

MultiPseudo, lenguaje de programación en pseudocódigo multi-idioma

Antonio López García
LITE – Laboratorio de Tecnologías de la
Información en la Educación
Universidad Rey Juan Carlos
Madrid, España
a.lopezgz.2018@alumnos.urjc.es

Jaime Urquiza-Fuentes
LITE – Laboratorio de Tecnologías de la
Información en la Educación
Universidad Rey Juan Carlos
Madrid, España
jaime.urquiza@urjc.es

Resumen

Usar el pseudocódigo al comienzo de la enseñanza de programación puede proporcionar ventajas a los alumnos, frente a comenzar directamente con un lenguaje de alto nivel. Esto puede que se deba al uso de la lengua materna y una sintaxis sencilla. MultiPseudo es un intérprete de pseudocódigo presentado como software educativo libre, pensado para su uso en la enseñanza de la programación. Permite realizar algoritmos en pseudocódigo utilizando 40 instrucciones en varios idiomas. Por el momento se han incluido las siguientes lenguas latinas de pseudocódigo: Castellano, Catalán, Francés, Gallego, Italiano, Portugués y Rumano. Además, se han incluido las especificaciones de Castellano Natural, mostrando algunas diferencias con el pseudocódigo Castellano Clásico, que incluye ciertas palabras clave obtenidas como resultado de un estudio previo, seleccionadas como preferidas por los estudiantes. Permite la traducción de algoritmos entre los diferentes lenguajes de pseudocódigo, así como la traducción del código a Python 2. MultiPseudo también permite la programación y desarrollo de algoritmos en Python 2, lo que contribuye a mejorar su idoneidad como herramienta de aprendizaje. Además, la herramienta contempla la posibilidad de evaluar y añadir nuevos lenguajes de pseudocódigo definidos por el usuario.

Abstract

Using pseudocode early in teaching programming can provide advantages to students over starting directly with a high-level language. This may be due to the use of the mother tongue and simple syntax. MultiPseudo is a pseudocode interpreter presented as free educational software, intended for use in teaching programming. It allows performing algorithms in pseudocode using 40 instructions in several languages. At the moment, the following pseudocode Latin languages have been included: Spanish, Catalan, French, Galician, Italian, Portuguese and Romanian. In addition, the Natural Spanish specifications

have been included, showing some differences with the Classic Spanish pseudocode, which includes certain key words obtained as a result of a previous study, selected as preferred by the students. It allows the translation of algorithms between the different pseudocode languages, as well as the translation of the code to Python 2. MultiPseudo also allows the programming and development of algorithms in Python 2, which contributes to improve its suitability as a learning tool. In addition, the tool contemplates the possibility of evaluating and adding new pseudocode languages defined by the user.

Palabras clave

Pseudocódigo, lenguaje de alto nivel, lengua materna, educación, programación, intérprete.

Introducción

Es normal encontrarnos con una serie de dificultades a la hora de aprender a programar [3], tales como desconocer la sintaxis del lenguaje de programación, desconocer la herramienta utilizada, las palabras clave en otro idioma, etc. Podría ser un principio lógico iniciarse con el pseudocódigo, ya que nos permite construir la solución de un problema en un lenguaje natural y con una sintaxis simple [2].

De hecho, en la enseñanza de programación es habitual, como un paso necesario, codificar un algoritmo en pseudocódigo, previamente a la utilización de un lenguaje de alto nivel [4].

Un planteamiento importante, podría ser buscar el lenguaje que resulta más idóneo para el aprendizaje de la programación, que presente menos problemas a los estudiantes iniciales.

Koulouri, Lauria y Macredie [5] informan que el uso de un lenguaje sintácticamente simple como Python, en lugar de uno más complejo como Java, facilita el aprendizaje de los conceptos de programación a los estudiantes iniciales. A los programadores iniciales les resulta más sencillo aprender con un lenguaje simple como Python frente a otro lenguaje

sintácticamente más complejo como Java [7]. El lenguaje Python, con su simplicidad de sintaxis, parecería enfocarse como una aproximación al pseudocódigo en lengua inglesa.

El uso de la lengua materna es sin duda un factor importante que puede repercutir en el aprendizaje y puede guardar una relación directa con la capacidad que tiene el alumno de prosperar en sus estudios.

Así, hay trabajos que establecen una relación entre el fracaso escolar y el lenguaje [9]. Tras la realización del estudio, en 13 colegios públicos de EGB (educación primaria), los autores concluyen que el lenguaje está íntimamente relacionado con el rendimiento escolar, por lo que sería posible elaborar un programa de intervención, para evitar el fracaso escolar.

Un estudio realizado en Nueva York [8], muestra una investigación en la que se identifican los factores que influyen en el rendimiento académico de los hijos de inmigrantes mexicanos residentes. Sus autores concluyen, que los alumnos con mayor conocimiento del idioma obtienen un mejor rendimiento académico. Se aprecia claramente, como el factor idioma influye directamente en el rendimiento escolar.

El pseudocódigo proporciona dos grandes ventajas, su sintaxis sencilla y la utilización de la lengua materna en la confección del código. Con la programación en lengua materna podemos conseguir que el entorno sea más amigable y asequible, reduciendo la carga cognitiva para el estudiante [1].

Parece lógico que, si no sobrecargamos al estudiante con demasiados elementos al principio, podremos reducir la carga cognitiva y conseguir mayor disponibilidad de su memoria de trabajo, para aprender cosas nuevas.

MultiPseudo, disponible para su descarga en el dominio *multipseudo.org*, es una aplicación multiplataforma desarrollada inicialmente en Java y posteriormente se creó una versión JavaScript para utilizar desde el navegador. Nos permite utilizar programación estructurada en pseudocódigo desde los fundamentos básicos, como variables, condicionales, bucles, hasta los objetos. Dispone de una interfaz sencilla que nos permite programar con relativa facilidad. Los lenguajes utilizados en MultiPseudo tienen una sintaxis simple, muy parecida a la de Python, y además se han añadido capacidades no presentes en este último por considerarse necesarias para el aprendizaje con pseudocódigo, como la declaración e inicialización de variables con asignación de tipos, o el bucle “repetir hasta”, entre otras.

En la actualidad disponemos de otra herramienta con características parecidas, PSeInt¹, que permite la programación de pseudocódigo en español complementado con un editor de diagramas de flujo. Aunque MultiPseudo no dispone de editor de diagramas de

flujo, permite trabajar con diferentes lenguas, traducción y ejecución de pseudocódigos y Python 2.

MultiPseudo

MultiPseudo en su versión 1.4 está desarrollado inicialmente en lenguaje Java, haciendo uso de las librerías Jython² versión 2.5.1. Por tanto, para su correcta ejecución y como requisito necesario, hay que tener instalado en el sistema las especificaciones JRE³ de Java 8. Actualmente MultiPseudo permite programación estructurada en pseudocódigo, desde fundamentos básicos, variables, condicionales, bucles hasta objetos, en los siguientes idiomas: Castellano, Castellano Natural, Catalán, Francés, Gallego, Italiano, Portugués, Rumano y programación en Python 2. Una vista general de cómo es el entorno de programación se puede ver en la Figura 1. Su interfaz general consta de tres partes, un menú, el área de escritura de código fuente y el área de salida.

Posteriormente, se desarrolló una versión online de MultiPseudo con uso de JavaScript y Skulpt⁴, que nos permite usar la aplicación desde el navegador de internet (Véase Figura 2). Ambas versiones están disponibles para uso y descarga desde el dominio antes mencionado.

El objetivo inicial de la herramienta es permitir programar usando pseudocódigo Castellano. Para la confección de este pseudocódigo, se han empleado las especificaciones de palabras clave comúnmente usadas en la literatura actual [4]. También se facilita un pseudocódigo llamado Castellano Natural, donde se han modificado algunas palabras clave, seleccionadas como preferidas por los estudiantes, resultado de un estudio previo [6]. En este estudio se preguntaba a los estudiantes la preferencia sobre algunas de las principales palabras clave de pseudocódigo. En la confección de los demás lenguajes de pseudocódigo se ha utilizado un traductor, la aplicación en línea Reverso⁵ y los diccionarios oficiales en línea de cada lengua, buscando la palabra clave adecuada con un significado computacional acorde a lo pretendido.

Podemos realizar traducción de algoritmos entre los diferentes idiomas de pseudocódigo, y esto facilita la portabilidad de código entre las diferentes lenguas. Así como, en cualquier momento, podemos realizar la traducción del pseudocódigo a Python 2. Se hacen uso de un total de 40 palabras reservadas. Como es sabido, las palabras reservadas no pueden usarse como variables, y los nombres de variables no deben contener caracteres especiales.

² Jython. Implementations of Python in Java. <https://www.jython.org>

³ JRE. Java Runtime Environment. <https://www.java.com>

⁴ Skulpt is an entirely in-browser implementation of Python. <https://skulpt.org>

⁵ Reverso. Free translation, dictionary. <https://www.reverso.net>

¹ Pablo Novara. PSeInt, 2014. <http://pseint.sourceforge.net>

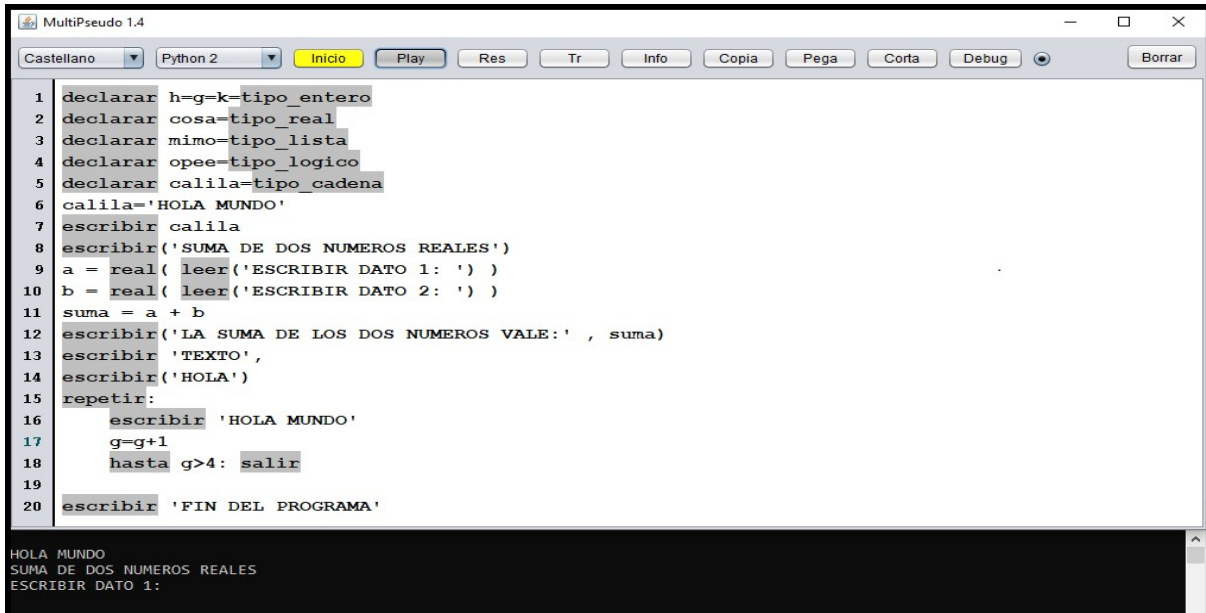


Figura 1. Pantalla principal de MultiPseudo, con un ejemplo de uso.

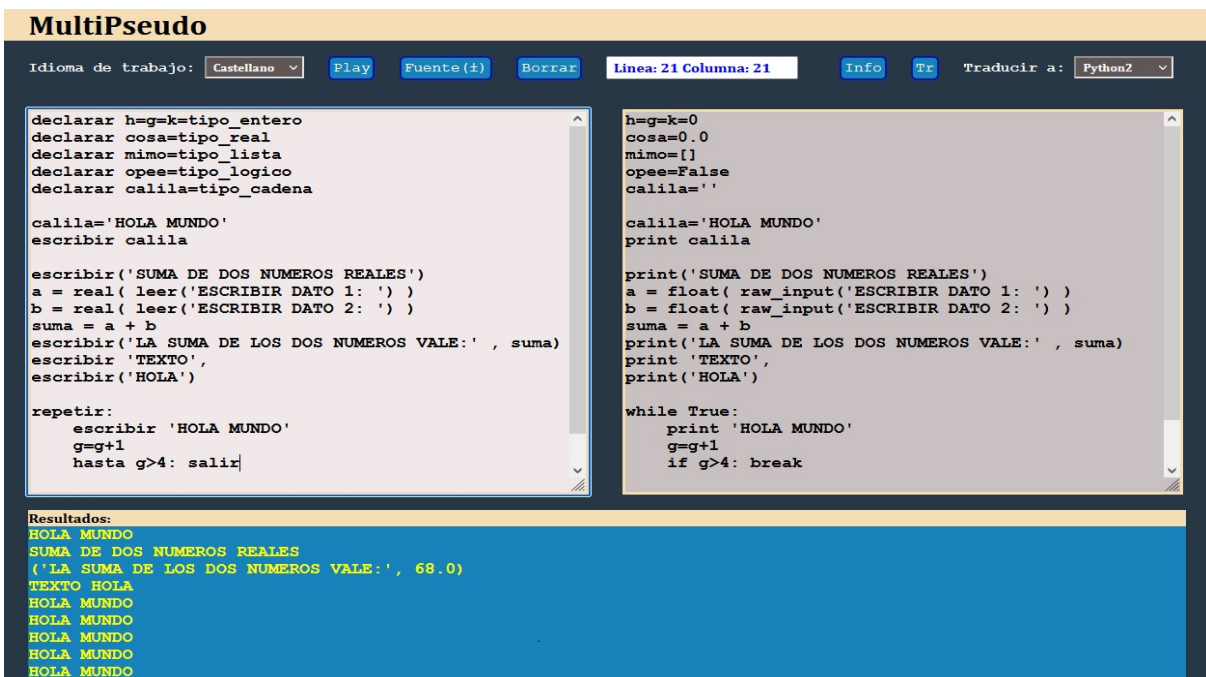


Figura 2. Pantalla principal de MultiPseudo Online, con un ejemplo de uso.

La aplicación, en el botón (Info) proporciona varios ejemplos de uso, donde se utilizan el total del conjunto de palabras reservadas para el lenguaje Castellano. MultiPseudo permite corrección de errores en el código, nos indicará la posición del código donde se encuentra el error, y acceso a un modo depurador (Debug) y poder ejecutar paso a paso las instrucciones. También podremos comprobar la sintaxis del código, con el resaltado de las palabras clave antes de la ejecución. Tenemos la posibilidad de aumentar el tamaño de la fuente de trabajo si el entorno de trabajo así lo requiere.

La versión Java de MultiPseudo tiene implementada una funcionalidad adicional que permite añadir nuevos lenguajes de pseudocódigo. En el paquete de descarga se adjunta también la versión MultiPseudo 1.4n para evaluación de nuevos pseudocódigos, mediante un configurador podremos definir nuevos idiomas de pseudocódigo (N1, N2, N3), comprobar la viabilidad del nuevo conjunto de palabras clave antes de usarlas con la aplicación, y todo esto con relativa facilidad. En unos minutos podremos tener operativo un nuevo Lenguaje de pseudocódigo y trabajar con él. En la Figura 3 vemos una imagen del configurador y

Castellano	CastNatural	Catalán	Francés	Gallego	Italiano	Portugués	Rumano	N	Descripción
tipo_entero	tipo_entero	tipus_enter	type_entier	tipo_entero	tipointero	tipo_inteiro	tip_intreg	tipo_entero	00_TIPO_DE_DATOS_NUMERO_ENTERO
tipo_real	tipo_real	tipus_real	type_reel	tipo_real	tipo_reale	tipo_real	tip_real	tipo_real	01_TIPO_DE_DATOS_NUMERO_REAL
tipo_logico	tipo_logico	tipus_logic	type_logique	tipo_loxico	tipo_logico	tipo_logico	tip_logic	tipo_logico	02_TIPO_DE_DATOS_LOGICO-BOOLEANO
tipo_cadena	tipo_cadena	tipus_cadena	type_chaine	tipo_cadea	tipo_stringa	tipo_cadeia	tip_lant	tipo_cadena	03_TIPO_DE_DATOS_CADENA-ALFANUM...
tipo_lista	tipo_lista	tipus_llista	type_list	tipo_lista	tipo_lista	tipo_lista	tip_lista	tipo_lista	04_TIPO_DE_DATOS_LISTA
global	global	global	global	global	globale	global	global	global	05_GLOBAL
entero	entero	enter	entier	enteiro	intero	inteiro	intreg	entero	06_ENTERO
real	real	real	reel	real	reale	real	real	real	07_REAL
cadena	cadena	cadena	chaine	cadea	stringa	cadeia	lant	cadena	08_CADENA-ALFANUMERICO
verdadero	verdadero	veritable	vrai	verdade	vero	verdadeiro	adevarat	verdadero	09_VERDADERO
falso	falso	fals	faux	falso	falso	falso	fals	falso	10_FALSO
;;	;;	;;	;;	;;	;;	;;	;;	;;	11_OPERADOR_LOGICO_Y(AND)
oo	oo	oo	oo	oo	oo	oo	oo	oo	12_OPERADOR_LOGICO_O(OR)
!!	!!	!!	!!	!!	!!	!!	!!	!!	13_OPERADOR_LOGICO_NO(NOT)
para	iniciando	per	pour	para	per	para	pentru	para	14_BUCLE_PARA-DESDE(FOR)
dentro_de	dentro_de	dins	dans	dentro	dentro	dentro	interior	dentro_de	15_DENTRO_DE-EN_EL
rango	rango	rang	rang	rango	intervallo	faixa	rang	rango	16_RANGO_DE_VALORES-INTERVALLO
mientras	siempre_que	mentre	pendant_que	mentres	mentre	enquanto	in_timp_ce	mientras	17_BUCLE_MIENTRAS(WHILE)
continuar	continuar	continuar	continuer	continuar	continua	continuar	continua	continuar	18_CONTINUAR
salir	salir	sortir	sortir	saír	uscire	saír	iesi	salir	19_SALIR-INTERRUMPIR
subprograma	subprograma	subprograma	sous_programme	subprograma	sottoprogramma	subprograma	subprogram	subprograma	20_SUBPROGRAMA-SUBALGORITMO
retornar	retornar	retornar	retourner	retornar	ritornare	retornar	intoarce	retornar	21_RETORNAR-DEVOLVER
leer	introducir	llegir	lire	ler	leggere	ler	lectura	leer	22_LEER
escribir	mostrar	escriure	ecrire	escribir	scrivere	escrever	scrie	escribir	23_ESCRIBIR
clase	clase	classe	classe	clase	classe	classe	clasa	clase	24_CLASE
longitud	longitud	longitud	longueur	longitude	lunghezza	comprimento	lungime	longitud	25_LONGITUD-TAMÑO
indice	indice	index	indice	indice	indice	indice	indice	indice	26_INDICE
contar	contar	comptar	compter	contar	contare	contar	conta	contar	27_CONTR
agregar	agregar	agregar	agregre	agregar	agregare	agregar	agrega	agregar	28_AGREGAR
retirar	retirar	retirar	retirer	retirar	ritirare	retirar	retrage	retirar	29_RETIRAR-BORRAR
insertar	insertar	insèrir	insèrer	insertar	insèrre	insertar	introduce	insertar	30_INSERTAR
eliminar	eliminar	eliminar	eliminar	eliminar	eliminare	eliminar	elimina	eliminar	31_ELIMINAR-BORRAR
extender	extender	estendre	etendre	estender	estendere	estender	extinde	extender	32_EXTENDER-AÑADIR
invertir	invertir	invertir	inverser	invertir	invertire	invertir	inversa	invertir	33_INVERTIR_ORDEN_ELEMENTOS
sino_si	sino_si	sino_si	autre_cas_si	outro_caso_si	altro_caso_se	outro_caso_se	caz_opus_daca	sino_si	34_EN_OTRO_CASO_SI
sino	sino	sino	autre_cas	outro_caso	altro_caso	outro_caso	caz_opus	sino	35_EN_OTRO_CASO
si	si	si	si	si	se	se	daca	si	36_SI
declarar	declarar	declarar	declerer	declarar	dichiarare	declarar	declara	declarar	37_DECLARAR
repetir	repetir	repetir	repete	repetir	ripetere	repetir	repeta	repetir	38_BUCLE_REPETIR
hasta	hasta	fins	jusqua	ata	fino	ate	pana	hasta	39_HASTA_QUE

Figura 3. Pantalla del configurador de nuevos lenguajes de pseudocódigo.

podemos observar las 40 instrucciones soportadas para las diferentes lenguas.

Conclusiones y trabajos futuros

Este poster presenta un entorno de programación en pseudocódigo que permite utilizar diferentes lenguajes. Actualmente la aplicación se encuentra en uso, se están realizando estudios sobre la capacidad de aprendizaje de los estudiantes usando pseudocódigo en Castellano Natural, frente a pseudocódigo Castellano y Python. La aplicación nos parece adecuada para estas funciones, no obstante, de cara a futuro, añadiremos nuevas funcionalidades, capacidades e idiomas a MultiPseudo.

Referencias

- [1] I-Jung Chen y Chi-Cheng Chang. *Teoría de Carga Cognitiva: Un Estudio Empírico sobre la Ansiedad y el Rendimiento en Tareas de Aprendizaje de Idiomas*. Electronic Journal of Research in Educational Psychology, 7(2), 729-746. 2009 (nº 18). ISSN: 1696-2095.
- [2] Juan Carlos Fonden-Calzadilla, Mavis Lis Stuart-Cárdenas y Lianne Rodríguez-Matos. *La algoritmización: requisito necesario para la solución de problemas con el empleo de un lenguaje de programación*. Luz, vol. 17, núm. 3, 2018.
- [3] Jorge Iván Fuentes-Rosado y Melquizedec Moo-Medina. *Dificultades de aprender a programar*. Revista Educación En Ingeniería, 12 (24), 76-82, 2017, doi:10.26507/rei.v12n24.728.
- [4] Luis Joyanes Aguilar. *Fundamentos de Programación Algoritmos, Estructura de Datos y Objetos*. McGraw-Hill (4ª edición), 2008.
- [5] Theodora Koulouri, Stanislao Lauria y Robert D. Macredie. *Teaching Introductory Programming: a Quantitative Evaluation of Different Approaches*. ACM Transactions on Computing Education, Volume 14, Issue 4, 26, pp 1–28, 2015.
- [6] Antonio López García y Jaime Urquiza-Fuentes. *First approach to the design of a natural pseudocode language: the case of keywords*. 2022 International Symposium on Computers in Education (SIIE), Coimbra, Portugal, 2022, pp. 1-6, doi: 10.1109/SIIE56031.2022.9982351.
- [7] Linda Mannila, Mia Peltomäki y Tapio Salakoski. *What about a simple language? Analyzing the difficulties in learning to program*. Computer Science Education. 16:3, 211-227, 2006.
- [8] Joaquina Palomar Lever, Sandra I. Montes de Oca Mayagoitia, Alma M. Polo Velázquez y Amparo Victorio Estrada. *Factores explicativos del rendimiento académico en hijos de inmigrantes mexicanos en Nueva York*. Psicología Educativa, vol. 22(2), pp. 125-133, 2016, doi: 10.1016/j.pse.2016.03.001.
- [9] Grace Shum, Angeles Conde, Carmen Díaz, Francisco Martínez y Lorenzo Molina. *Lenguaje y rendimiento escolar: Un estudio predictivo*. Comunicación, Lenguaje y Educación, vol. 2(5), pp. 69-79, 1990.