



UNIVERSIDAD TECNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

Carrera:

Computación

Materia:

Fundamentos computacionales

Nombres y Apellidos:

Nixon Javier Vuele Irene.

Cedula:

1104384811

Magister:

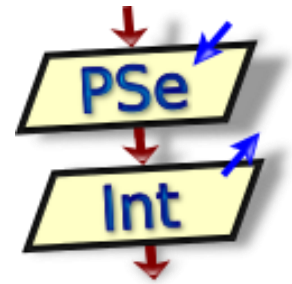
Ingeniero Rene Elizaldes

Período académico:

Octubre 2020 - febrero 2021

HISTORIA PSEINT

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. (Carolina, 2017)

¿Quién lo creo?

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. (Carolina, 2017)

Descripción

PSeInt es la abreviatura de los estados de computación de PSeudo Intérprete, una herramienta educativa creada en Argentina, utilizada principalmente por estudiantes para aprender los fundamentos de la programación y el desarrollo de la lógica. Es un software muy popular de su tipo y es ampliamente utilizado en universidades de Latinoamérica y España.

Utiliza pseudocódigo para la solución de algoritmos.

Propósito de PSeInt

PSeInt está pensado para asistir a los estudiantes que se inician en la construcción de programas o algoritmos computacionales. El pseudocódigo se suele utilizar como primer contacto para introducir conceptos básicos como el uso de estructuras de control, expresiones, variables, etc., sin tener que lidiar con las particularidades de la sintaxis de un lenguaje real. Este software pretende facilitarle al principiante la tarea de escribir algoritmos en este pseudolenguaje presentando un conjunto de ayudas y asistencias, y brindarle además algunas herramientas adicionales que le ayuden a encontrar errores y comprender la lógica de los algoritmos.

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.

Características y Funcionalidades de PSeInt:

- 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 en 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) (PSeInt, 2015)

Hay **funciones ya integradas** en PSeInt, casi todas matemáticas, son las siguientes:

FUNCIÓN	SIGNIFICADO
RC(X) o RAIZ(X)	Raíz cuadrada de X
ABS(X)	Valor absoluto de X
LN(X)	Logaritmo natural de X
EXP(X)	Función exponencial de X
SEN(X)	Seno de X
COS(X)	Coseno de X
TAN(X)	Tangente de X
ASEN(X)	Arcoseno de X
ACOS(X)	Arcocoseno de X
ATAN(X)	Arcotangente de X
TRUNC(X)	Parte entera de X
REDON(X)	Entero más cercano a X
AZAR(X)	Entero aleatorio en el rango [0;x-1]
ALEATORIO(A,B)	Entero aleatorio en el rango [A;B]
LONGITUD(S)	Longitud de la cadena S
MAYUSCULAS(S)	Devuelve una copia de la cadena S en mayúsculas
MINUSCULAS(S)	Devuelve una copia de la cadena S en minúsculas

FUNCIÓN

SIGNIFICADO

SUBCADENA(S,X,Y)	Devuelve una nueva cadena que consiste en la parte de la cadena S que va desde la posición X hasta la posición Y (incluyendo ambos extremos)
CONCATENAR(S1,S2)	Devuelve una cadena nueva resultado de unir las cadenas S1 y S2
CONVERTIRANUMERO(X)	Convierte a número la variable X
CONVERTIRATEXTO(S)	Convierte a texto la variable S

(Tere, 2020)

Las palabras reservadas de PsInt son:

Son palabras propias en la sintaxis de un lenguaje de programación y no puede ser usada como el nombre de una variable.

En PsInt son las siguientes:

proceso
inproceso
definir
como
logico
entero
real
caracter
Si
entonces
Sino
Para
paso
hacer
mientras
finSi
finpara
finMientras
Subproceso
repetir
Escribir
leer
Dimension
Borra Pantalla

Estas palabras no importan si se escriben en mayúsculas o minúsculas.

(colin, 2016)

Estructuras de control

Básicamente, **las estructuras de control en Pseint** se dividen en **condiciones y repetitivas:**

Condicionales:

Condicional Si-Entonces: La secuencia de instrucciones ejecutadas por la instrucción Si-Entonces-Sino dependerá del valor de una condición lógica. Ejemplo:

Si <condición>

Entonces

<instrucciones>

Sino

<instrucciones>

FinSi

Cuando se ejecuta esta instrucción, se evalúa la condición y se ejecutan las instrucciones que sean necesarias: las instrucciones que le siguen al **“Entonces”** si la condición es verdadera, o las instrucciones que le siguen al **“Sino”** si la condición es falsa. La condición deberá ser una expresión lógica, que al ser evaluada retorna un valor **“Verdadero”** o **“Falso”**.

En este punto, la cláusula **“Entonces”** debe aparecer siempre, pero la cláusula **“Sino”** puede no estar presente. En ese caso, si la condición es falsa no se ejecuta instrucción alguna y la ejecución del programa continúa con la instrucción que sigue.

Selección Multiple: La secuencia de instrucciones ejecutada por una instrucción **“Según”** depende del valor de una variable numérica. Ejemplo:

Segun <variable> Hacer

<número1>: <instrucciones>

<número2>,<número3>: <instrucciones>

<...>

De Otro Modo: <instrucciones>

FinSegun

En este punto, la mencionada instrucción permite ejecutar de manera opcional varias acciones posibles, **dependiendo del valor que se haya almacenado en una variable de tipo numérico**. Al

ejecutarse, se evalúa el contenido de la variable y se ejecuta la secuencia de instrucciones asociada con dicho valor.

Cada opción está formada por uno o más números separados por comas, dos puntos y una secuencia de instrucciones. Si una opción incluye varios números, la secuencia de instrucciones asociada se debe ejecutar **cuando el valor de la variable es uno de esos números**.

De forma opcional, también se puede añadir una opción final, denominada **“De Otro Modo”**, cuya secuencia de instrucciones asociada se ejecutará sólo si el valor que se haya almacenado en la variable no coincide con ninguna de las opciones anteriores.

Repetitivas

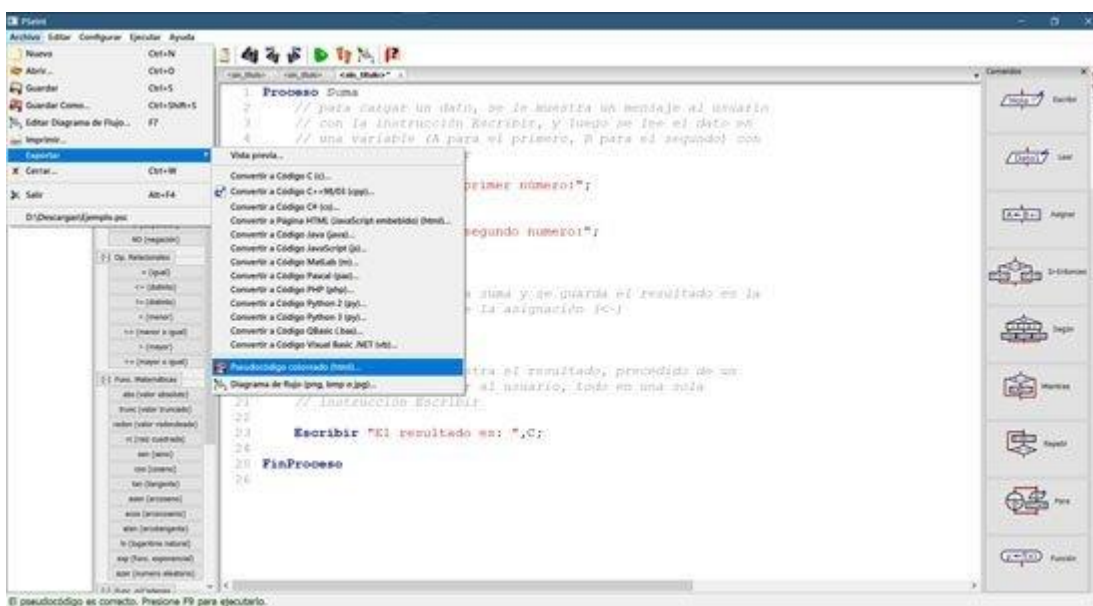
Lazos Mientras: La instrucción **“Mientras”** ejecuta una secuencia de instrucciones mientras una condición sea verdadera. Ejemplo:

Mientras <condición> Hacer

<instrucciones>

FinMientras

Cuando esta instrucción se ejecuta, la condición se evalúa para comprobar si resulta **“Verdadera”**, en ese caso, se ejecuta una vez la secuencia de instrucciones que forman el cuerpo del ciclo. Al finalizar la ejecución del cuerpo del ciclo se vuelve a evaluar la condición y, si es **“Verdadera”**, la ejecución se repite. Estos pasos se repiten mientras la condición sea la misma, es decir verdadera. Cabe destacar en este punto que las instrucciones del cuerpo del ciclo pueden no ejecutarse nunca, si al evaluar por primera vez la condición resulta ser **“Falsa”**.



Si la condición siempre es verdadera, **al ejecutar esta instrucción se produce un ciclo infinito**. A fin de evitarlo, las instrucciones del cuerpo del ciclo deben contener alguna instrucción que modifique la o las variables involucradas en la condición, de modo que ésta sea falsificada en algún momento y así finalice la ejecución del ciclo.

Lazos Repetir: La instrucción “**Repetir-Hasta Que**” ejecuta una secuencia de instrucciones hasta que la condición sea verdadera. Ejemplo:

Repetir

<instrucciones>

Hasta Que <condición>

Al ejecutarse esta instrucción, la secuencia de instrucciones que forma el cuerpo del ciclo se ejecuta una vez y luego se evalúa la condición para comprobar si la misma es verdadera o falsa. Si la condición resulta “**Falsa**”, entonces el cuerpo del ciclo se ejecuta nuevamente y se vuelve a evaluar la condición. Esto se repite hasta que la condición sea “**Verdadera**”.

Cabe destacar que debido a que la condición en este caso se evalúa al final, las instrucciones del cuerpo del ciclo serán ejecutadas al menos una vez. Por ello, con el propósito de evitar ciclos infinitos, **el cuerpo del ciclo debe contener alguna instrucción que modifique la o las variables involucradas en la condición** de modo que en algún momento la condición sea verdadera y se finalice la ejecución del ciclo.

El caso que el usuario utilice sintaxis flexible en las opciones de PSeInt se permite opcionalmente utilizar “**Mientras Que**” en lugar de “**Hasta Que**”, de modo que el conjunto de acciones contenidas en el bucle se ejecuta mientras que la condición sea “**Verdadera**”. En este punto, cabe destacar que la palabra “**Que**” es el elemento que diferencia el uso de la palabra “**Mientras**” en la estructura repetir de la estructura “**Mientras**”. Esto significa que en el caso de omitirse la palabra que se considera como el comienzo de un bucle “**Mientras**” en lugar del final de un bucle “**Repetir**”.

Para Cada <elemento> De <Arreglo> Hacer ...

En este caso, el segundo identificador debe corresponder a un arreglo. El primero será el que variará en cada iteración. **El ciclo realizará tantas iteraciones como elementos contenga el arreglo**, y en cada uno el primer identificador servirá para referirse al elemento del arreglo al que se refiere. (Marker, 2020)

Bibliografía

Carolina, D. (20 de abril de 2017). *Programación con PSeInt*. Obtenido de Historia:

<https://lobasicodepseint.blogspot.com/2017/04/historia.html#:~:text=HISTORIA%20PSEINT,los%20laboratorios%20de%20la%20facultad.>

colin, U. d. (16 de Septiembre de 2016). *Facebook*. Obtenido de PALABRAS RESERVADAS DE PSEINT:

<https://www.facebook.com/861377833962762/posts/palabras-reservadas-de-pseint-palabras-reservadas-en-psintson-palabras-propia-en/866300590137153/>

Marker, G. (15 de Julio de 2020). *Tecnología+Informática*. Obtenido de PSeInt: ¿Qué es? Descargar y usar PSeInt. Ejemplos: <https://www.tecnologia-informatica.com/pseint/>

PSeInt. (1 de Septiembre de 2015). *PSeInt*. Recuperado el 09 de Enero de 2021, de Características y Funcionalidades de PSeInt: <http://pseint.sourceforge.net/index.php?page=features.php>

Tere. (14 de Junio de 2020). *Aprendiendo a programar programando*. Obtenido de Funciones en PSeInt: <https://terepebernal.com/blog/pseudocodigo/algoritmos-pseint-lo-mas-basico/>