

A graphic on the left side of the slide features four stacked, 3D-style rectangular blocks in purple, orange, yellow, and blue. An orange arrow points to the right, passing through the middle of the blocks.

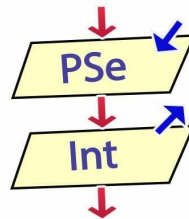
Agencia de
Aprendizaje
a lo largo
de la vida

CODO A CODO INICIAL

Clase 2

Algoritmia 2

Tipos de dato y PSeInt



Les damos la bienvenida

Vamos a comenzar a grabar la clase

Clase 01

Algoritmia 1 - Introducción a la Algoritmia y Diagramas de Flujo

- Introducción al cómputo.
- Hardware y Software.
- Definición de datos, algoritmos y programas
- Características de los algoritmos.
- Diagramas de flujo.
- PSeInt

Clase 02

Algoritmia 2 - Tipos de datos y PSeInt

- Conceptos básicos de programación: variables y tipos de datos simples.
- Interfaz de PSeInt: Configuración, menús y vistas.
- Aplicación de variables en pseudocódigo.

Clase 03

Algoritmia 3 - Variables, Operadores y Salida de Datos

- Salida de valores por pantalla.
- Operadores aritméticos: suma, resta, multiplicación y división. Jerarquía.
- Operador de concatenación.

Tipos de datos simples

En programación, los **tipos de datos simples** (también llamados *tipos de datos primitivos* o *básicos*) son aquellos que representan valores únicos y no descomponibles en componentes más elementales.

Estos tipos de datos proporcionan la base para construir estructuras de datos más complejas. Los tipos de datos simples suelen ser directamente compatibles con las operaciones de bajo nivel de la máquina subyacente.

Tipos de datos

El **tipo de dato** indica a la computadora la naturaleza del valor que debe procesar. Se puede definir qué clase o rango de valores acepta y qué operaciones se pueden realizar con ese valor. Los tipos comunes son:

- **ENTEROS** (números enteros): por ejemplo, 12, 24, 65, 1044.
- **REALES** (números decimales y fraccionarios): 1,25; $\frac{1}{4}$; 3,1415; 1,42856.
- **CARACTERES** (alfanuméricos + símbolos): “Hola, Mundo!”; “A4YNm@34”.
- **LÓGICOS** (Verdadero o Falso / True or False): también llamados ‘booleanos’, pueden ser Verdadero o Falso.

Tipos de datos | Numéricos

Si el dato es exclusivamente un **número** y se pretende operar con él como tal, entonces su tipo de dato es **numérico**. Estos pueden ser, a su vez, positivos, negativos, decimales o enteros.

Los datos del tipo numérico pueden sumarse, restarse, multiplicarse, etcétera.

Según qué lenguaje de programación usemos, el tipo de dato puede estar discriminado en corto o largo (para los enteros), coma flotante simple o coma flotante doble (para los decimales), información que se usará para disponer el espacio en memoria que deberá reservarse.

Tipos de datos | Caracteres (alfanuméricos)

El **carácter** es un dato alfanumérico: puede ser una letra, un símbolo o un número entre comillas. Cuando agrupamos más de un carácter se forma una cadena de texto, una **cadena de caracteres**.

“Hola”, “asd@3”, “42”, “Argentina es campeón” son todos ejemplos de tipo de dato carácter. En el caso del “42”, lo que hace que esto sea de tipo carácter y no numérico es el uso de las comillas para enmarcarlo: aquí, eso no es un número, sino una cadena de caracteres que, coincidentemente, representa un número en forma de texto.

Tipos de datos | Lógicos (booleanos)

Este tipo de dato puede **representar sólo dos valores**, como pueden ser Verdadero o Falso, 1 o 0.

Son normalmente usados para representar las alternativas de determinadas condiciones que pueden resultar verdaderas o falsas.

Ejemplo: la expresión booleana $|8 < 4|$ daría Falso como resultado, dado que 8 no es menor que 4 y la expresión $|8 < 9|$, en cambio, dará Verdadero.

En esencia, nos permiten convalidar si una determinada situación se ha dado o no durante la ejecución de nuestro programa, para tomar una decisión en función de eso.

Variables | ¿Que son?

Es un espacio en la **memoria RAM**, que permite almacenar un dato específico de manera **temporal**.

Se puede imaginar a las variables como “*cajitas temporarias*”, en las que se almacenan individualmente cada dato necesario, para luego ser procesado.



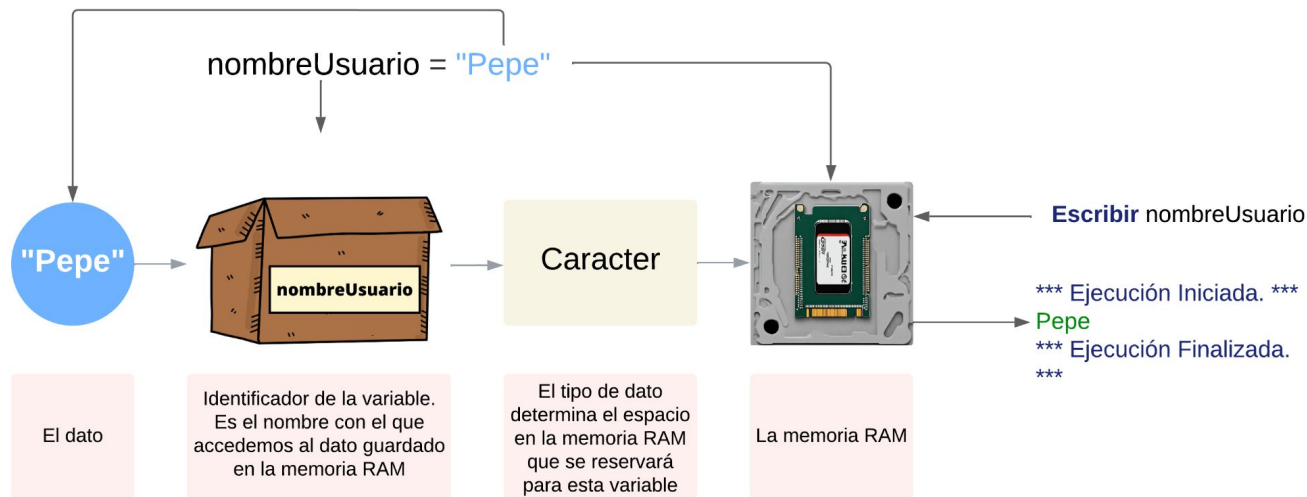
Variables | Características

Las variables poseen tres características principales:

- **Asignación de memoria:** cada variable tiene su propio espacio reservado en la memoria RAM.
- **Declaración:** tienen un nombre que las identifica (único, y que cumple con determinadas reglas) y posee un dato de un tipo determinado.
- **Alcance:** es el ámbito donde la variable es accesible.

Variables | Asignación de memoria

La computadora le reserva a la variable un espacio en su memoria, en función del tipo de dato que vaya a almacenar.



Variables | Declaración

- Mediante este proceso, **PSeInt** le asigna un **nombre** y un **valor** a la variable. De esta forma, el sistema asigna una posición de la memoria a la variable, a la que accedemos por su nombre para almacenar o recuperar el dato que contiene.
- Hay lenguajes en los que se debe explícitamente declarar el **tipo de dato** que almacena la variable antes de poder usarla (ej.: Java) y otros en los que no hace falta (ej.: Python).

Variables | Declaración

Si bien con la configuración flexible de PSeInt no es obligatorio definir una variable antes de asignarle un valor, igualmente podemos hacerlo:

```
Definir cantidad Como Entero  
Definir nombre Como Caracter  
Definir altura Como Real  
Definir activo Como Logico
```

Cómo vimos, si no lo hacemos, PSeInt identificará el tipo de dato de manera dinámica, en función del valor que le asignemos.

Variables | Alcance

En función de dónde sean creadas y/o declaradas, las variables pueden clasificarse en **variables locales** y **variables globales**.

- **Globales:** son accesibles a todo lo largo del algoritmo, sin limitaciones.
- **Locales:** son accesibles sólo dentro del ámbito (scope) en el que fueron declaradas y/o creadas; por ejemplo, dentro de una función. Las variables locales existen y dejan de existir en el ámbito en el que fueron creadas.

Variables | Reglas para su nombre

El nombre que identifica a una variable debe cumplir con determinadas reglas, que suelen variar de un lenguaje a otro. PSeInt, y muchos otros lenguajes, posee los siguientes requisitos para los nombres de las variables:

- Debe comenzar con una letra
- Puede contener letras, números y guiones bajos
- No puede llamarse igual que una palabra reservada del lenguaje
- No debe contener caracteres especiales (“ñ”, acentos, etc.)

Variables | Nombres válidos

Los ejemplos que tienen la tilde son válidos, mientras que los que tienen la cruz son ejemplos de nombres que no se pueden usar en PSeInt.

✓ edad

✓ nombre_usuario

✗ año

✓ miCarrito2

✗ Calle+numero

✗ 99letra

Variables | Convenciones para el nombre

En variables cortas, de una sola palabra, se recomienda definir el nombre siempre en minúsculas, incluso su primera letra. Cuando el nombre está formado por más de una palabra, existen diferencias convenciones:

- **Camel Case** (o *Lower Camel Case*): nombreVariable
- **Pascal Case** (o *Upper Camel Case*): NombreVariable
- **Snake Case**: nombre_variable (la usamos para el nombre del algoritmo)
- **Kebab Case**: nombre-variable

Para los nombres de variable en PSeInt, usaremos **Camel Case**.

Tipado en los lenguajes

Los lenguajes de programación pueden ser más o menos laxos respecto de los tipos de dato. Existen los que son:

- **Fuertemente tipados:** no podemos operar entre distintos tipos de dato ("23"+4 da error). [Ej.: Java, C++, Python]
- **Débilmente tipados:** podemos operar entre distintos tipos de dato ("23"+4 da "234"). [Ej.: JavaScript]
- **De tipado estático:** se define el tipo de dato antes de usar la variable. [Ej.: Java, C++]
- **De tipado dinámico:** se fija el tipo automáticamente al asignarle un valor. [Ej.: Python, JavaScript]

Variables | Ejemplos

En este ejemplo no sólo podemos ver nuevamente la asignación de valores a variables, sino también la aplicación de la convención Camel Case para sus nombres:

```
pais = "Argentina" // Caracter  
poblacionEnMillones = 45.81 // Real  
anioIndependencia = 1816 // Entero  
hablaHispana = Verdadero // Lógico
```

¿Qué es una constante?

Si consideramos una variable como una "cajita" que puede almacenar un valor que puede cambiar durante la ejecución del programa, entonces podríamos decir que una constante es una "cajita" que tiene un **valor fijo** que **no cambia** durante la ejecución del programa.



PseInt posee las constantes **PI** ($\pi = 3.14159265359$) y **Euler**, el valor base de los logaritmos naturales ($e = 2.71828182846$)

Operador de asignación

En programación, el **operador de asignación** se encarga de asignar a la variable que se encuentra a su izquierda el valor resultante de evaluar la expresión que se encuentra a su derecha.

En **PSeInt** usamos el signo “=” (igual) como operador de asignación.

Operador de asignación

Cuando se asigna un valor a una variable se reemplaza su valor anterior o su valor inicial por uno nuevo.

```
variable = "valor"
```

```
variable = "otro valor"
```



Inicialización y reasignación

A la acción de asignar por primera vez un valor a una variable se le llama **inicializar la variable**.

```
miVariable = "primer valor almacenado"
```

Si más adelante modificamos ese valor almacenado, le llamaremos **reasignar**.

```
miVariable = "nuevo valor almacenado"
```


PSelnt

PSelnt (abreviatura de "Pseudo-Intérprete") es un entorno de desarrollo integrado (IDE) y un **intérprete de pseudocódigo**.

PSelnt utiliza un pseudocódigo estructurado que se asemeja a un lenguaje de programación real y proporciona un ambiente gráfico donde **los usuarios pueden diseñar, escribir y ejecutar algoritmos** de manera visual y educativa. Aunque no es un lenguaje de programación completo, PSelnt sirve como una herramienta didáctica para introducir conceptos de programación antes de avanzar hacia lenguajes de programación más complejos.

PSeInt | Descarga

PSeInt se puede descargar desde [este enlace](#).

Al ingresar, se presentan opciones como las de la derecha. Descargá la versión correspondiente a tu sistema operativo y ejecuta el instalador.

A continuación, te mostramos cómo instalar paso a paso PSeInt en Windows.



Descargar PSeInt para GNU/Linux (tgz - 9.7MB)

[Ver todas las opciones de descarga para GNU/Linux](#)



Descargar PSeInt para Microsoft Windows (exe - 9.7MB)

[Ver todas las opciones de descarga para Microsoft Windows](#)



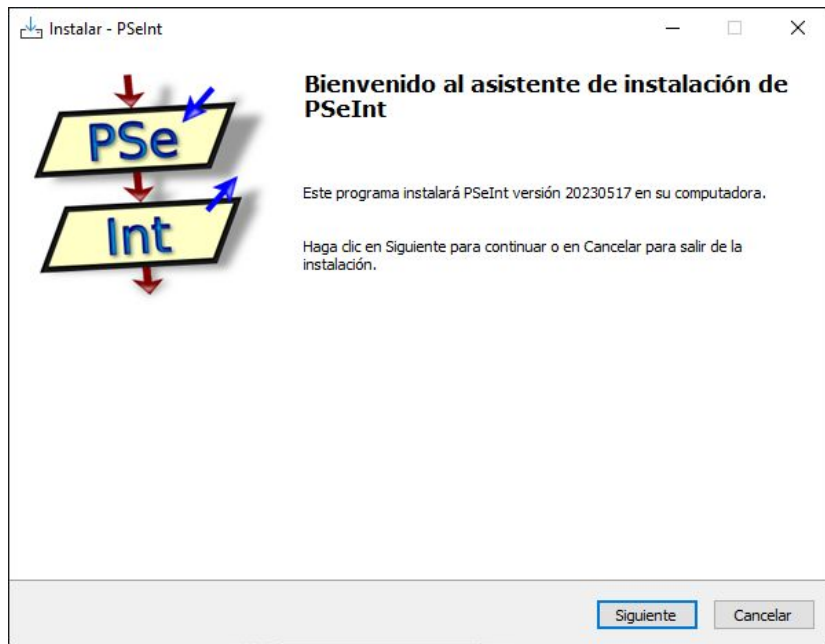
Descargar PSeInt para macOS (tgz - 8.7MB)

[Ver todas las opciones de descarga para macOS](#)

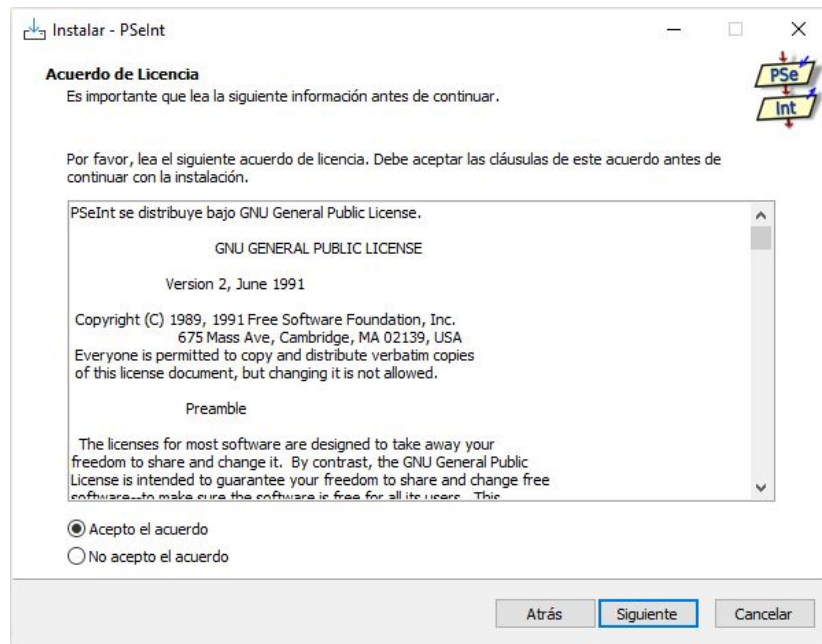
[Ver descargas de la versión anterior](#)

PSeInt | Instalación

Hacer click en siguiente.

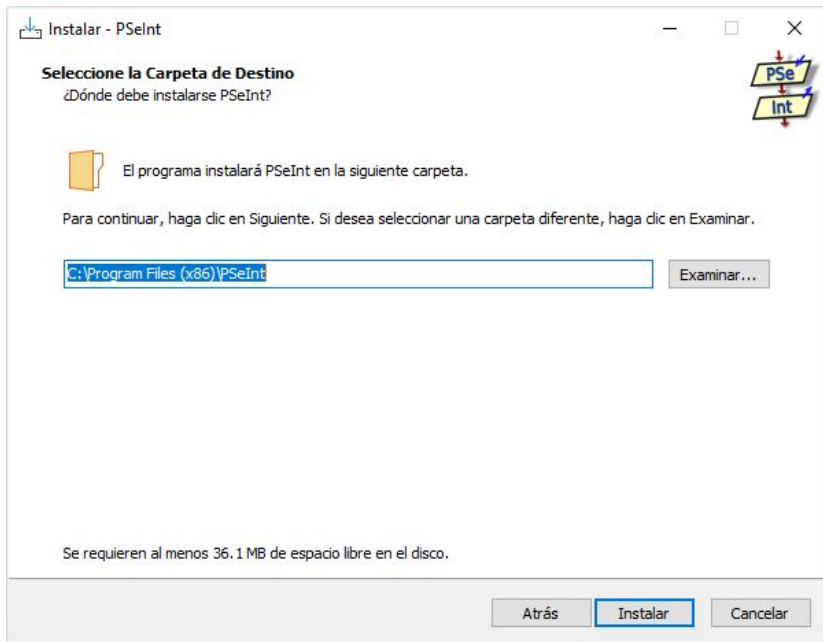


Aceptar el acuerdo.

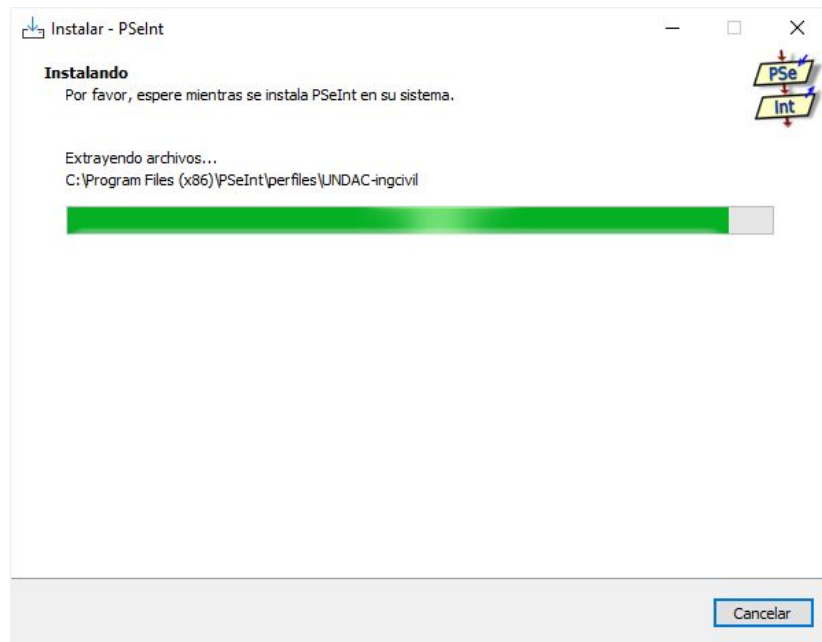


PSeInt | Instalación

Hacer click en Instalar.

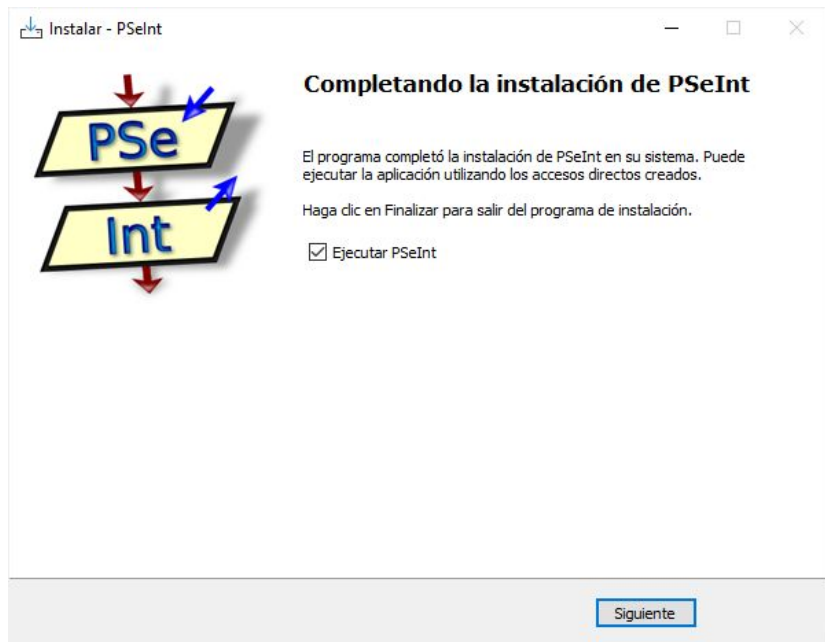


Esperar a que finalice la instalación.



PSeInt | Instalación

Tildar “Ejecutar PSeInt”. Apretar Siguiente.



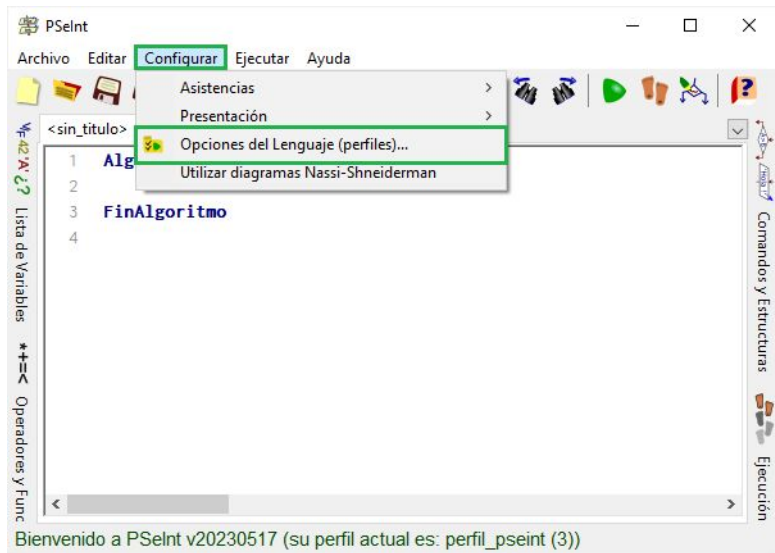
En este punto, si todo ha salido bien, PSeInt está instalado en tu computadora.

Si dejaste marcada la opción “*Ejecutar PSeInt*”, el programa arrancará automáticamente.

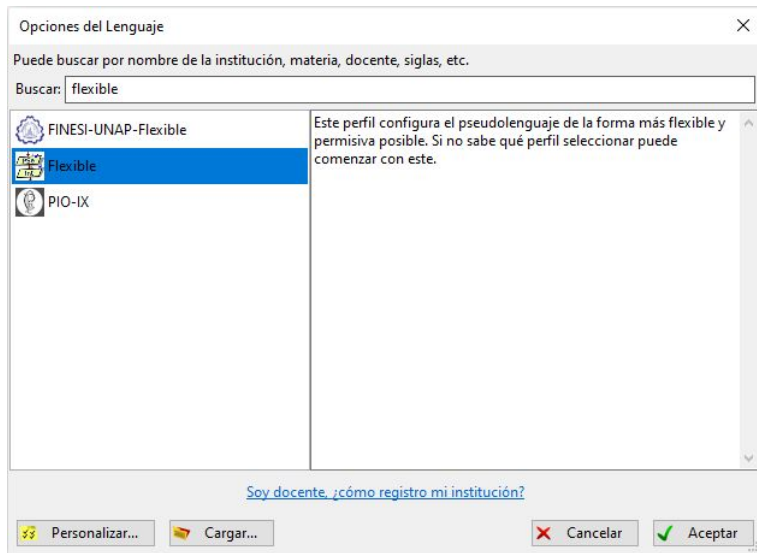
A continuación, te mostramos cómo configurarlo (ya casi terminamos).

PSeInt | Configuración

Ir al menú Configurar y luego a Opciones del Lenguaje (perfiles). Hacer click.

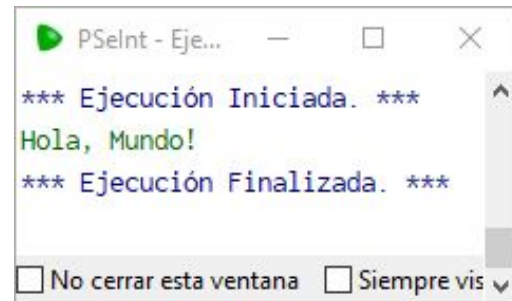
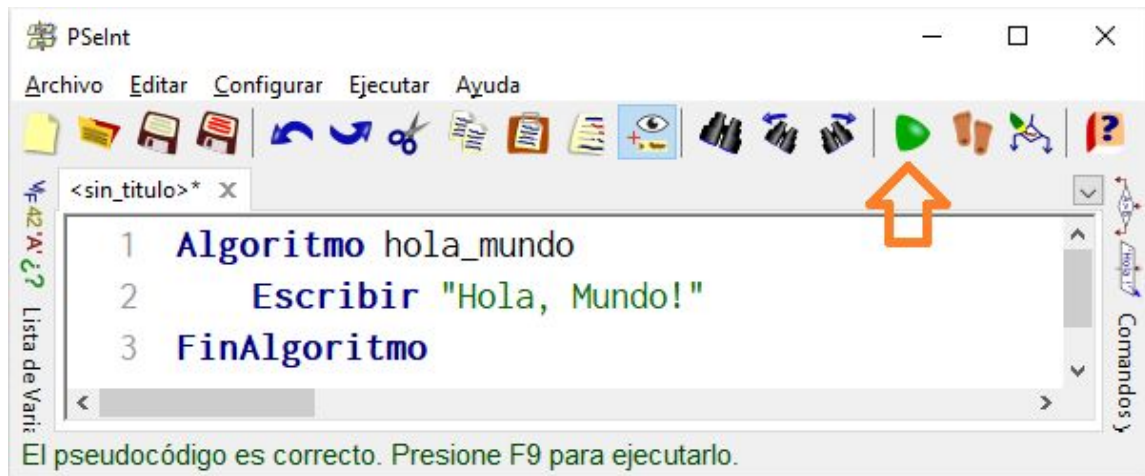


Escribir “flexible” en el buscador, seleccionar “Flexible” en los resultados y hacer click en Aceptar. La configuración ha finalizado.



“Hola, Mundo!” en PSeInt

Para realizar nuestro primer algoritmo, entre *Algoritmo* y *FinAlgoritmo* poné **Escribir “Hola, Mundo!”** y luego hacé click en el botón verde (es el botón que ejecuta el algoritmo). Se abre la consola de la derecha.



PSeInt | Ejemplos de asignación

Veamos la **asignación** a distintas variables de todos los tipos de dato que conocemos: Caracter, Entero, Lógico y Real:

```
Algoritmo asignacion
    nombre = "Enzo"
    nivel = 5
    estaConectado = Verdadero
    puntaje = 183.79
FinAlgoritmo
```


PSeInt | Tipado dinámico

Cuando se asigna un valor a una variable, PSeInt detecta automáticamente su tipo de dato, ya que es un lenguaje de tipado dinámico.

Nombre de la variable	Dato almacenado	Tipo de dato
numEntero	2022	Entero
numReal	7.142857	Real
nombreUsuario	"Lionel Messi"	Caracter
cursoAprobado	Verdadero	Lógico

Advertencia

Si la variable ya fue inicializada, el valor que se le reasigne debe ser compatible con el tipo de dato que se le asignó previamente.

En el siguiente ejemplo, se intenta reasignar una cadena de caracteres a una variable que fue inicializada con un número entero. Pero la variable, en este caso, ya había sido definida como de tipo Entero, entonces **su tipo no puede modificarse**.

```
var1 = 27
```

```
var1 = "hola"
```

en inst. 1: No coinciden los tipos.

```
Escribir var1
```

Material extra

Artículos de interés

Material extra:

- [Ejemplos de algoritmos](#) en PSeInt

Videos:

- [Instalación de PSeInt](#) | Danisable Programacion

No te olvides de dar el presente

Recordá:

- Revisar la Cartelera de Novedades.
- Hacer tus consultas en el Foro.
- Realizar el Ejercicio de Repaso.

Todo en el Aula Virtual.

Muchas gracias por tu atención.

Nos vemos pronto