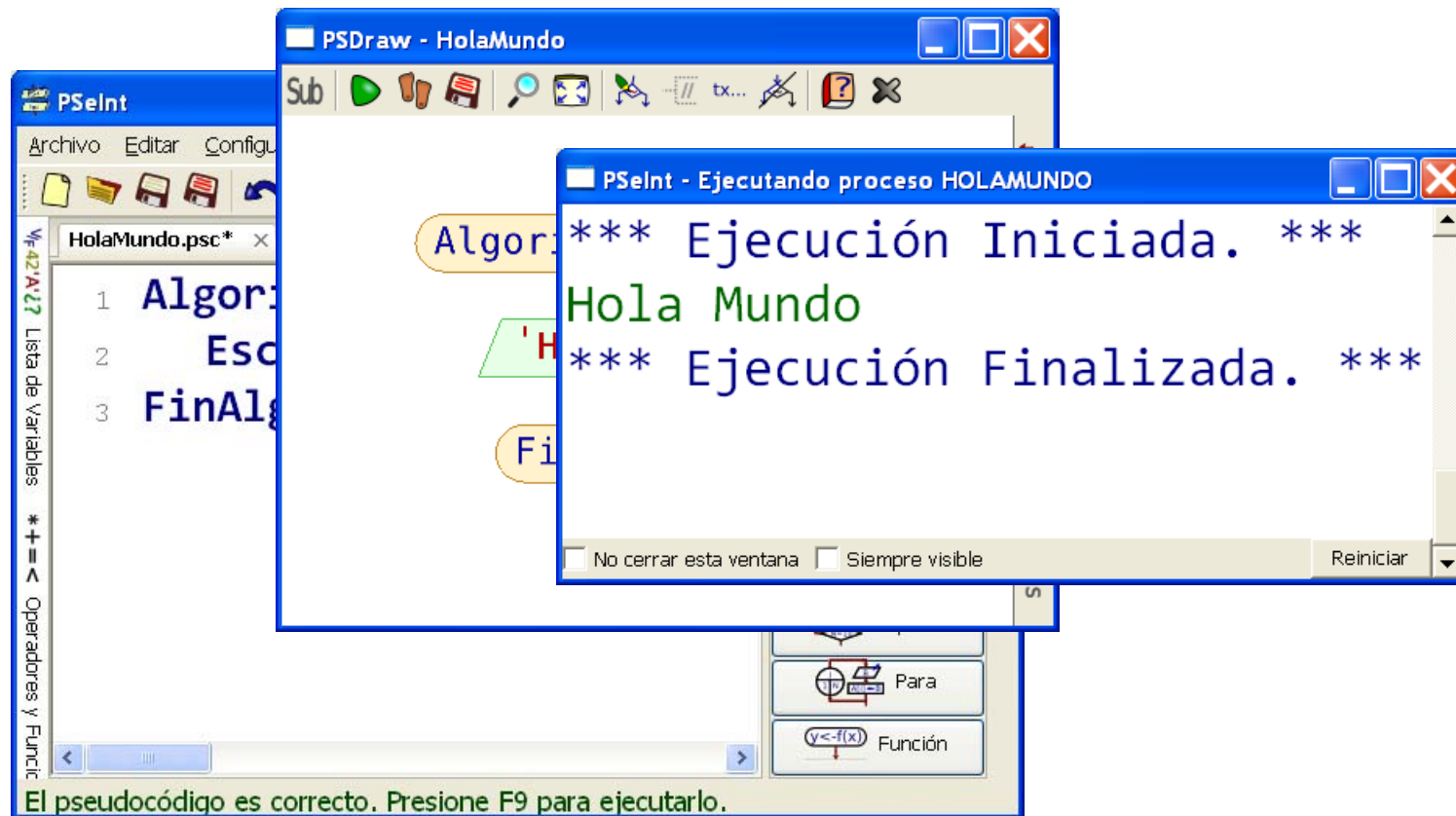
# Hola Mundo

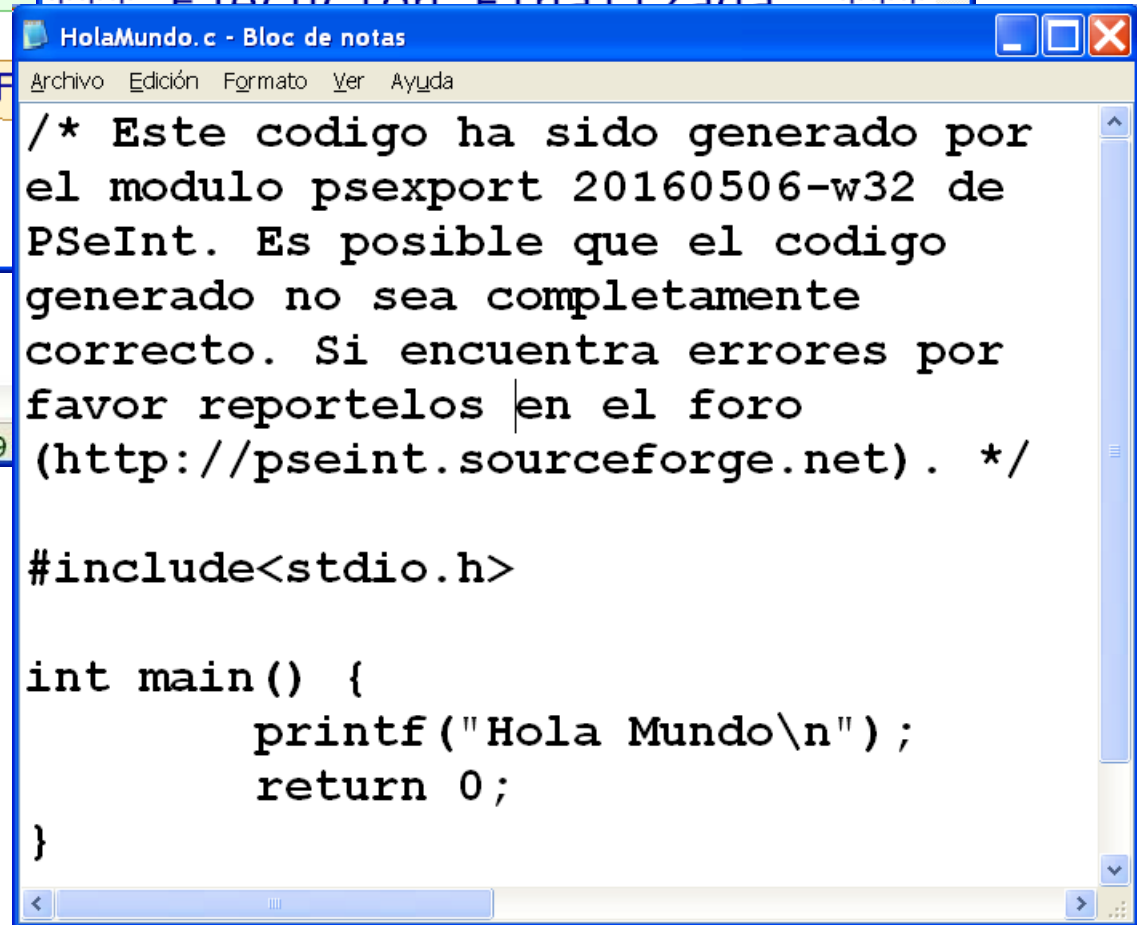
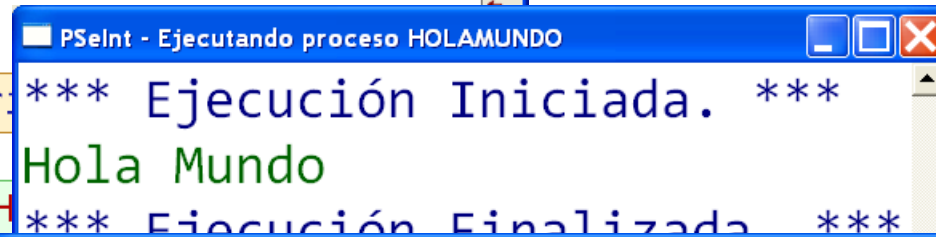
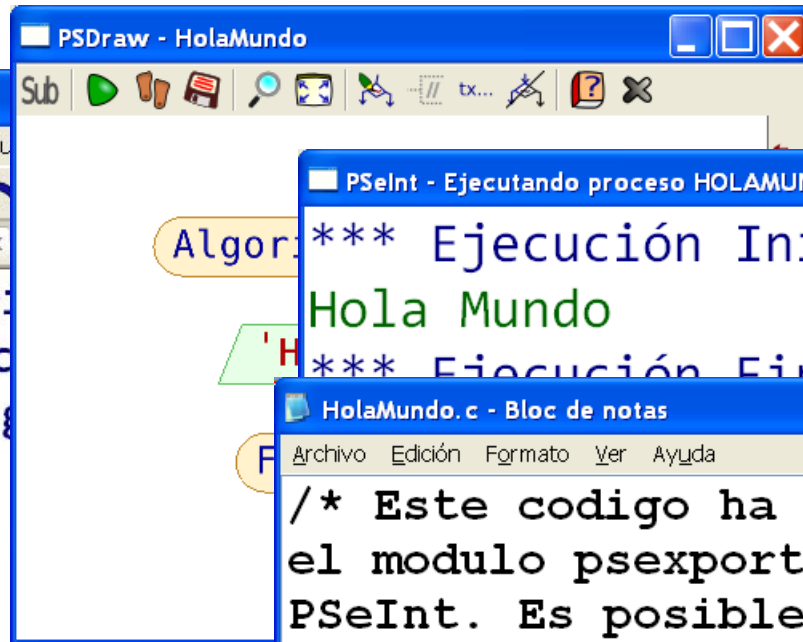
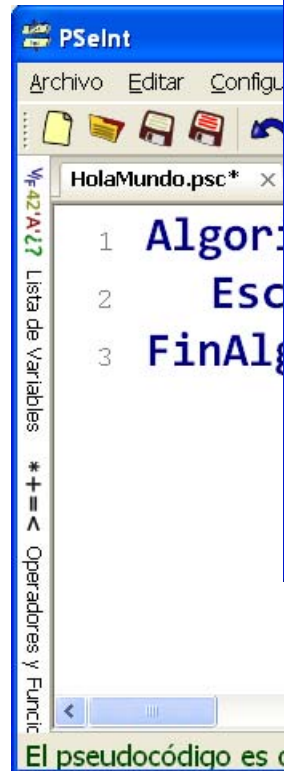
- Entender el problema del HolaMundo
- Intención es superar la barrera técnica del Lenguaje de Programación:
  - Instalación
  - Cómo escribir en el Lenguaje, lo mínimo que se necesita para un programa
  - Cómo Compilar o Interpretar
  - Cómo ponerlo a funcionar











# Compilar y Ejecutar en Lenguaje C

> **gcc** HolaMundo.c

> ./a.out

> **gcc** HolaMundo.c -o hola

> ./hola

# Compilación Lenguaje C



# Compilación Lenguaje C



**gcc** HolaMundo.c -o hola

Debe saber que cada lenguaje de programación es diferente y la notación algorítmica (pseudo código) permite ***abstraernos*** de estos detalles de implementación.



*"... los lenguajes o notaciones  
que usamos para expresar o  
grabar nuestros pensamientos  
son los factores que  
determinan lo que podemos  
pensar o expresar..."*

**E. Dijkstra. El Humilde  
Programador (1972)**