

# Tutorial PSeInt

Herramienta que permite introducir a la programación



Buenos Aires Ciudad



Vamos Buenos Aires

## Prólogo

Este tutorial se enmarca dentro de los lineamientos del Plan Integral de Educación Digital (PIED) del Ministerio de Educación del Gobierno de la Ciudad Autónoma de Buenos Aires que busca integrar los procesos de enseñanza y de aprendizaje de las instituciones educativas a la cultura digital.

Uno de los objetivos del PIED es "fortalecer el conocimiento y la apropiación crítica de las Tecnologías de la Información y de la Comunicación (TIC) en la comunidad educativa y en la sociedad en general".

Cada una de las aplicaciones que forman parte de este banco de recursos son herramientas que, utilizándolas de forma creativa, permiten aprender y jugar en entornos digitales. El juego es una poderosa fuente de motivación para los alumnos y favorece la construcción de saber. Todas las aplicaciones son de uso libre y pueden descargarse gratuitamente de Internet e instalarse en cualquier computadora.

De esta manera, se promueve la igualdad de oportunidades y las posibilidades para que todos puedan acceder a herramientas que desarrollen la creatividad.

En cada uno de los tutoriales se presentan "consideraciones pedagógicas" que funcionan como recordatorios pero que no deben impedir a los usuarios explorar y desarrollar sus propios usos educativos.

La aplicación de este tutorial no constituye por sí misma una propuesta pedagógica. Su funcionalidad cobra sentido cuando se integra a una actividad. Cada docente o persona que quiera utilizar estos recursos podrá construir su propio recorrido.

## Índice

- [¿Qué es?](#)
- [Requerimientos técnicos](#)
- [Consideraciones pedagógicas](#)
- Nociones básicas
  - [Instalación](#)
  - [Configurar perfil de lenguaje](#)
  - [Conceptos previos](#)
- Paso a paso
  - [Interfaz del programa](#)
  - [Descripción de comandos](#)
    - [Escribir](#)
    - [Leer](#)
    - [Asignar](#)
    - [Si-Entonces](#)
    - [Según](#)
    - [Mientras](#)
    - [Repetir](#)
    - [Para](#)
    - [Subproceso o Funciones](#)
- [Enlaces de interés](#)

## ¿Qué es?

PSeInt es una herramienta que permite ayudar a los estudiantes en sus primeros pasos en programación.



## Requerimientos técnicos

- Puede utilizarse bajo los sistemas operativos Windows y GNU/Linux.
- URL para descargar el programa: <http://pseint.sourceforge.net/>



## Consideraciones pedagógicas

Nivel: Primario - Medio

Áreas sugeridas: todas

- Permite explorar y experimentar con los conceptos de programación.
- Se puede utilizar para centrar la atención en los conceptos fundamentales de la algoritmia computacional, minimizando las dificultades propias de un lenguaje y proporcionando un entorno de trabajo con numerosas ayudas y recursos didácticos.



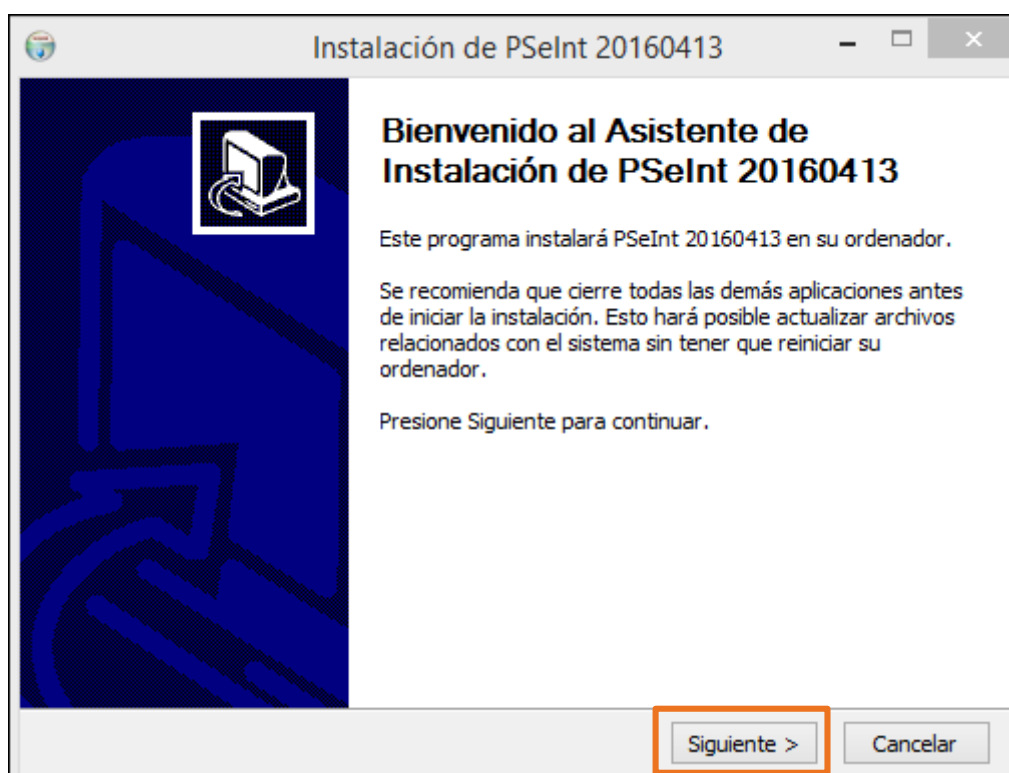
## Nociones básicas

### Instalación

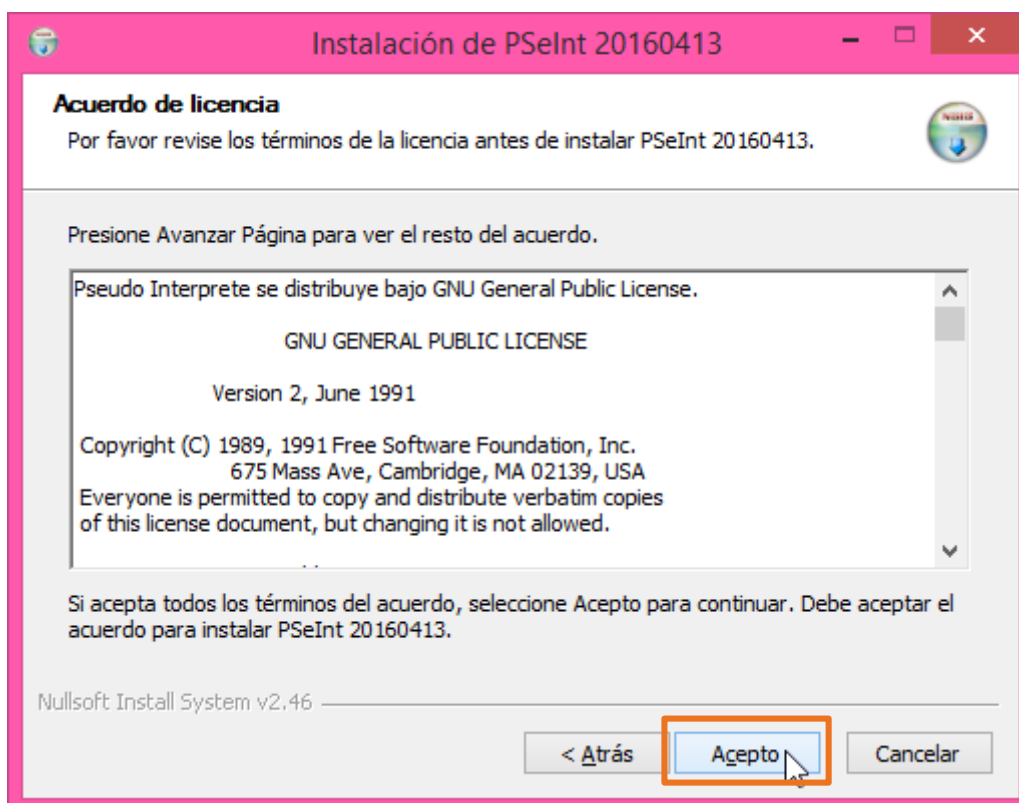
La descarga de programa se realiza desde cualquier navegador ingresando a la página <http://pseint.sourceforge.net/index.php?page=descargas.php>



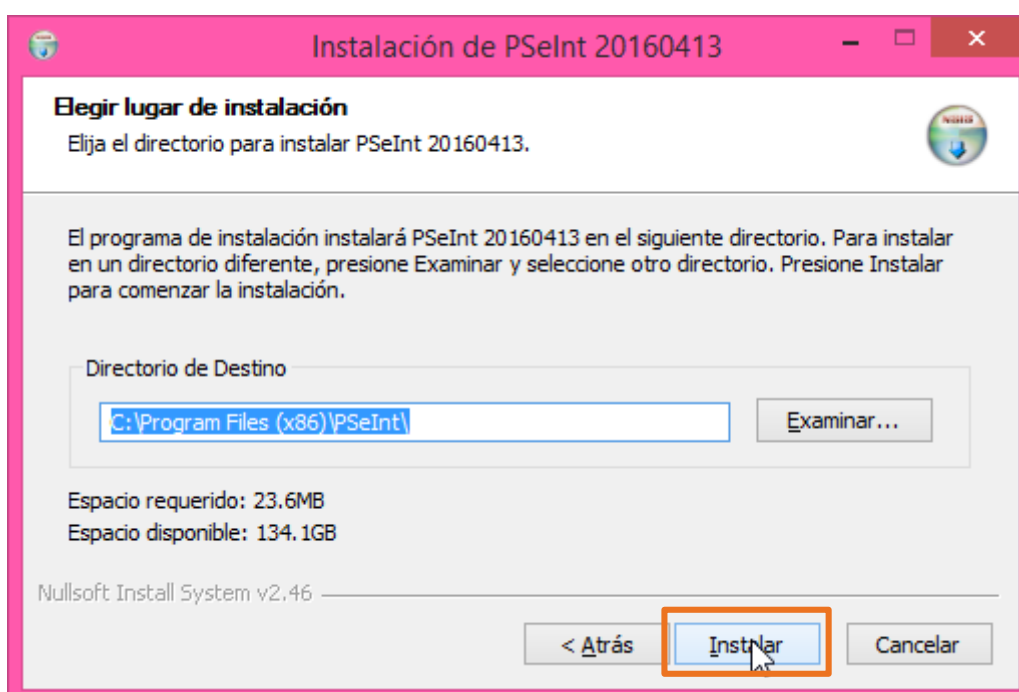
Se descargará un archivo ejecutable. Pulsar dos veces en él para comenzar la instalación.



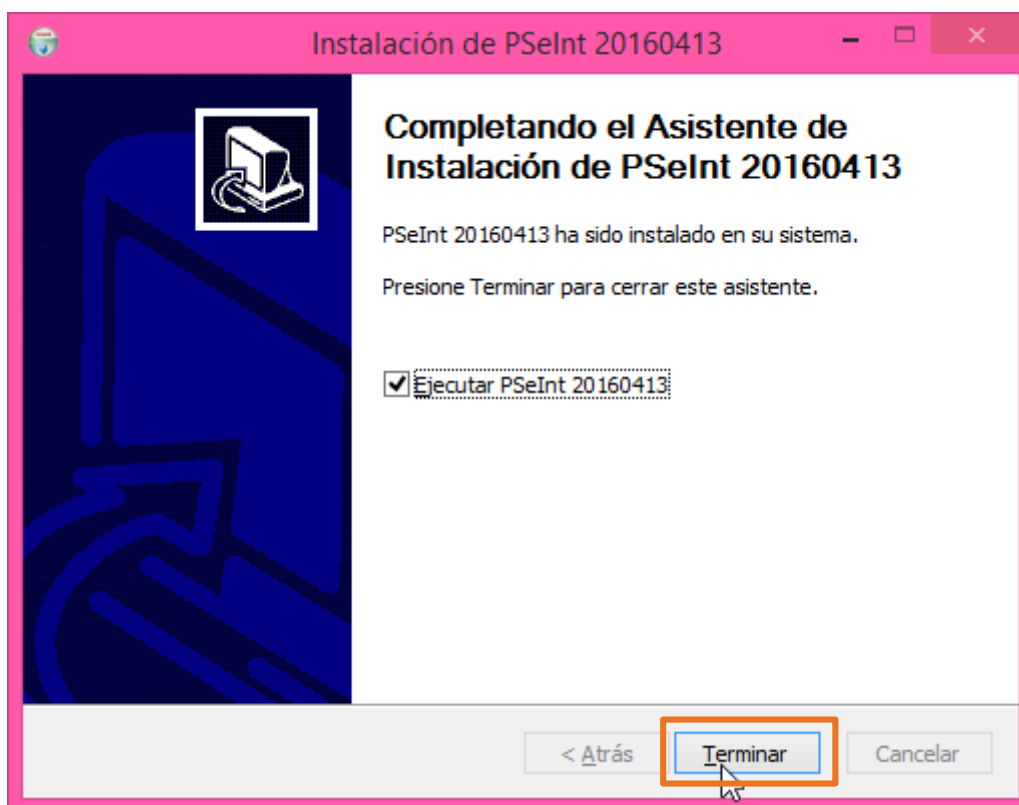
Pulsar el botón **Siguiente>**.



Luego de leer e **Acuerdo de licencia**, presionar e botón **Acepto**.



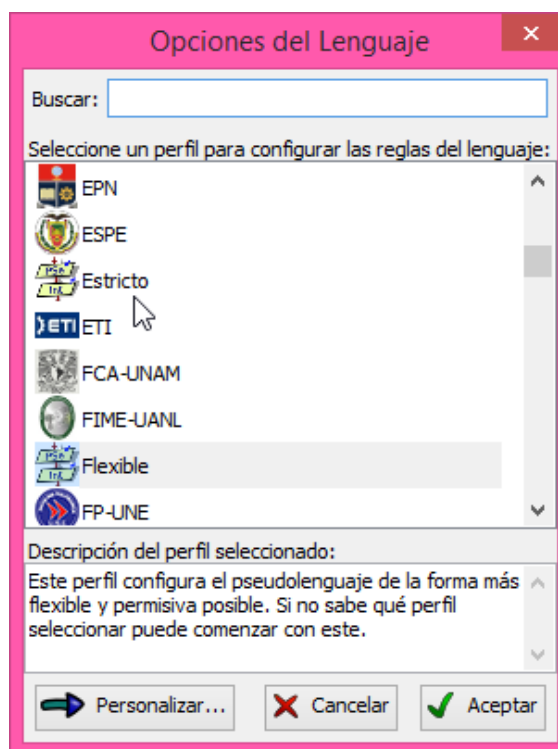
En la siguiente ventana indicar la ubicación de directorio de instalación. Para finalizar presionar el botón **Instalar**.



## Configurar perfil de lenguaje

Elegir Opciones del Lenguaje (Perfiles) de menú Configurar.

En esta sección se pueden configurar distintos perfiles para las reglas del lenguaje. Se destacan **Estricto** (con utilización de sintaxis de programación) y **Flexible** (admite menos sintaxis, por ejemplo no requiere la utilización de punto y coma, comas.)



## Conceptos previos

### Algoritmo

Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución. En la vida cotidiana, se emplean algoritmos frecuentemente para resolver problemas.

### Variable

En programación, una variable está formada por un espacio en el sistema de almacenamiento (memoria principal de una computadora) y un nombre simbólico (un identificador) que está asociado a dicho espacio. Ese espacio contiene una cantidad o información conocida o desconocida, es decir un valor. El nombre de una variable es a forma usual de referirse al valor almacenado: esta separación entre nombre y contenido permite que el nombre sea usado independientemente de la información exacta que representa.

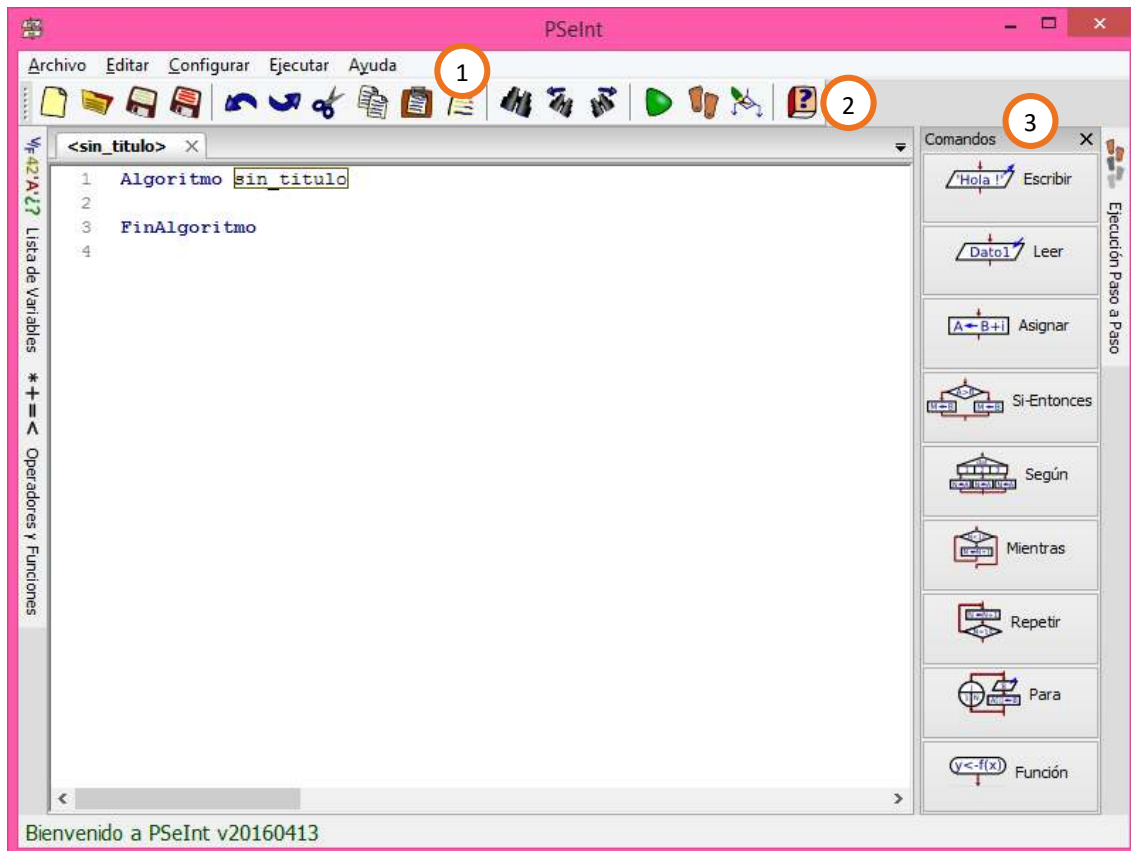
“Definir” se utiliza para declarar las variables “como” para el tipo de datos utilizar.



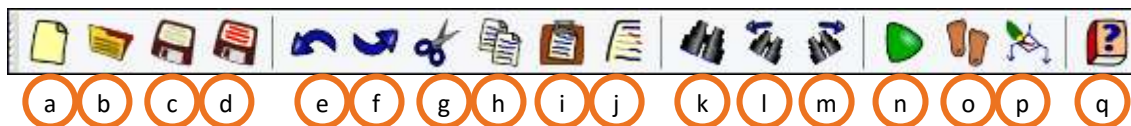


## Paso a paso

### Interfaz del programa



1. Barra de menú
2. Barra de herramientas



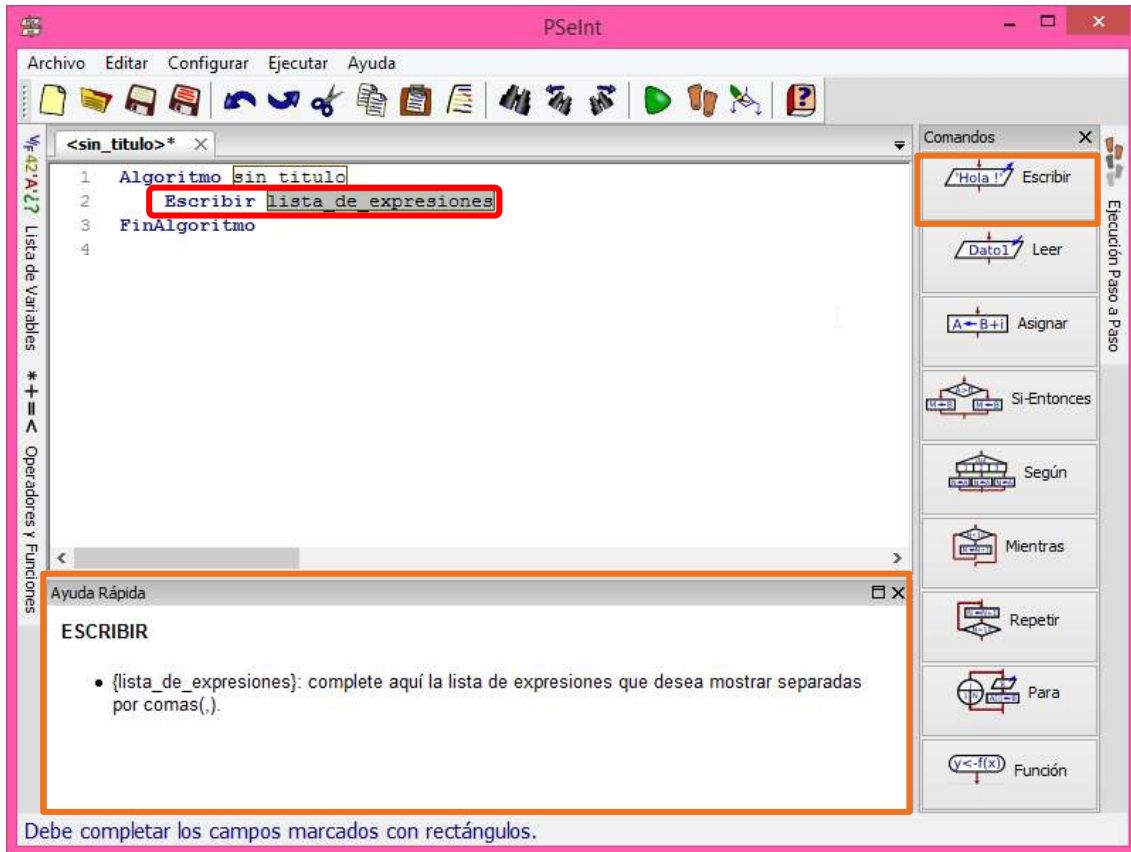
- |                  |                         |                               |
|------------------|-------------------------|-------------------------------|
| a. Nuevo.        | g. Cortar.              | m. Buscar siguiente.          |
| b. Abrir.        | h. Copiar.              | n. Ejecutar.                  |
| c. Guardar.      | i. Pegar.               | o. Ejecutar paso a paso.      |
| d. Guardar como. | j. Corregir intentando. | p. Dibujar diagrama de flujo. |
| e. Deshacer.     | k. Buscar.              | q. Ayuda.                     |
| f. Rehacer.      | l. Buscar anterior.     |                               |

3. Menú de comandos



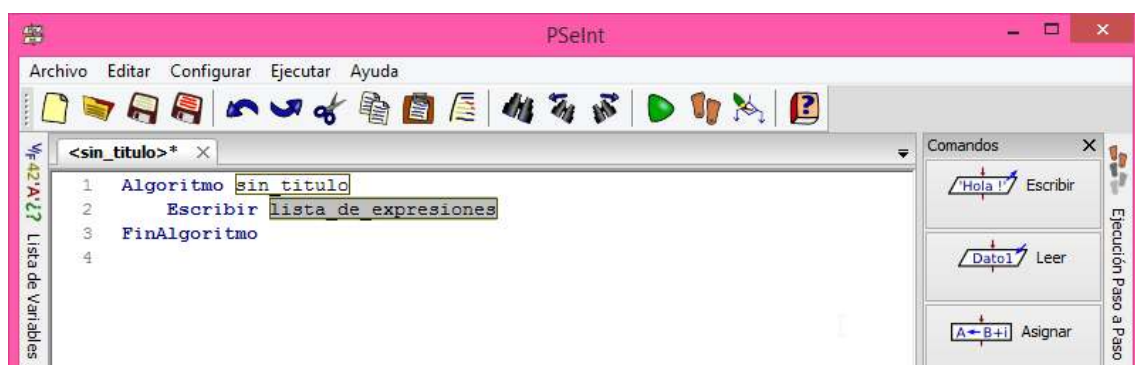
## Descripción de comandos

Pulsando cualquiera de los botones se insertan estructuras o pseudocódigos, se marcan con recuadros los argumentos que deben completar y en la parte inferior de la ventana se muestra una ayuda rápida.



## Escribir

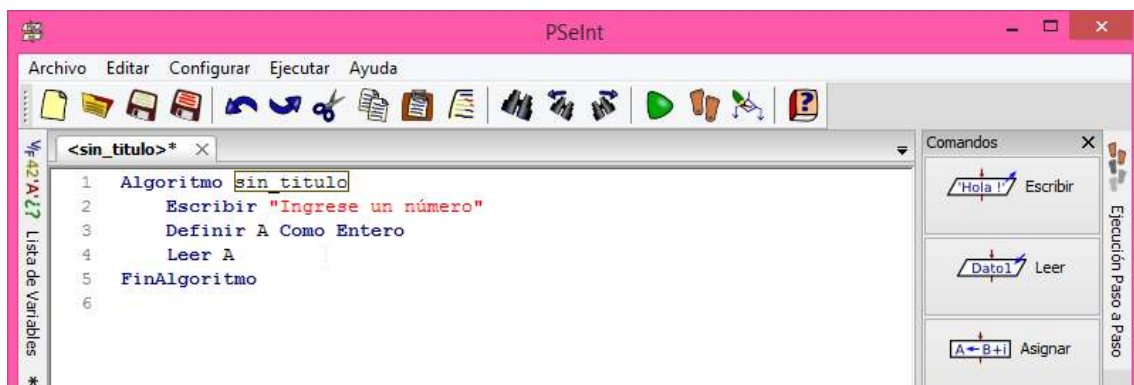
Muestra el mensaje al usuario. Por ejemplo:





## Leer

Lee el dato ingresado por teclado, asignándolo a una variable (en este ejemplo "A").



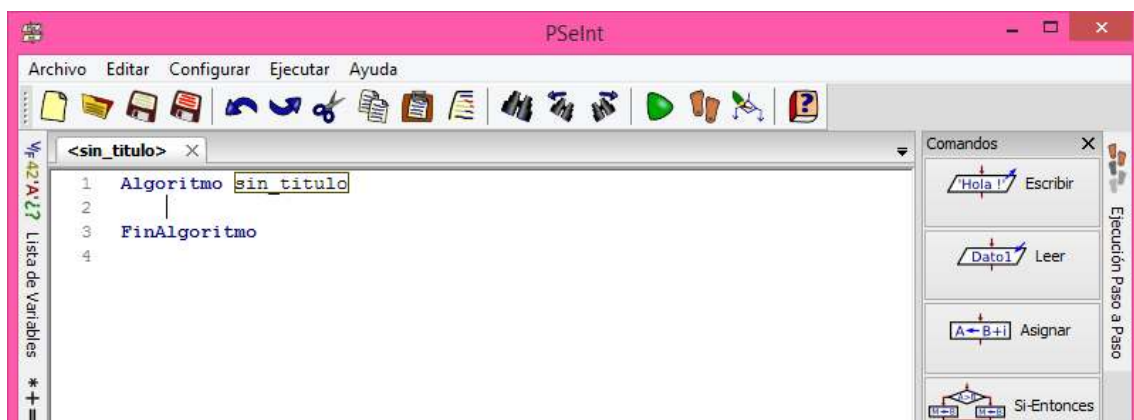
## Asignar

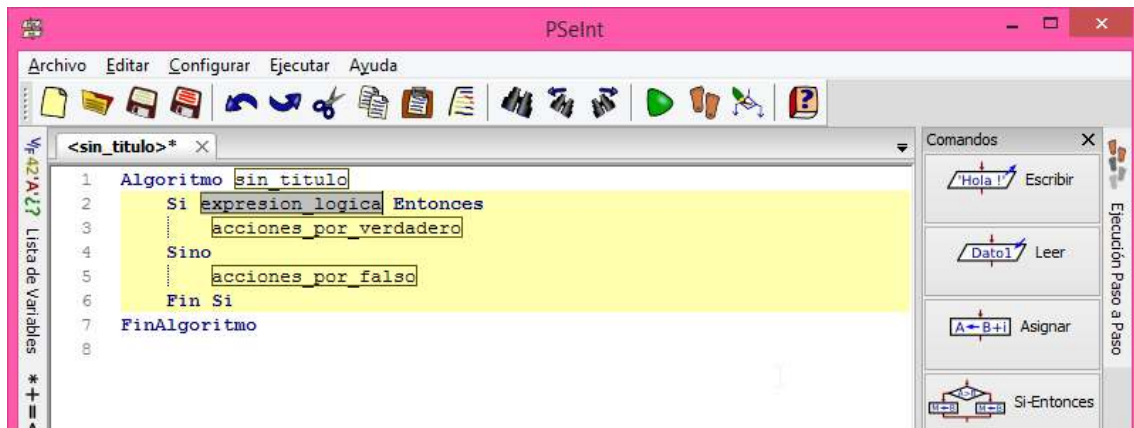
Carga un valor a una variable. En el caso del ejemplo suma 2 valores asignados a las variables A y B. El resultado lo asigna a una variable C.





## Si-Entonces





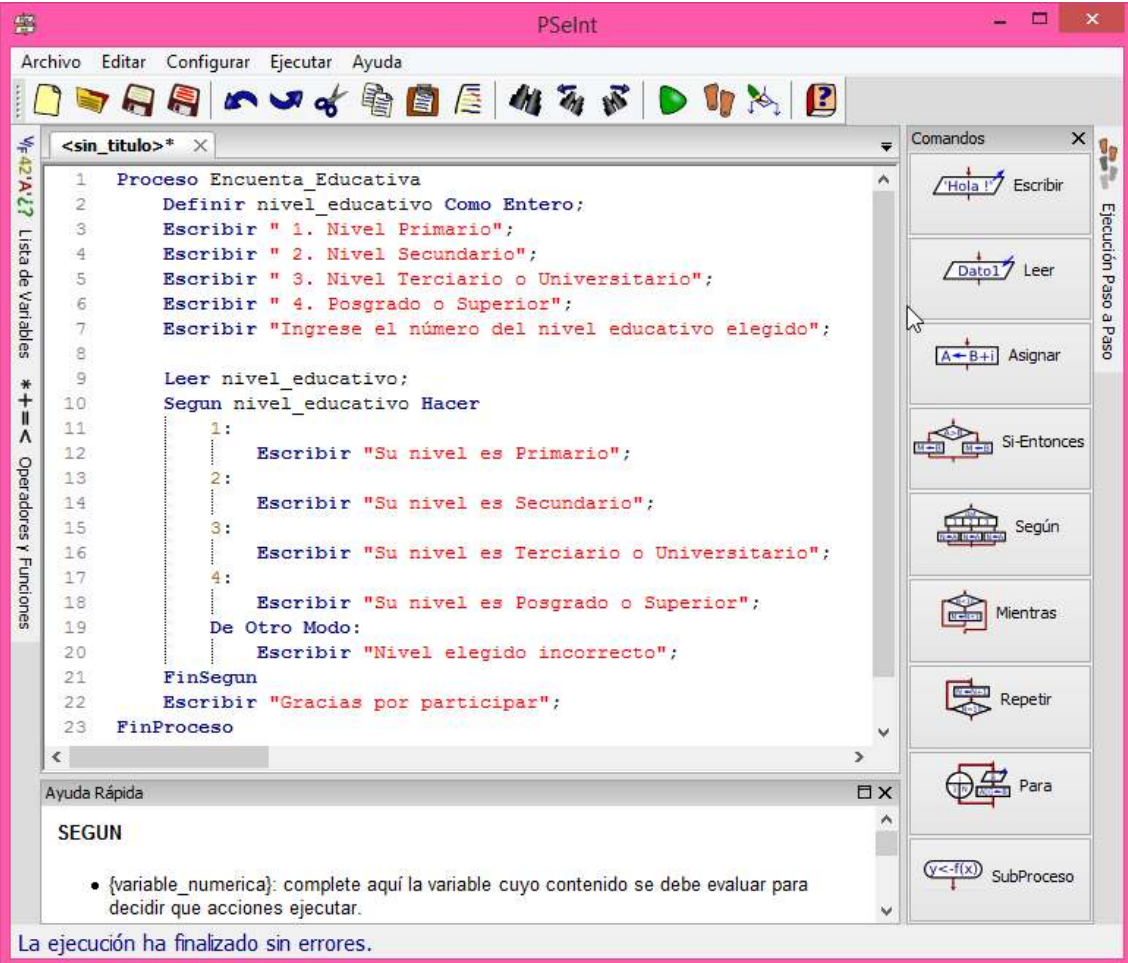
- expresión lógica: se debe ingresar la condición a evaluar.
- acciones\_por\_verdadero: acciones que se deben ejecutar si la expresión\_lógica se cumple.
- acciones\_por\_falso: acciones que se deben ejecutar si la expresión\_lógica no se cumple.

Por ejemplo:



- Definir A, B y C como enteros.
- Ingresar un número y asignarlo en "A".
- Ingresar un número y asignarlo en "B".
- La condición es " $A > B$ " (si A es mayor a B)
- Se presentan 2 caminos posibles:
  - o si A es mayor que B, asignarle a la variable C el valor de A
  - o si A no es mayor que B, asignarle a C el valor de B.

## Según



The screenshot shows the PSeInt IDE with a flowchart for a 'Según' (switch) statement. The code is as follows:

```

1 Proceso Encuenta_Educativa
2   Definir nivel_educativo Como Entero;
3   Escribir " 1. Nivel Primario";
4   Escribir " 2. Nivel Secundario";
5   Escribir " 3. Nivel Terciario o Universitario";
6   Escribir " 4. Posgrado o Superior";
7   Escribir "Ingrese el número del nivel educativo elegido";
8
9   Leer nivel_educativo;
10  Según nivel_educativo Hacer
11      1: Escribir "Su nivel es Primario";
12      2: Escribir "Su nivel es Secundario";
13      3: Escribir "Su nivel es Terciario o Universitario";
14      4: Escribir "Su nivel es Posgrado o Superior";
15      De Otro Modo:
16          Escribir "Nivel elegido incorrecto";
17  FinSegún
18  Escribir "Gracias por participar";
19 FinProceso

```

The 'Comandos' panel on the right shows the 'Según' command selected. The status bar at the bottom indicates 'La ejecución ha finalizado sin errores.'

- Definir en primer lugar una variable donde se guardará la información ingresada. En este ejemplo "Definir Nivel\_elegido Como entero";
- En este ejemplo el programa mostrará en pantalla las distintas opciones: "1. Nivel Primario, 2. Nivel Secundario, 3. Nivel Terciario o Universitario, 4. Posgrado".
- Se pide al usuario, por pantalla, elegir entre las opciones. Luego al leerla se guarda la información en la variable:
  - o Escribir "Ingrese el número de la opción deseada";
  - o Leer Nivel\_elegido; //guarda el nivel elegido en la variable.
- En caso contrario dar aviso de error.
- En este caso Según funciona de acuerdo a la opción ingresada por pantalla. El programa mostrará el nivel elegido.

## Mientras

```

1 Proceso adivina_numero
2   Definir intentos, numero_ingresado, numero_secreto Como Entero;
3   intentos <- 10;
4   numero_secreto <- 28;
5   Escribir "Adivina un número de 1 a 100";
6   Leer numero_ingresado;
7
8   Mientras numero_secreto <> numero_ingresado y intentos >= 1 Hacer
9     Si numero_secreto > numero_ingresado Entonces
10      Escribir "Número muy bajo";
11     Sino
12      Escribir "Número muy alto";
13     FinSi
14     intentos <- intentos - 1;
15     Escribir "Le quedan ", intentos, " intentos";
16   FinMientras
17   Si numero_secreto = numero_ingresado Entonces
18     Escribir "Correcto, usted adivinó en ", 10-intentos, " intentos";
19   Sino
20     Escribir "Fallo, usted no adivinó, el número secreto era ", numero_secreto;
21   FinSi
22 FinProceso
  
```

El pseudocódigo es correcto. Presione F9 para ejecutarlo.

- Primero se deben definir las variables que se utilizarán. En este ejemplo son 3:  
**Definir intentos, numero\_ingresado, numero\_secreto Como Entero;** // variables
- Asignar valores a las variables necesarias.  
**intentos <- 10;** // cantidad de oportunidades que tendrá el usuario para adivinar.  
**numero\_secreto <- 28;** // número a adivinar.
- Pedir al usuario que adivine el número y lo ingrese.  
**Escribir "Adivina un número de 1 a 100";** // muestra en pantalla al usuario.  
**Leer numero\_ingresado;** // carga el valor ingresado por el usuario en la variable.
- Aquí se ve la función de la instrucción **Mientras**. Se realiza infinitamente hasta que dejar de cumplirse las condiciones iniciales.  
**Mientras numero\_secreto <> numero\_ingresado y intentos >= 1 Hacer**  
**//condiciones iniciales.**

Si numero\_secreto > numero\_ingresado Entonces //ayuda a aproximarse.

    Escribir "Número muy bajo";

Sino

    Escribir "Número muy alto";

FinSi

intentos <- intentos - 1; //contador para ir restando los intentos.

Escribir "Le quedan ", intentos, " intentos"; //informa intentos restantes.

FinMientras

- Sale del Mientras cuando dejaron de cumplirse las condiciones iniciales. En este ejemplo, el usuario acertó y el número\_secreto = número\_ingresado o el usuario utilizó los 10 intentos, para los que se utilizará un si-entonces para mostrar ambas opciones.

Si numero\_secreto = numero\_ingresado Entonces //caso correcto

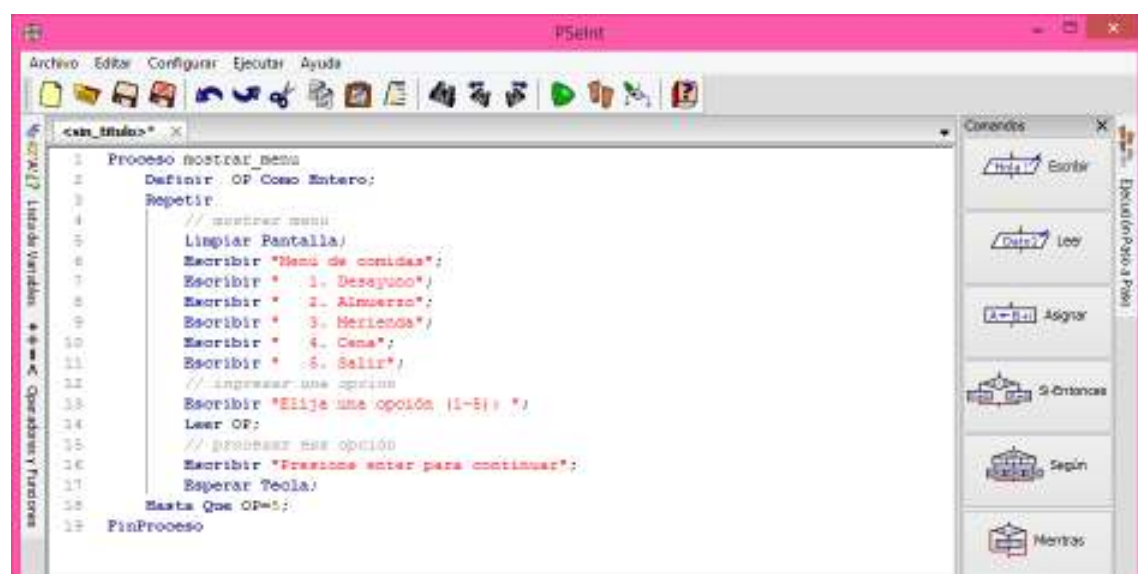
    Escribir "Correcto, usted adivinó en ", 10-intentos, " intentos";

Sino // caso en que no acertó

    Escribir "Falló, usted no adivinó. El número secreto era ",  
    numero\_secreto;

FinSi

## Repetir





- Definir las variables a utilizar.  
**Definir OP Como Entero;** //definir la variable.
- Ciclo "repetir" que se repite indefinidamente "hasta".

**Repetir**

// Mostrar menú

**Limpiar Pantalla;** // limpia la pantalla para cada repetición, luego muestra el menú.

**Escribir "Menú de comidas";**

**Escribir " 1. Desayuno";**

**Escribir " 2. Almuerzo";**

**Escribir " 3. Merienda";**

**Escribir " 4. Cena";**

**Escribir " 5. Salir";**

// Ingresar una opción por teclado

**Escribir "Elija una opción (1-5): ";**

**Leer OP;**// grabar la opción en la variable

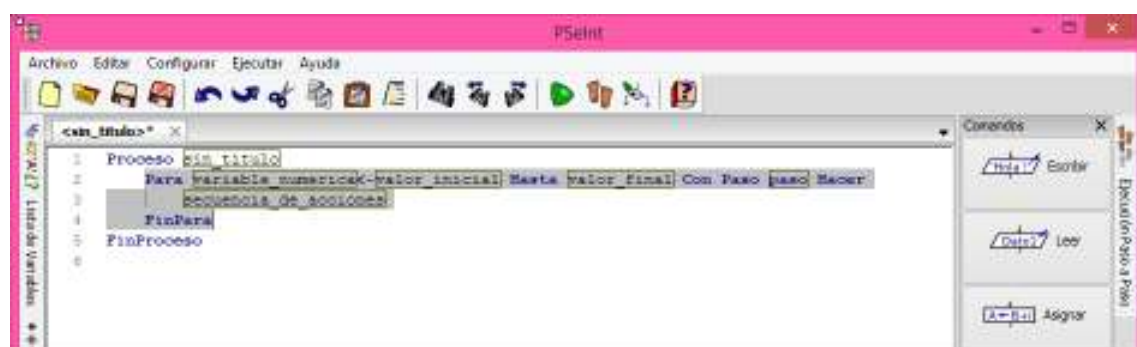
// Solicitar que el usuario presione la tecla Enter

**Escribir "Presione la tecla Enter para continuar";**

**Esperar Tecla;**

**Hasta Que OP=5;**

**Para**



- Variable numérica se debe ingresar el nombre de la variable por ejemplo "i".
- Valor inicial debe tener el valor numérico o variable desde el que comenzará.
- Valor final debe tener el valor numérico o variable hasta el que cumplirá el ciclo.

- Paso indica la secuencia de incrementos, si es de 1 en 1 "1", 2 en 2 "2".
- Secuencia de acciones debe tener las instrucciones del programa que efectuará con cada iteración.

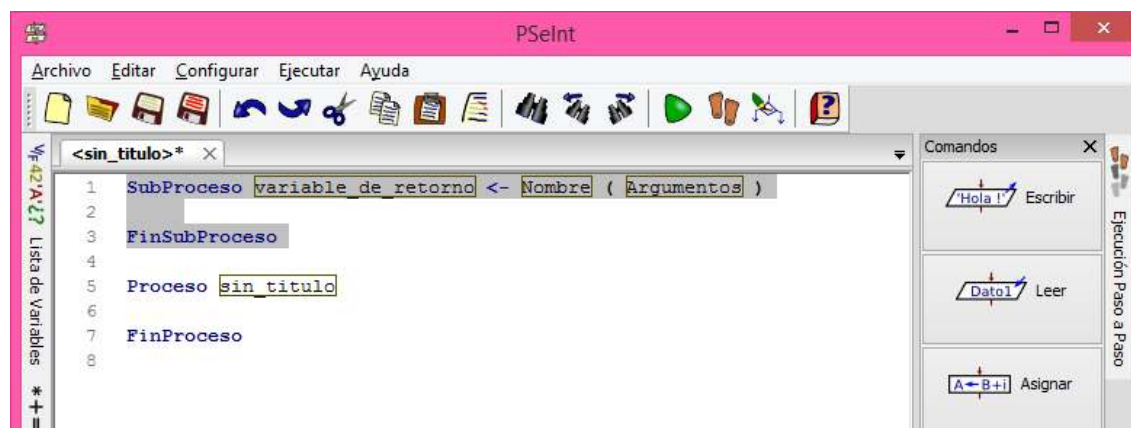
```

1 Proceso suma_numeros
2   Definir AcumNum Como Entero;
3   AcumNum<-0;
4   Definir SumaTotal Como Entero;
5   SumaTotal<-0;
6   Definir N Como Entero;
7   Definir i Como Entero;
8   Definir Num Como Entero;
9   Escribir "Cuantos números vamos a sumar ";
10  Leer N;
11  Para i<-1 Hasta N Con Paso 1 Hacer
12    Escribir "Ingrase el número ";
13    Leer Num;
14    AcumNum <- (AcumNum + Num);
15  FinPara
16  SumaTotal<-AcumNum;
17  Escribir "El resultado es ", SumaTotal;
18 FinProceso

```

- En el ejemplo se definen las variables a utilizar: **AcumNum**, **SumaTotal**, **N**, **i**, **Num**. Todas se definen como entero y se inicializan en 0.
- Se solicita por pantalla "Cuantos números vamos a sumar".
- Lee el número ingresado por el usuario y lo guarda en la variable **N**.
- Luego ingresa en el ciclo **PARA**. Se repetirá desde **i = 1** hasta **N** (que es el valor ingresado previamente).
- A continuación se pedirá al usuario que vaya ingresando los números, mientras se van acumulando en la variable **AcumNum** y sumándolas por cada ciclo.
- Finalmente se asigna la suma acumulada a una nueva variable **SumaTotal** y se muestra el resultado por pantalla.

## SubProceso o Funciones



- En **variable de retorno** se ingresa el valor a retornar. Si el subproceso no devuelve nada, se puede eliminar.
- Luego en **Nombre** se le asigna un nombre.
- Los **argumentos** son los que requiere el subproceso para funcionar. Si no hay argumentos se deja vacío o se quitan los paréntesis.



- El subproceso **calculo** multiplica por 2 el argumento ingresado "Arg" y retornarlo en una variable "retorno".
- El proceso **Multiplicar\_2** primero define las variables a utilizar; luego lee la entrada y se pasa al subproceso como el argumento "Arg". Le pide al usuario que ingrese una entrada y la pasa al subproceso para que realice el cálculo.
- Finalmente se muestra el resultado del cálculo.

## Enlaces de interés

- Sitio oficial: <http://pseint.sourceforge.net>
- Programación con PseInt – Introducción:  
<https://campusvirtualintec.buenosaires.gob.ar/course/view.php?id=552>
- Programación con PseInt – Primeros pasos:  
<https://campusvirtualintec.buenosaires.gob.ar/course/view.php?id=553>
- Programación con PseInt – Operaciones con variables:  
<https://campusvirtualintec.buenosaires.gob.ar/course/view.php?id=554>
- Programación con PseInt – Condicionales:  
<https://campusvirtualintec.buenosaires.gob.ar/course/view.php?id=555>
- Programación con PseInt – Estructuras de repetición:  
<https://campusvirtualintec.buenosaires.gob.ar/course/view.php?id=556>



**Contacto:**

[ayuda.pedagogico.digital@bue.edu.ar](mailto:ayuda.pedagogico.digital@bue.edu.ar)



Esta obra se encuentra bajo una Licencia Attribution-NonCommercial-ShareAlike 2.5 Argentina de Creative Commons. Para más información visite <http://creativecommons.org/licenses/by-nc-sa/2.5/ar/>



**Vamos Buenos Aires**