

PSeInt:

programando en pseudocódigo



MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

Dirección General de Evaluación y Cooperación Territorial

Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF)

Recursos Educativos Digitales

Junio 2023

NIPO (web) 847-22-068-1

ISSN (web) 2695-4176

DOI (web) 10.4438/2695-4176_OTE_2019_847-19-121-5

NIPO (formato html) 847-20-116-0

NIPO (formato pdf) 847-20-115-5

DOI (formato pdf) 10.4438/2695-4176_OTEpdf103_2020_847-19-134-3

PSeInt: programando en pseudocódigo

Por Rafael Morales Caumel para INTEF

<https://intef.es>

Obra publicada con licencia de Creative Commons



Reconocimiento-Compartir Igual 4.0 Licencia Internacional.

<https://creativecommons.org/licenses/by-sa/4.0/>



Para cualquier asunto relacionado con esta publicación contactar con:
Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado
C/Torrelaguna, 58. 28027 Madrid.
Tfno.: 91-377 83 00. Fax: 91-368 07 09
Correo electrónico: recursos.educativos@educacion.gob.es



 EducaMadrid Rafael
Morales Caumel
 Scratch tecnoalumnos



El autor de este artículo

Rafael Morales Caumel es Ingeniero Técnico Industrial con más de veinte años de experiencia como funcionario de carrera del cuerpo de profesores de Enseñanza Secundaria en la especialidad de Tecnología.

En su trayectoria profesional cuenta con numerosos cursos, seminarios y ponencias impartidas, así como diferentes publicaciones en revistas especializadas. Se considera un apasionado de la Electrónica y la Robótica Educativa, pasión que intenta transmitir en sus clases a todos sus alumnos y alumnas. Ha formado parte del jurado calificador del "Premio Nacional de Iniciación a la Investigación Tecnológica" y desde 2019 colabora con la Editorial McGraw Hill en la publicación de sus textos educativos para Tecnología.

Actualmente pertenece al Cuerpo de Catedráticos de Enseñanza Secundaria y desarrolla su actividad docente en el IES Gran Capitán (Madrid).



Introducción

Este artículo está dedicado a PSeInt, una herramienta que nos ofrece la posibilidad de programar un algoritmo empleando instrucciones sencillas en castellano y facilitar así la transición a otros lenguajes de programación textuales. Con ayuda de esta herramienta, los docentes podrán abordar algunas de las competencias específicas relacionadas con el pensamiento computacional y el razonamiento lógico-matemático en materias como Tecnología y Digitalización en la etapa de Educación Secundaria o en otras materias optativas.



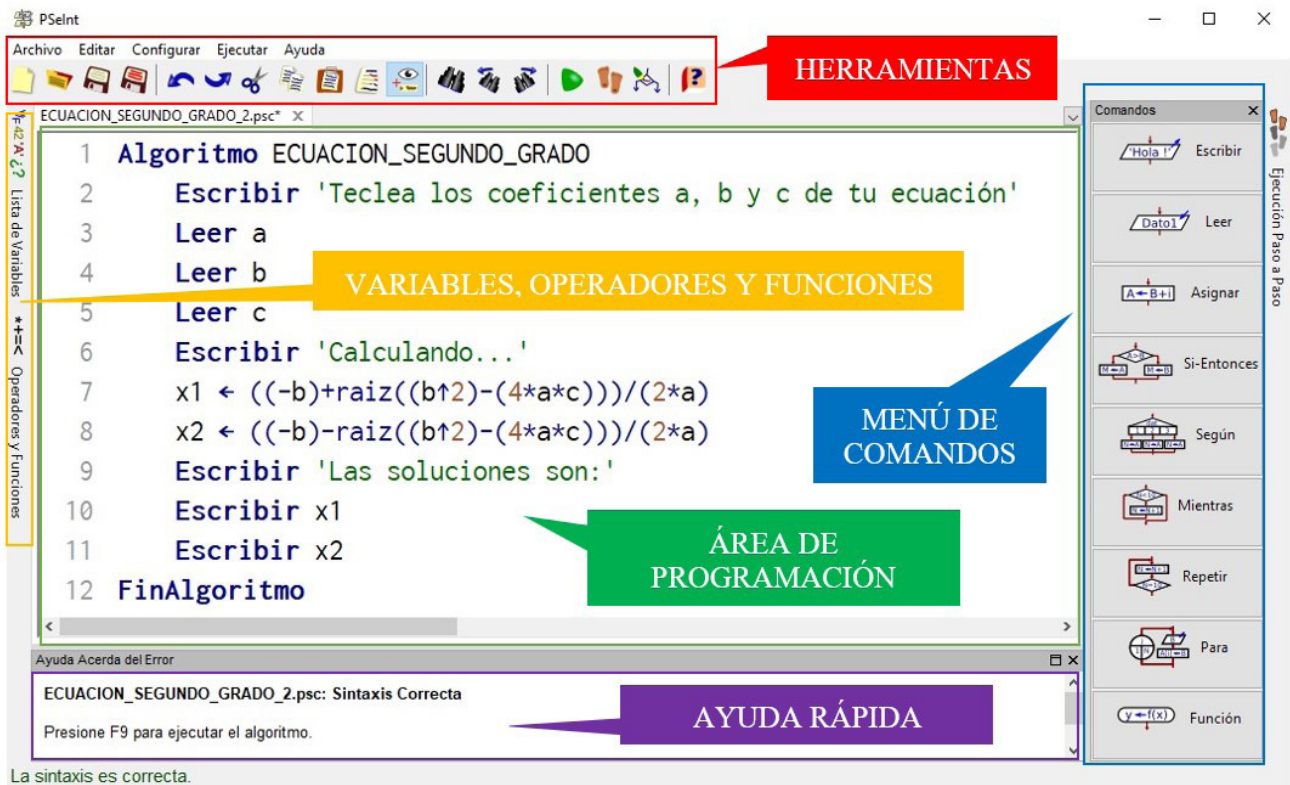
La Herramienta

Los lenguajes de programación son, hoy día, el nuevo idioma a estudiar desde edades muy tempranas. El punto de partida para el estudio de un lenguaje de programación pasa inevitablemente por los **lenguajes gráficos o de bloques** que facilitan su estudio al mostrar de forma visual las instrucciones del algoritmo programado. Un ejemplo muy conocido es el famoso **lenguaje gráfico Scratch** utilizado ampliamente en el ámbito educativo para iniciarse en la programación.

Sin embargo, el salto al desarrollo de algoritmos en lenguajes de programación textuales (Python, C, Java...) puede resultar complicado, aunque se haya adquirido cierta destreza en la programación por bloques. Por esta razón, los **lenguajes de pseudocódigo** son una herramienta muy útil para expresar la secuencia de instrucciones de un algoritmo empleando un lenguaje intermedio entre el lenguaje natural y el lenguaje de programación cuya sintaxis suele ser, por lo general, muy rígida y poco amigable.

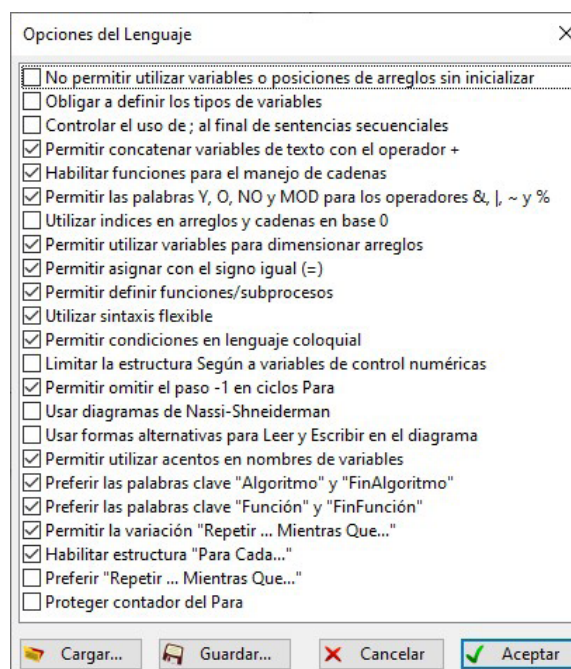
PSeInt es la abreviatura de "Intérprete de Pseudocódigo" y es una herramienta enfocada a facilitar el desarrollo de programas escritos en un lenguaje de programación textual mediante un sencillo e intuitivo pseudolenguaje en idioma castellano, pero con la particularidad de utilizar una sintaxis flexible e incluso configurable por el propio usuario. Con PSeInt, el alumno o alumna puede escribir un programa en pseudocódigo de manera sencilla gracias a las ayudas y asistencias que ofrece sin preocuparse por las estrictas reglas de sintaxis que cualquier lenguaje de alto nivel exige en su escritura. Por ejemplo, podríamos crear sencillos algoritmos para calcular el área de diferentes figuras geométricas elegidas por el usuario, o determinar las raíces de una ecuación cuadrática a partir de los coeficientes de dicha ecuación, por poner dos simples ejemplos.

Este software es multiplataforma y puede descargarse de su [web oficial](#).



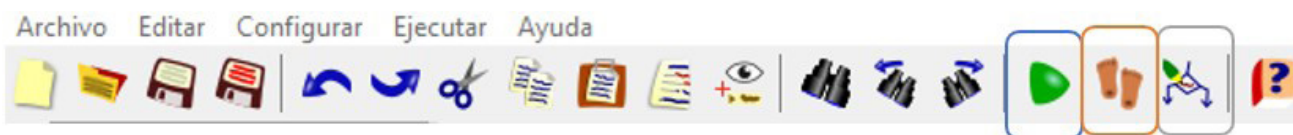
Captura de pantalla de PSeInt

Una vez abierto (ya sea en su versión instalable o versión portable) nos encontramos con una interfaz muy sencilla e intuitiva con varias ventanas y una ventana emergente para configurar la sintaxis y su grado de flexibilidad. Por ejemplo, en esta configuración podemos seleccionar si debemos definir previamente el tipo de variables o controlar el uso de ";" al final de cada instrucción (como ocurre con el lenguaje de programación C). Todas estas opciones podemos seleccionarlas a gusto del programador y guardar dicha configuración para su uso posterior.



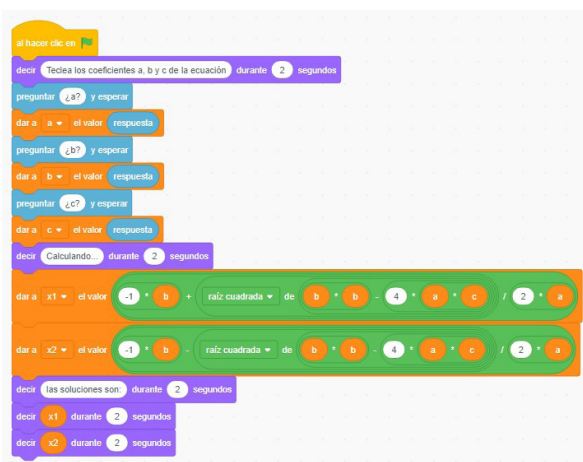
Opciones de configuración

La interfaz de programación dispone de **nueve comandos básicos de programación** a seleccionar en los botones del menú lateral derecho que permiten establecer estructuras de programación secuenciales, condicionales y bucles, así como funciones (subprocesos) para descomponer un algoritmo complejo en elementos más sencillos. El menú izquierdo ofrece un catálogo de funciones matemáticas y operadores algebraicos, lógicos y relacionales. Finalmente, la ventana inferior informará de posibles errores durante la ejecución del programa señalando la línea de código que genera el problema. La barra de herramientas dispone de la ejecución del programa “paso a paso” que facilita la comprensión de la secuencia de instrucciones. Además, una utilidad muy interesante es la posibilidad de visualizar el **diagrama de flujo** del algoritmo programado gracias a la opción disponible en la barra de herramientas:



Barra de herramientas

En el ejemplo adjunto podemos comparar el algoritmo escrito en el lenguaje gráfico (Scratch) y su equivalente en lenguaje de pseudocódigo (PSeInt). Puede observarse las instrucciones básicas “Leer” y “Escribir” y la asignación a las variables x1 y x2 de los valores calculados. Todo ello, insertando las instrucciones por teclado, aunque también puede optarse por hacer clic en el menú de comandos lateral y completar los parámetros de cada instrucción.



Algoritmo en Scratch

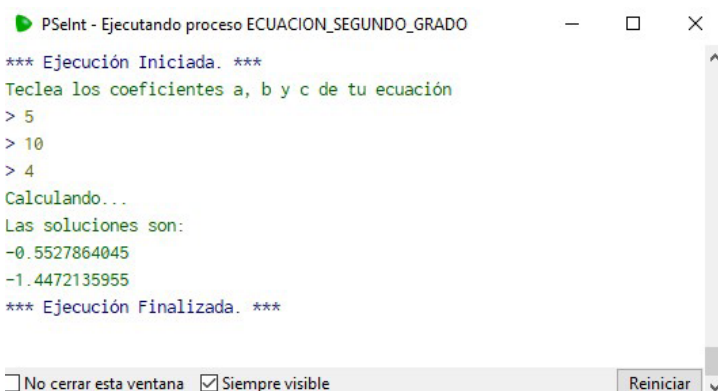
```

1 Algoritmo ECUACION_SEGUNDO_GRADO
2   Escribir 'Teclea los coeficientes a, b y c de tu ecuación'
3   Leer a
4   Leer b
5   Leer c
6   Escribir 'Calculando...'
7   x1 = ((-b)+raiz((b^2)-(4*a*c)))/(2*a)
8   x2 = ((-b)-raiz((b^2)-(4*a*c)))/(2*a)
9   Escribir 'Las soluciones son:'
10  Escribir x1
11  Escribir x2
12 FinAlgoritmo
  
```

Algoritmo en PSeInt

Haciendo clic en ejecutar, aparecerá una ventana emergente con la ejecución del programa solicitando los datos de entrada y mostrando los datos de salida. En el ejemplo, el algoritmo calcula las raíces de una ecuación de segundo grado:

$$ax^2 + bx + c = 0$$



Ventana emergente de ejecución



Explicación del uso en el ámbito educativo

Esta herramienta es adecuada para trabajar la introducción de la programación textual antes de abordar la didáctica de lenguajes de programación de alto nivel como Python o C++. Por ello, es aconsejable su inclusión como paso intermedio entre un lenguaje gráfico y un lenguaje textual, de manera que, una vez que nuestro alumnado ya conoce las estructuras básicas de programación de forma visual, podemos ofrecerle esta herramienta para que se familiarice con estas mismas estructuras, pero escritas por ellos mismos en texto. Además, como docentes seleccionaremos para nuestro alumnado el grado de flexibilidad en la sintaxis que deben escribir. Por ejemplo, en cursos iniciales de ESO, podemos proponer una flexibilidad casi total, de manera que no sea necesario definir el tipo de variables a utilizar o la inicialización de las mismas, resultando muy cómodo la programación del código como si de una redacción de texto se tratará. Posteriormente, puede aumentarse los requisitos de la sintaxis y en cursos más avanzados de ESO o Bachillerato es más aconsejable proponer una configuración algo más exigente, para que el salto al lenguaje de programación posterior sea más suave.

Esta herramienta es gratuita, sin registro alguno y se ofrece para varios sistemas operativos tanto para instalar como portable por lo que su uso es inmediato con el alumnado. Además, no requiere de un hardware exigente.



Opciones de descarga de PSeInt

PSeInt permite ejecutar el algoritmo y modificarlo mientras se ejecuta observando los cambios que se producen. La interfaz ofrece diferentes tipos de ayudas mientras se escribe como el autocompletado, la indentación y plantillas para los comandos básicos con sus correspondientes descripciones sin olvidar un completo manual de ayuda con ejemplos solucionados que pueden emplearse como prácticas guiadas para el alumnado.

```
<sin_titulo>* [X]
1  Proceso sin_titulo
2      Escribir "Ingrese un número:"
3      Leer N
4      Si expresion_logica Entonces
5 +   ..... acciones_por_verdadero
6      Sino
7 +   ..... acciones_por_falso
8      Fin Si
9  FinProceso
10
```

Plantilla del comando condicional

La herramienta puede empezarse a trabajar explicando algunos algoritmos sencillos a través de los diagramas de flujo y el pseudocódigo de texto correspondiente y para el resto de la sesión se pueden proponer actividades con enunciados similares a los descritos. La duración total de las sesiones dependerá del nivel de programación que deseamos alcanzar con nuestro alumnado y el nivel adquirido previamente en entornos gráficos.

Por ejemplo, para niveles de 1.º a 3.º ESO el manejo de estructuras condicionales y bucles son objetivos que pueden alcanzarse con garantías en pocas sesiones, mientras que el desarrollo de funciones (subprocesos en PSeInt) y listas (arreglos en PSeInt) son más aconsejables en cursos superiores como 4.º ESO y Bachillerato.

En cualquier caso, estas actividades deberán ser de dificultad creciente proponiendo ejercicios sencillos que vayan incorporando las diferentes estructuras de programación (disponibles en el menú de comandos) hasta llegar a la creación de funciones y listas como estructuras de programación más complejas, si el objetivo final es ese.

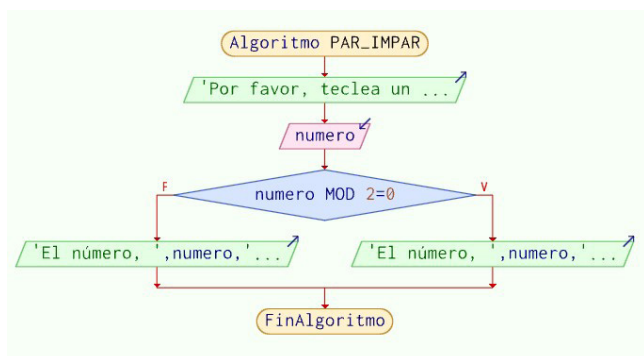


Diagrama de flujo algoritmo PAR-IMPAR

```

1  Algoritmo PAR-IMPAR
2  Escribir 'Por favor, teclea un número'
3  Leer numero
4  Si numero MOD 2=0 Entonces
5  |   Escribir 'El número, ',numero,' es par'
6  SiNo
7  |   Escribir 'El número, ',numero,' es impar'
8  FinSi
9  FinAlgoritmo

```

Pseudocódigo algoritmo PAR-IMPAR



Metodología y Didáctica Aplicada

Como en cualquier proceso de aprendizaje de un lenguaje de programación, PSeInt es una herramienta que facilita a nuestro alumnado el protagonismo de su aprendizaje con la figura del profesorado como dinamizador de un proceso de enseñanza-aprendizaje basado en el desarrollo de **prácticas guiadas** y adaptadas a las capacidades y necesidades personales de cada alumno y alumna. Este **autoaprendizaje** implica dos fases:

- ▶ **Primera fase.** Cada estudiante realiza las actividades solucionadas y comprueba los resultados de los algoritmos, adquiriendo las destrezas necesarias para comprender las estructuras de programación propuestas y la sintaxis flexible que ofrece el lenguaje de pseudocódigo.
- ▶ **Segunda fase.** En esta fase, interviene una **metodología no directiva** donde el alumnado debe buscar soluciones a una propuesta final planteada por su docente como actividad global, en este caso un algoritmo que resuelva un problema planteado. En esta fase el trabajo en **grupos cooperativos** es esencial para alcanzar la solución a la propuesta final.

Como complemento a la didáctica propuesta, puede añadirse una **fase final** en la que los grupos de trabajo cooperativos den a conocer los algoritmos programados en una **exposición oral** al resto de compañeros y compañeras de manera que toda la clase sea conocedora de las diferentes soluciones adoptadas en la práctica final. Esta mini **clase invertida** propuesta (*flipped classroom*) permitiría enriquecer los conocimientos adquiridos a través de las estrategias adoptadas por otros grupos de trabajo en la misma clase.

Con todo lo expuesto anteriormente, podemos afirmar que con esta herramienta logramos abordar algunas competencias clave como la competencia matemática, la competencia en ciencia, tecnología e ingeniería, la competencia personal, social y de aprender a aprender y por supuesto la competencia digital.



Valoración Personal

En mi opinión, esta herramienta es muy útil, con una curva de aprendizaje rápida y es perfecta para rellenar el hueco existente entre los lenguajes gráficos y los lenguajes textuales ya que el salto de un lenguaje de bloques como Scratch hacia un lenguaje de programación puede resultar difícil. En este sentido, el alumnado puede sentirse cómodo visualizando una estructura condicional en Scratch y utilizar un ratón para “pinchar y arrastrar” pero no tanto si esta estructura debe escribirse por teclado y mostrar las sentencias de la selección entre llaves, por ejemplo. Obviamente, esto suele ocurrir en edades muy tempranas, donde la escritura del código sin errores de sintaxis ya de por sí es una dificultad añadida, frente a la mecánica de *click and drag* donde los bloques de código encajan unos con otros, si ello es posible.



Recomendación final

Recomiendo el uso de esta herramienta a todos los docentes que imparten materias relacionadas con la Tecnología y Digitalización y otras materias y optativas que incluyen contenidos relacionados con el pensamiento computacional como en el caso de Matemáticas. Al tratarse de pseudocódigo, no es necesario ser un experto programador para incorporar esta herramienta en algún momento del curso. Los comandos son muy intuitivos y la documentación que se facilita en la propia web es más que suficiente. Aun así, la comunidad internauta ofrece una gran variedad de documentación, prácticas guiadas y videotutoriales que harán más fácil el uso de esta aplicación, en caso de decidirse a utilizarla.



Información y materiales complementarios

- ▶ [Web oficial de PSeInt.](#)
- ▶ [Ejercicios resueltos de algoritmos con PSeInt.](#)
- ▶ <https://aprenderaprogramar.pro/category/pseint/>
- ▶ [Curso Pseudocódigo Con PSeint de Disco Duro de Roer.](#)

Derechos de uso

- § Todas las marcas nombradas en el artículo son nombres y/o marcas registradas por sus correspondientes propietarios.
- § Las imágenes han sido proporcionadas por el autor. Algunas de ellas corresponden a capturas de pantalla de la herramienta.
- § El texto ha sido elaborado por el autor expresamente para este artículo.

