



**UNIVERSIDAD AUTÓNOMA "GABRIEL RENÉ MORENO"**  
**CARRERA DE INGENIERÍA INFORMÁTICA E**  
**INGENIERÍA DE SISTEMAS**



# **“PSEINT EN LA LÓGICA ALGORÍTMICA”**

Elaborada por: Iveth Bescie Robles Catari

Santa Cruz, Camiri, Diciembre de 2019

## INDICE GENERAL

<b>1. Definición de Algoritmo .....</b>	<b>2</b>
<b>2. Diagramas de flujo .....</b>	<b>2</b>
<b>2.1 Herramientas de Software para crear diagramas de flujo .....</b>	<b>3</b>
<b>2.1.1 FreeDFD (DFD) .....</b>	<b>3</b>
<b>2.1.2 PSeInt .....</b>	<b>5</b>
<b>2.1.3 Historia de PSeInt .....</b>	<b>5</b>
<b>2.1.4 El creador de PSeInt .....</b>	<b>6</b>
<b>2.1.5 Características y funcionalidades de PSeInt .....</b>	<b>6</b>
<b>3. Aplicación .....</b>	<b>15</b>

## INDICE DE IMÁGENES

Imagen 1: PSeInt interfaz de creación de pseudocódigo .....	20
Imagen 2: PSeInt interfaz de creación de Diagramas .....	21
Imagen 3: PSeInt interfaz de ejecución de un algoritmo .....	22
Imagen 4: PSeInt interfaz de Acerca de .....	22

# **DESARROLLO DE ALGORITMOS**

## **1. Definición de Algoritmo**

Según (Instituto Tecnológico de Sonora, 2016) lo define: “Un algoritmo es un conjunto de instrucciones que especifica la secuencia de operaciones a realizar, en orden, para resolver un problema específico; en otras palabras, un algoritmo es una fórmula para la resolución de un problema” (p.3).

Para (Gómez & Salgado Reyes, 2016) lo define de la siguiente manera: “Algoritmo es una serie de pasos sucesivos, con el objetivo de dar solución a un problema, y cuya característica es tener un inicio y un fin”(p.4).

Por último y siendo este uno de los autores más sobresalientes en lo que refiere a libros relacionados con la metodología de programación, (Joyanes Aguilar, 2008) define: “Un algoritmo es un método para resolver un problema. (...) proviene (...) de Mohammed alKhoWârizmi (...), la traducción al latín del apellido en la palabra algorismus derivó posteriormente en algoritmo” (p.64).

Los conceptos antes mencionados convergen de manera común en que un algoritmo representa una serie de pasos a seguir para resolver un problema.

## **2. Diagramas de flujo**

La historia de los diagramas de flujos según (Lucid Software Inc., 2018) tiene sus inicios entre las décadas del 1920 y 1930, creándose para documentar procesos de negocios. En 1921,

los ingenieros industriales Frank y Lillian Gilbreth presentaron el "diagrama de flujo de procesos" en la Sociedad Americana de Ingenieros Mecánicos (ASME – American Society of Mechanical Engineers) (par.2).

A fines de la década de 1940, Herman Goldstine y John Van Neumann usaron diagramas de flujo para desarrollar programas informáticos, la creación de diagramas se volvió cada vez más popular para los programas informáticos y algoritmos de todo tipo, hoy en día se sigue utilizando los diagramas de flujo para la programación por la facilidad en la comprensión de los algoritmos a través de diagramas.

## **2.1 Herramientas de Software para crear diagramas de flujo**

Actualmente existe una gran variedad de herramientas de software, tanto para la elaboración de diagramas de flujo, pseudocódigo, unos son comerciales y otros de uso libre (IEDA, n.d.) : FreeDFD, Dia, Raptor, PSeInt.

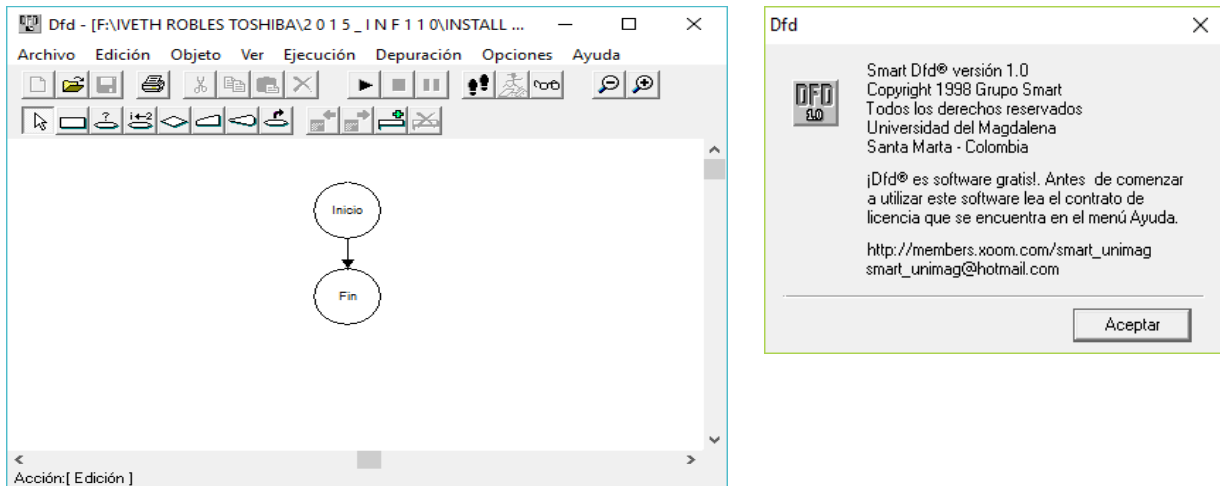
### **2.1.1 FreeDFD (DFD)**

FreeDFD es una herramienta de software de libre disposición para ayuda al diseño e implementación de algoritmos expresados en diagramas de flujo (DF).

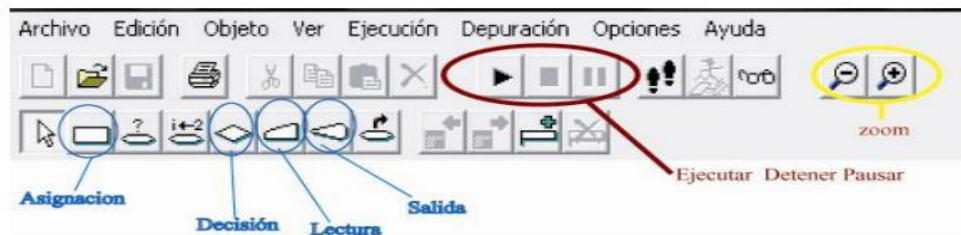
Esta herramienta (Chino Mamani M. Á., 2018) :

Surge en 1988 de un proyecto colombiano llamado editor e intérprete de algoritmos representados en diagramas de flujo con el objetivo de lograr eliminar ciertas dificultades presentes en el estudio de algoritmos básicos. La última versión disponible en línea (octubre de 2008) lleva el nombre de FreeDFD y tiene su página web en googleCode. En esta página DFD se redefine como “un editor, intérprete y depurador de algoritmos representados en diagramas de flujo”. Con esta herramienta el usuario puede trabajar con expresiones complejas que

involucren constantes, variables, funciones y operadores. Los tipos de datos que maneja son: reales, cadenas de caracteres y lógicos; también se permite el uso de arreglos. Durante la ejecución de un diagrama se detectan errores de sintaxis y de conformación de subprogramas.(p.5)



La Barra de herramientas es la siguiente:



Entre las Ventajas y desventajas de FreeDFD, según (Softonic, 2018) se tiene:

Ventajas:

- Ejecución y depuración de diagramas de flujo.
- Evaluador de expresiones.
- Fácil de usar.

Desventajas:

- No puede deshacer acciones.

- Edición gráfica muy limitada.

Esta herramienta ha sido utilizada en la asignatura en anteriores semestres para la creación de diagramas en la resolución de algoritmos de la asignatura de Introducción a la Informática.

Sin embargo, como se indica anteriormente esta herramienta resulta muy limitada ya que sólo es posible ingresar el algoritmo representado por el diagrama, ejecutarlo, analizarlo y depurarlo.

### **2.1.2 PSeInt**

La definición de PSeInt según (SourceForge, 2018), es la siguiente:

PSeInt es una herramienta para asistir a un estudiante en sus primeros pasos en programación. Mediante un simple e intuitivo pseudolenguaje en español (complementado con un editor de diagramas de flujo), le permite centrar su 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. (parr.1)

Se toma la anterior definición para enfatizar que el PSeInt es una herramienta muy útil para los estudiantes para profundizar los conceptos fundamentales del algoritmo computacional ya que se puede diseñar y ejecutar diagramas, pseudocódigo y exportar a código.

### **2.1.3 Historia de PSeInt**

La historia de este programa según (Programación con PSeInt, 2017) es la siguiente:

El software comenzó a desarrollarse en octubre del 2003 utilizando Borlan C++ Builder debido a que era esta herramienta que recomendaba la cátedra por la facilidad que brinda para el desarrollo de interfaces visuales y estar presente en los laboratorios de la facultad. comenzó con una prueba de concepto, ya que era el primer proyecto que iba a realizar con C++. Al ser un software para uso exclusivamente didáctico, no se presentan grandes problemas de rendimiento (no

se tienen que interpretar algoritmos de miles de líneas sumamente complejos) por los que pudo tomar ciertas libertades en la implementación.

Casi dos años más tarde, gracias a la lista de correo del grupo de Usuarios de Linux del litoral (Lugli) llegó la invitación a participar en la recopilación de Software libre de la colección que se distribuiría gratuitamente entre los docentes que lo solicitaron por el ministerio de Educación de la Nación y llevado a cabo por los grupos gleducar, solar y educ.ar. Sugirió entonces a través de una lista de correo creada para la recopilación del PSEINT en dicho CD. (parr.1-2)

La historia de este programa es que nace como un proyecto final, según (Castillo Suazo & Caro, 2016) quien indica lo siguiente:

PSeInt, es la abreviatura de Pseudocode Interpreter, Intérprete de Pseudocódigo. Este programa fue creado como proyecto final para la materia Programación 1 de la Carrera Ingeniería en Informática de la Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral, del en aquel momento estudiante Pablo Novara. (p.3)

#### **2.1.4 El creador de PSeInt**

El creador de esta herramienta de software en (Programación con PSeInt, 2017), indica que:

PSeInt es una herramienta desarrollada por Pablo Navara para aprender de la lógica de la programación, dirigida a estudiantes sin experiencia en esta área. Mediante la utilización de un simple y limitado pseudo-lenguaje intuitivo y en español, permite comenzar a comprender conceptos básicos y fundamentales de un algoritmo computacional. (párr. 3)

La creación de esta herramienta de software significa un gran aporte al área de programación y desarrollo de algoritmos.

#### **2.1.5 Características y funcionalidades de PSeInt**

Para (SourceForge, 2018)al respecto de las características y funcionalidades indica:

**Presenta herramientas de edición para escribir algoritmos en pseudocódigo en español**

- Autocompletado
- Ayudas Emergentes
- Plantillas de Comandos
- Coloreado de Sintaxis
- Resaltado de bloques lógicos
- Indentado Inteligente
- Listados de funciones, operadores y variables

**Permite generar y editar el diagrama de flujo del algoritmo**

- Puede trabajar con diagramas clásicos y de Nassi-Shneiderman

**Permite la edición simultánea de múltiples algoritmos**

**El pseudo-lenguaje utilizado es configurable**

- Ofrece perfiles de configuración predefinidos para numerosas instituciones

**Puede interpretar (ejecutar) los algoritmos escritos**

- Puede modificar el algoritmo y ver los cambios en la ejecución inmediatamente (sin reingresar los datos)
- Permite modificar uno o más datos selectos de una ejecución ya finalizada para observar cómo varían los resultados
- Permite deshacer una ejecución para reiniciarla o repetirla desde un punto arbitrario
- Permite ejecutar el algoritmo paso a paso controlando la velocidad e inspeccionando variables y expresiones
- Puede confeccionar automáticamente una tabla de prueba de escritorio
- Ofrece un modo especial en el que describe las acciones realizadas en cada paso

**Determina y marca claramente los errores**

- Señala errores de sintaxis en tiempo real (mientras escribe)
- Señala claramente errores en tiempo de ejecución
- Ofrece descripciones detalladas de cada error, con sus causas y soluciones más frecuentes.

**Permite convertir el algoritmo de pseudocódigo a código numerosos lenguajes de programación.**

- C, C++, C#, Java, JavaScript, MatLab, Pascal, PHP, Python 2, Python 3, QBasic Visual Basic

**Ofrece un sistema de ayuda integrado acerca del pseudocódigo y el uso del programa**

- Incluye un conjunto de ejemplos de diferentes niveles de dificultad

**Es multiplataforma (probado en Microsoft Windows, GNU/Linux y Mac OS X)**

**Es totalmente libre y gratuito (licencia GPLv2)(parr.2)**

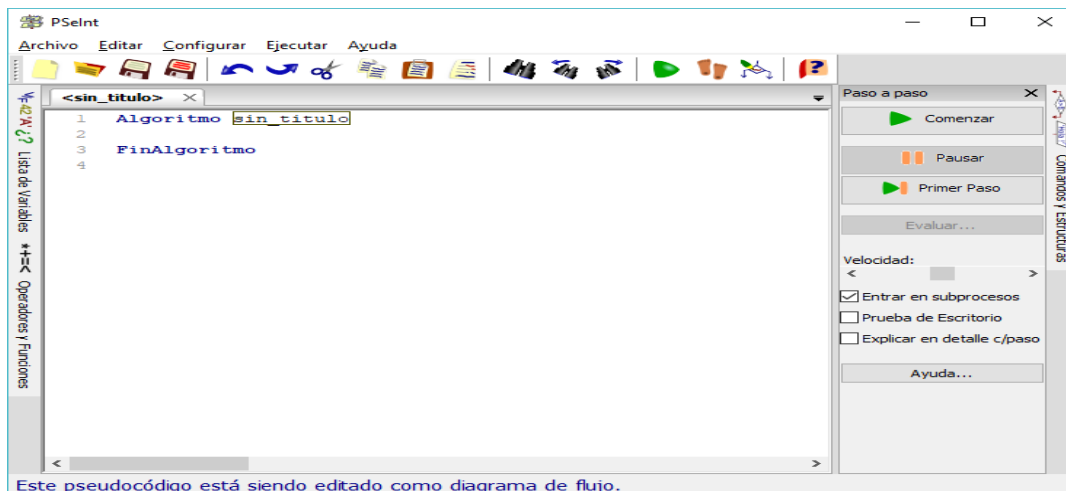
Según (Castillo Suazo & Caro, 2016) es recomendable usar PSeInt por los siguiente:



- 1) Porque es software libre, sin necesidad de andar gastando dinero, haciendo giros, etc., violando los derechos de autor ni andar creando o consiguiendo cracks, que a veces sus links están inactivos y/o los programas no dejan craquearse.
- 2) Está constantemente atendido por su creador, a diferencia de los otros compiladores e intérpretes de pseudocódigo que están descontinuados.
- 3) Posee un foro para reportar errores y obtener ayuda, también está constantemente atendido por su creador, lo que ayuda a mejorar el programa.
- 4) Posee una extensa ayuda, que valga la redundancia ayuda a aprender a usarlo, y a aprender el lenguaje.
- 5) Está disponible su código fuente, y con instrucciones para ejecutarlo, de modo que si sabemos C++ podremos personalizarlo y corregirlo.
- 6) Posee previsualización y exportación a C, C++ y otros lenguajes para que podamos ver el mismo código implementado en C y otros lenguajes, lo que ayuda a aprender estos y otros lenguajes;
- 7) Se trata de un compilador que compila automáticamente cuando el usuario pulsa ejecutar, el algoritmo se guarda automáticamente en un archivo del disco duro, dentro de la carpeta del ejecutable PSeInt, para su posterior ejecución, haciendo más cómodo su uso. (pp.3-4)

En la presente investigación se utiliza PSeInt por las recomendaciones antes expuestas y en especial porque es un software libre (**licencia GPLv2**) lo cual no implica para la Dirección de la Carrera de Ingeniería Informática ningún costo en su adquisición.

Es importante destacar de entre las anteriores características antes mencionadas que PSeInt es totalmente libre y gratuito además de que es posible convertir el algoritmo del Pseudocódigo a código.



## Botones

**Panel de Comandos:** Permite introducir acciones o estructuras de control en su algoritmo mediante un clic. Se ubica por defecto en la parte izquierda de la ventana y al seleccionar una de sus primitivas introduce en el código del proceso actual la plantilla de la misma

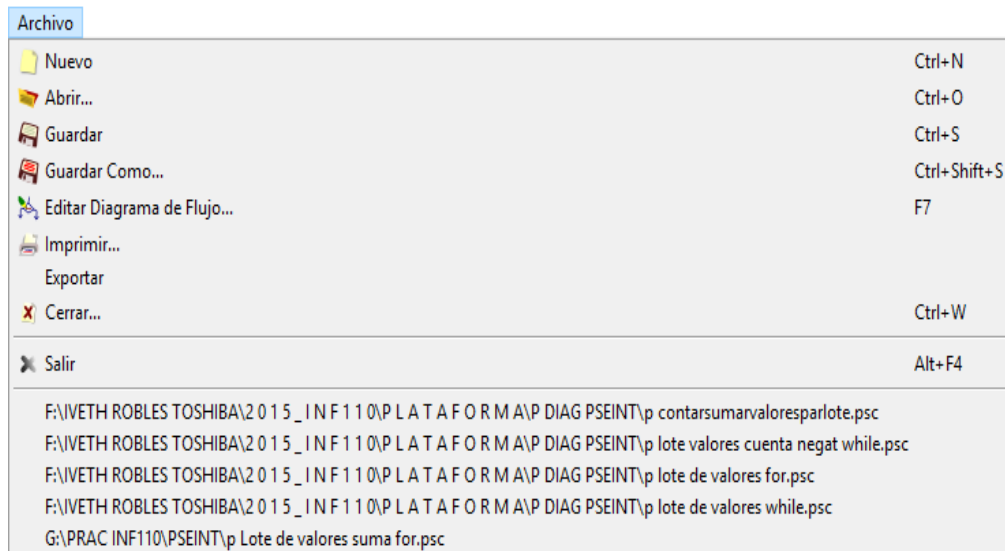
**Área de Ejecución:** Se actualiza cada vez que se ejecuta el algoritmo. En ella aparecen los errores encontrados, ya sean de sintaxis (antes de

ejecutarse) o errores en tiempo de ejecución. Al hacer clic sobre los mismos, automáticamente se seleccionará la línea que contiene dicho error en el área de trabajo



## Menú:

- **Menú Archivo**




















**Nuevo:** Crea un nuevo algoritmo y escribe el prototipo básico en el área de trabajo.

- **Abrir:** Abre un algoritmo desde un archivo.
- **Guardar:** Guarda todos los cambios realizados en el algoritmo (pseudocódigo) actual.
- **Guardar Como:** Guarda el algoritmo actual con un nuevo nombre.
- **Cerrar:** Cierra la aplicación.
- **Exportar** , a partir del pseudocódigo: Convertir a Código C(c), Convertir a Código C++ (cpp), Convertir a Código HTML (html), Convertir a Código Java (java), Convertir a Código JavaScript (js), Convertir a Código MatLab (m), Convertir a Código Pascal (pas), Convertir a Código PHP (php), Convertir a Código Python 2 (py), Convertir a Código Python 3 (py), Convertir a Código QBasic (.bas), Convertir a Código Visual Basic.Net (vb), Convertir a Código, Convertir a Código

- **Menú Editar:**

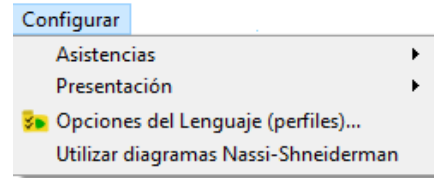
- **Deshacer:** Deshace el último cambio realizado al pseudocódigo.
- **Rehacer:** Deshace el último cambio realizado al pseudocódigo
- **Cortar, Copiar y Pegar:** Interactúan entre el texto del área de trabajo y el portapapeles de Windows para cortar, copiar o pegar texto seleccionado.
- **Mover Hacia Arriba:** Mueve el ó los elementos seleccionados a la línea de arriba.
- **Mover Hacia Abajo:** Mueve el ó los elementos seleccionados a la línea de abajo.
- **Buscar:** Localiza una cadena dentro del algoritmo. No distingue entre mayúsculas y minúsculas.
- **Buscar Siguiente:** Repite la última búsqueda realizada comenzando desde la ubicación del cursor.
- **Buscar Anterior:** Repite la última búsqueda realizada comenzando desde la ubicación del cursor hasta encontrar la coincidencia anterior.
- **Seleccionar Todo:** Selecciona todo el pseudocódigo.
- **Duplicar Líneas:** Permite duplicar las líneas seleccionadas.
- **Eliminar Líneas:** Permite eliminar las líneas seleccionadas.
- **Comentar Líneas:** Permite comentar las líneas seleccionadas.

Editar		
	Deshacer	Ctrl+Z
	Rehacer	Ctrl+ Shift+Z
	Cortar	Ctrl+X
	Copiar	Ctrl+C
	Pegar	Ctrl+V
	Mover Hacia Arriba	Ctrl+T
	Mover Hacia Abajo	Ctrl+ Shift+T
	Buscar...	Ctrl+F
	Buscar Anterior	Shift+F3
	Buscar Siguiente	F3
	Reemplazar...	Ctrl+R
	Seleccionar Todo	Ctrl+A
	Duplicar Líneas	Ctrl+L
	Eliminar Líneas	Ctrl+ Shift+L
	Comentar Líneas	Ctrl+D
	Descomentar Líneas	Ctrl+ Shift+D
	Corregir Indentado	Ctrl+I

- **Descomentar Líneas:** Permite descomentar las líneas seleccionadas.

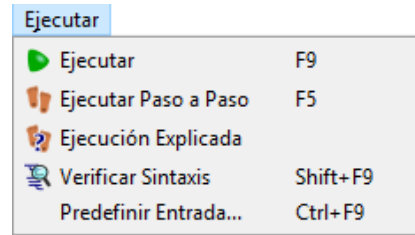
- **Menú Configurar:**

Permite realizar distintas configuraciones desde el autocompletado, ayudas emergentes, opciones del lenguaje, diagramas etc.



- **Menú Ejecutar**

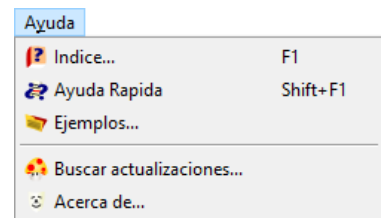
- **Ejecutar:** Ejecuta el algoritmo en una nueva ventana de consola. Si el algoritmo contiene errores de sintaxis, no será ejecutado.



- **Ejecutar Paso a Paso:** La ejecución paso a paso permite realizar un seguimiento más detallado de la ejecución del algoritmo. Es decir, permite observar en tiempo real qué instrucciones y en qué orden se ejecutan, así como también observar el contenido de variables o expresiones durante el proceso.
- **Verificar Sintaxis:** La verificación de sintaxis permite realizar un seguimiento más detallado de la sintaxis del algoritmo y en caso de encontrar un error mostrará una descripción de la instrucción inválida y sugerencias de cómo solucionar dicho error sintáctico.

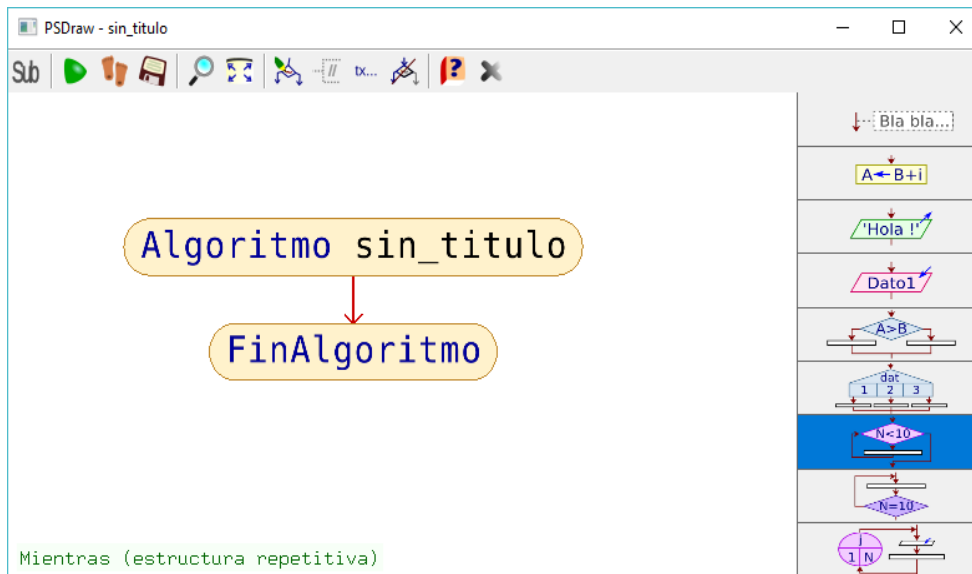
- **Menú Ayuda**

- **Índice:** Muestra un índice de contenidos con los temas de ayuda del interprete.



- **Ayuda Rápida:** Muestra una ventana en donde se puede introducir la palabra que hace referencia al tema de búsqueda y en caso de encontrar información muestra la ayuda correspondiente.
- **Ejemplos:** PSeInt incluye un conjunto de algoritmos de diferentes niveles de dificultad para ejemplificar la sintaxis y el uso del pseudocódigo.
- **Acerca de :** Muestra información de la aplicación.

Presenta una interfaz en la que se desarrolla los algoritmos con diagramas y los símbolos que corresponden a las distintas estructuras necesarias para su desarrollo:



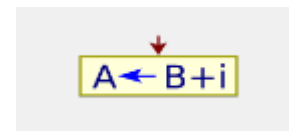
- **Comentario**

Texto libre que el intérprete ignora.



- **Asignación**

Se realiza la ASIGNACIÓN/DIMENSIÓN/DEFINICIÓN



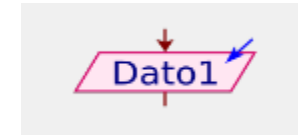
- **Entradas**

A través del cual se realiza la ESCRITURA (Instrucciones para generar salidas por pantalla)



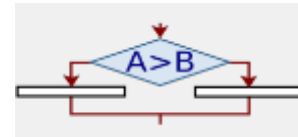
- **Salidas**

Permite realizar la LECTURA de datos desde el teclado (Instrucción para obtener entrada de Datos)



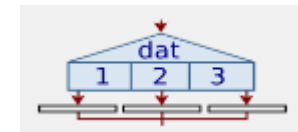
- **Condicional Si-Entonces**

Permite representar una Condicional Simple, para ejecutar algo por VERDAD o SI y algo por FALSO o NO



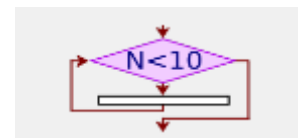
- **Selección múltiple (CASE)**

Permite de acuerdo a la opción seleccionada ejecutar algo SEGÚN (CASE - estructura de selección múltiple)



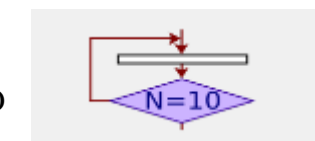
- **Lazo Mientras (WHILE)**

Permite ejecutar algo, MIENTRAS cumple una condición por VERDAD (Estructura repetitiva)



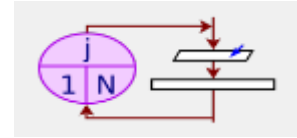
- **Lazo Repetir (REPEAT - UNTIL)**

Permite REPETIR HASTA QUE, toda vez que sea FALSO (Estructura repetitiva)



- **Lazo Para (FOR - TO)**

PARA (estructura repetitiva)



### 3. Aplicación

Se han desarrollado varios Vídeos y han sido colocados en forma ordenada a la plataforma educativa Moodle así como también subidos al canal de YouTube del docente para reforzar el aprendizaje de algoritmos, estos son:

- Video 1 - Familiarización para el ingreso a la plataforma educativa Moodle.

<https://www.youtube.com/watch?v=xzIn94v6azg&t=8s>

- Video 2 - Diagrama para la introducción o lectura de datos desde teclado, sentencia en pseudocódigo y código pascal.

<https://www.youtube.com/watch?v=jzuHAuL1XRA&t=49s>

- Video 3 - Diagrama para representar salida de datos por pantalla, pseudocódigo y código pascal.

<https://www.youtube.com/watch?v=5YFRLGqvd2s&t=4s>

- Video 4 - Asignación de Datos en PSeInt con diagramas, pseudocódigo y código Pascal.

<https://www.youtube.com/watch?v=aT1b14YZVpg&t=27s>

- Video 5 - Condición Simple representado con Diagramas en PSeInt, Pseudocódigo y código Pascal.

<https://www.youtube.com/watch?v=vLLE4fC2MQc&t=41s>

- Video 6 - Condicional simple con varias sentencias en PSeInt, pseudocódigo y código pascal.



<https://www.youtube.com/watch?v=RN4xr2woDZQ&t=29s>

- Video 7 - Condicional simple con varias sentencias por Verdad en PSeint, pseudocódigo y código pascal.

[https://www.youtube.com/watch?v=1qF\\_01RRrnM&t=22s](https://www.youtube.com/watch?v=1qF_01RRrnM&t=22s)

- Vídeo 8 - Condiciones anidadas múltiples en PSeInt, pseudocódigo y código pascal.

<https://www.youtube.com/watch?v=fqwYGy7vXJA>

- Video 9 - Condiciones múltiples CASE/ OF con PSeInt, pseudocódigo y código pascal.

<https://www.youtube.com/watch?v=0vStQA8y0Qo>

- Vídeo 10 - Conociendo la estructura FOR / TO / DO

<https://www.youtube.com/watch?v=ZXxjCpCiHAA&feature=youtu.be>

- Video 11 - Conociendo la estructura WHILE / DO

<https://www.youtube.com/watch?v=L4yUkjKeH1Y>

- Vídeo 12 - Estructura repetitiva condicional REPEAT / UNTIL.

<https://www.youtube.com/watch?v=Rp4btkdjy1k&feature=youtu.be>

- Video 13 - Algoritmo de saludo implementado en sus 3 formas diagrama, pseudocódigo y código.

<https://www.youtube.com/watch?v=bNcEGi2Wrlw&t=2s>

- Video 14 - Algoritmo implementado en sus 3 formas diagrama, pseudocódigo y código que permite leer datos del usuario como apellidos y nombres y muestra un mensaje de saludo.

<https://www.youtube.com/watch?v=EYqmo7J7rLc>

- Vídeo 15 - Cálculo de la edad en sus 3 formas diagrama, pseudocódigo y código.

<https://www.youtube.com/watch?v=m8uzoLYWkdo&feature=youtu.be>

- Video 16 - Operaciones matemáticas básicas de suma, resta, multiplicación y división (real y entera) en sus 3 formas diagrama, pseudocódigo y código.

<https://www.youtube.com/watch?v=QadalPqtEEo&feature=youtu.be>

- Video 17 - Verificación del mayor de 2 valores distintos en sus 3 formas diagrama, pseudocódigo y código.

<https://www.youtube.com/watch?v=2Z6sE0F3hXc&feature=youtu.be>

- Video 18 - Verificación de 2 valores si son iguales o cual es el mayor en sus 3 formas diagrama, pseudocódigo y código.

<https://www.youtube.com/watch?v=9uclSJ2EiME&feature=youtu.be>

- Video 19 – Verificación del mayor de 3 valores en sus 3 formas diagrama, pseudocódigo y código.

<https://www.youtube.com/watch?v=tvfxV-sHQBg&feature=youtu.be>

- Video 20 - Generación de serie par – ANALISIS.

<https://www.youtube.com/watch?v=RAK0jlBcKxM&feature=youtu.be>

- Video 21 - Generación serie par – IMPLEMENTACION.

<https://www.youtube.com/watch?v=WhzHCBDqhfo&feature=youtu.be>

- Video 22 - Generación de una serie con numeradores múltiplos de 2 y denominadores múltiplos de 3 – ANALISIS.

[https://www.youtube.com/watch?v=9SY\\_g8C\\_6fk&feature=youtu.be](https://www.youtube.com/watch?v=9SY_g8C_6fk&feature=youtu.be)

- Video 23 – Generación de una serie con numerador múltiplos de 2 denominadores múltiplos de 3 – IMPLEMENTACION.

<https://www.youtube.com/watch?v=Y62UUyWIGO0&feature=youtu.be>

- Vídeo 24 – Generación de una serie con números pares positivos y negativos de manera intercalada, usando SWITCHS/BANDERAS – ANALISIS.

<https://www.youtube.com/watch?v=DKzhq3ZxbCQ&feature=youtu.be>

- Video 25 - Generación de una serie con números pares positivos y negativos de manera intercalada, usando SWITCHS/BANDERAS IMPLEMENTACION.

<https://www.youtube.com/watch?v=dwFLraWUxLU&feature=youtu.be>

- Vídeo 26 - Generación de una serie con múltiplos de 3 positivos y negativos, suma de los mismos utilizando SWITCH/BANDERAS – ANALISIS.

<https://www.youtube.com/watch?v=WhLfqHXX7r0&feature=youtu.be>

- Vídeo 27 - Generación de una serie con múltiplos de 3 positivos y negativos, suma de los mismos utilizando SWITCH/BANDERAS – IMPLEMENTACIÓN.

<https://www.youtube.com/watch?v=163Kdi5769o&feature=youtu.be>

- Video 28 - Generación de una serie con múltiplos de 2 alternado con múltiplos de 5 negativos utilizando SWITCH/BANDERAS – ANALISIS

<https://www.youtube.com/watch?v=qqc8uSbyMjQ&feature=youtu.be>

- Vídeo 29 - Generación de una serie con múltiplos de 2 alternado con múltiplos de 5 negativos utilizando SWITCH/BANDERAS – IMPLEMENTACIÓN.

<https://www.youtube.com/watch?v=1jveU2ujL7o>

- Vídeo 30 - Lote de valores numéricos, sin secuencia lógica – ANALISIS.

<https://www.youtube.com/watch?v=GeLHyfGjvkY&feature=youtu.be>

- Video 31 - Lote de valores numéricos, sin secuencia lógica usando estructura repetitiva WHILE/DO – IMPLEMENTACIÓN

<https://www.youtube.com/watch?v=R79egKawk3o&t=7s>

- Video 32 - Lote de valores numéricos, sin secuencia lógica usando estructura repetitiva FOR/TO/DO – IMPLEMENTACIÓN

<https://www.youtube.com/watch?v=d6dUSUi9TI&feature=youtu.be>

- Video 33 - Lote de valores numéricos, sin secuencia lógica y se cuenta valores negativos usando estructura repetitiva WHILE/DO – ANALISIS

<https://www.youtube.com/watch?v=dJICU5oGP-E&feature=youtu.be>

- Video 34 - Lote de valores numéricos, sin secuencia lógica y se cuenta valores negativos usando estructura repetitiva WHILE/DO – IMPLEMENTACIÓN

[https://www.youtube.com/watch?v=Qh6p\\_g7NQJI&feature=youtu.be](https://www.youtube.com/watch?v=Qh6p_g7NQJI&feature=youtu.be)

- Video 35 - Lote de valores numéricos, sin secuencia lógica y se suma y cuenta valores pares usando estructura repetitiva WHILE/DO – ANALISIS

<https://www.youtube.com/watch?v=K5238PIXMNU&feature=youtu.be>

- Video 36 - Lote de valores numéricos, sin secuencia lógica y se suma y cuenta valores pares usando estructura repetitiva WHILE/DO – IMPLEMENTACIÓN

<https://www.youtube.com/watch?v=pAMDyduiaSE&feature=youtu.be>

En cuanto al software PSeInt se debe tomar en cuenta que para por cada aplicación se debe realizar el diagrama y pseudocódigo

Imagen 1: PSeInt interfaz de creación de pseudocódigo

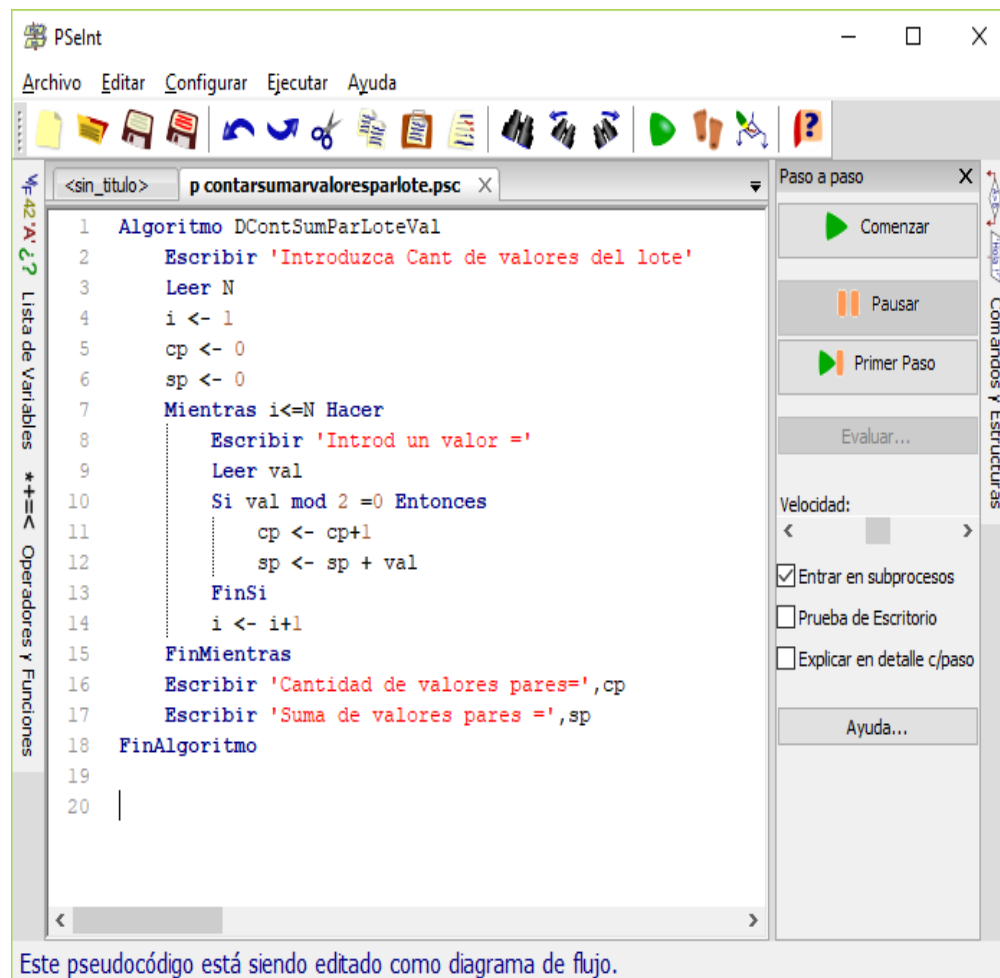


Imagen 2: PSeInt interfaz de creación de Diagramas

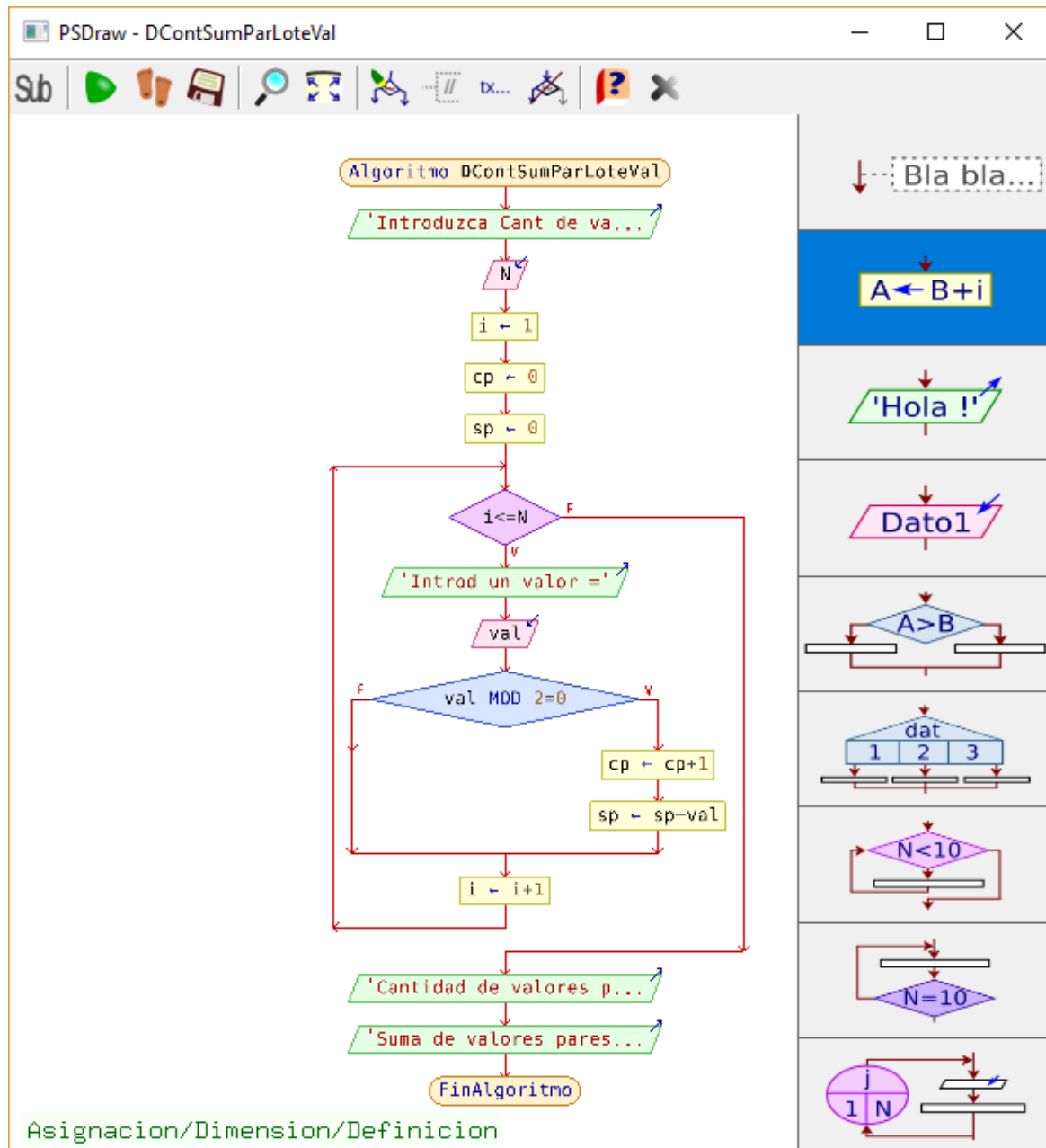
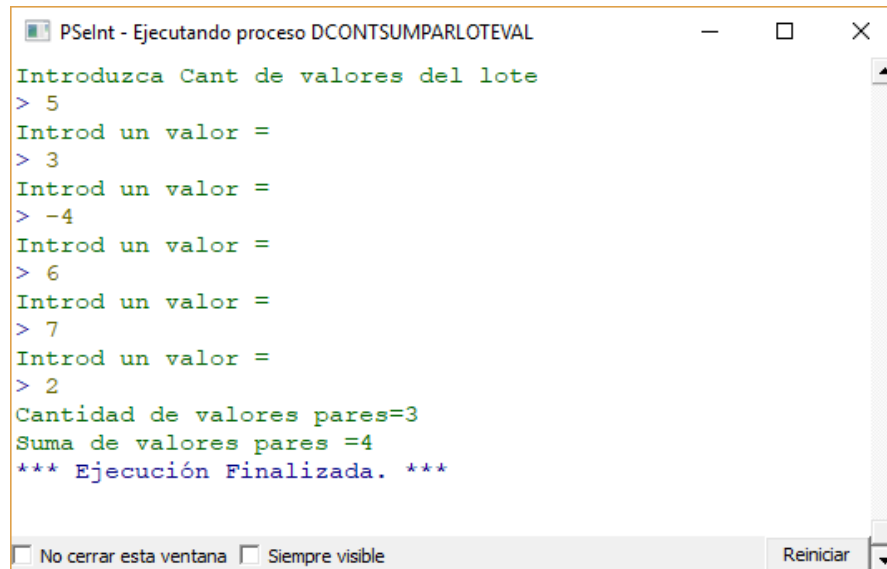


Imagen 3: PSeInt interfaz de ejecución de un algoritmo



```
PSeInt - Ejecutando proceso DCONTSUMPARLOTEVAL
Introduzca Cant de valores del lote
> 5
Introd un valor =
> 3
Introd un valor =
> -4
Introd un valor =
> 6
Introd un valor =
> 7
Introd un valor =
> 2
Cantidad de valores pares=3
Suma de valores pares =4
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

Imagen 4: PSeInt interfaz de Acerca de

