

Comprensión de pseudocódigo básico en estudiantes de ciclos formativos

Antonio López-García¹, Jaime Urquiza-Fuentes¹

a.lopezg.2018@alumnos.urjc.es; jaime.urquiza@urjc.es

¹ LITE - Laboratorio de Tecnologías de la Información en la Educación, Universidad Rey Juan Carlos, Madrid, España.

DOI: 10.17013/risti.51.37-50

Resumen: El aprendizaje de la programación presenta múltiples retos, uno de ellos es el dominio de lenguajes de programación de alto nivel. Entre los diferentes enfoques para solventar este problema nos fijamos en el uso del pseudocódigo como paso previo a los lenguajes de alto nivel. La motivación de este trabajo es investigar en qué medida comprenden los estudiantes, sin nociones de programación, un determinado pseudocódigo. Se comparan tres tipos distintos: pseudocódigo en inglés, pseudocódigo estándar en español (mayoritariamente usado) y un pseudocódigo natural, usando el español y diseñado en función de las preferencias de los estudiantes (pseudocódigo natural). En este estudio preliminar han participado 58 estudiantes y se ha medido su capacidad de comprensión de algoritmos sencillos con construcciones básicas de programación. Los resultados reflejan una ligera ventaja en cuanto a comprensión, aunque no significativa, del pseudocódigo natural, seguido del estándar y terminando con el inglés.

Palabras-clave: Pseudocódigo; lenguaje de alto nivel; lengua materna; educación; programación.

Understanding of basic pseudocode in students of training cycles

Abstract: Learning programming presents multiple challenges, one of which is mastering high-level programming languages. Among the different approaches to solve this problem, we look at the use of pseudocode as a previous step to high-level languages. The motivation of this work is to investigate to what extent students understand, without programming notions, a given pseudocode. Three different types are compared: pseudocode in English, standard pseudocode in Spanish (mostly used) and a natural pseudocode, using Spanish and designed based on student preferences (natural pseudocode). In this preliminary study, 58 students have participated and their ability to understand simple algorithms with basic programming constructs has been measured. The results reflect a slight advantage in terms of comprehension, although not significant, of the natural pseudocode, followed by the standard and ending with English.

Keywords: Pseudocode; high level language; mother tongue; education; programming.

1. Introducción

Una característica fundamental del pseudocódigo es el uso de la lengua materna, cuyo objetivo es disminuir la carga cognitiva del aprendizaje de los conceptos básicos de programación. Sin embargo, gran parte de los pseudocódigos existentes parecen provenir de traducciones literales de palabras reservadas de lenguajes de alto nivel.

El pseudocódigo es un instrumento utilizado en la enseñanza con estudiantes noveles de programación de ordenadores. Es frecuente el uso de un pseudocódigo estándar (Condor & De la Cruz, 2020; Joyanes, 2008), que en su mayor parte, su conjunto de palabras clave es el resultado de la traducción de un lenguaje de alto nivel.

En un estudio previo realizado con los estudiantes objeto del estudio actual (López-García & Urquiza-Fuentes, 2022), resultaron elegidas como preferidas ciertas palabras clave, que daban lugar a un pseudocódigo que podemos llamar natural. Hasta el momento algunas de estas palabras eran palabras poco usadas o difundidas en la utilización de pseudocódigos, y otras palabras eran coincidentes con el pseudocódigo estándar. La lengua materna parece inducir al estudiante palabras clave que no son las utilizadas comúnmente. Palabras clave comúnmente utilizadas como “leer”, “escribir”, “mientras” y “para”, habían sido sustituidas por las palabras “introducir”, “mostrar”, “siempre que” e “iniciando” respectivamente. Otras instrucciones estándar como: “sí”, “sino” y “repetir” resultaron coincidentes.

La enseñanza de la programación es una tarea difícil para los docentes, a veces los estudiantes de programación no tienen suficiente motivación y las clases pueden resultar tediosas, los docentes deben implementar estrategias que favorezcan un ambiente agradable (Beltrán, Sánchez & Rico, 2021). La hipótesis de partida de este trabajo es que se puede facilitar el aprendizaje de la programación usando un pseudocódigo que facilite su comprensión mediante el uso un lenguaje natural utilizando el idioma materno. Así, podría reducirse la carga cognitiva que los estudiantes deben soportar en el momento de aprender programación, lo que finalmente podría redundar en un mejor aprendizaje y mejores resultados académicos (Karthikeyan, 2017).

En este trabajo, nos referiremos a ese pseudocódigo en lenguaje natural que usa el idioma materno como pseudocódigo natural. Podría ser que un pseudocódigo natural fuese ideal para estudiantes noveles de programación, por lo que su diseño debe tener relevancia. Además, algunos autores mantienen que la metodología de diseño orientada al usuario puede ser útil y el idioma puede afectar significativamente la calidad y la seguridad del software, así como al rendimiento y la productividad del programador. Aunque en muchos casos, estas circunstancias no son tenidas en cuenta (Coblentz, Aldrich, Myers & Sunshine, 2018).

El pseudocódigo tiende a ser la descripción de un algoritmo expresado en cualquier lengua existente, y presenta principalmente dos ventajas fundamentales, simplicidad de sintaxis y uso de la lengua materna. De hecho, existen estudios que vinculan la falta de dominio de una lengua con el fracaso escolar (Palomar, Montes de Oca, Polo & Victorio, 2016) o con la ansiedad que crea en los estudiantes (Chen & Chang, 2009). Por ello, la programación en lengua materna puede hacer que el entorno sea más amigable y asequible, reduciendo la carga cognitiva para el estudiante.

En definitiva, en este estudio pretendemos observar de qué manera el impacto del uso de la lengua materna en el pseudocódigo puede afectar a la comprensión por los estudiantes. Y es por esto que evaluamos la comprensión utilizando un pseudocódigo natural comparándolo con un pseudocódigo estándar y un pseudocódigo basado en inglés, el idioma de la gran mayoría de los lenguajes de programación de alto nivel.

2. Descripción del Estudio

2.1. Contexto educativo

El estudio se realiza en el Instituto de Enseñanza Secundaria (IES) Enrique Tierno Galván, de Madrid, España. En este centro se imparte, Enseñanza Secundaria Obligatoria (ESO) y Formación Profesional (FP), en horario de mañana y tarde.

Las titulaciones en las que se enmarca este estudio son dos: Formación Profesional Básica en Informática y Comunicaciones (FPB INF), y Formación Profesional, Ciclo Formativo de Grado Medio en Sistemas Microinformáticos y Redes (SMR).

El Ciclo Profesional Básico en Informática y Comunicaciones¹, tiene una duración de 2000 horas, se realizan prácticas profesionales en empresas y en él los estudiantes aprenderán a realizar operaciones auxiliares sobre montaje de sistemas microinformáticos, operaciones auxiliares sobre sistemas operativos, herramientas ofimáticas, periféricos y redes. Los requisitos de acceso para este ciclo exigen el cumplimiento de tres condiciones: tener entre 15 y 17 años, haber cursado segundo de Enseñanza Secundaria Obligatoria, y ser propuestos para la realización del ciclo por el equipo docente del centro.

El Ciclo Formativo de Grado Medio en Sistemas Microinformáticos y Redes², tiene una duración de 2000 horas, se realizan prácticas profesionales en empresas y en él los estudiantes aprenderán a instalar y configurar software y hardware, montaje de equipos, uso y manejo de herramientas ofimáticas y redes. Los requisitos de acceso para este ciclo son reunir alguna de las siguientes circunstancias: tener el título de Educación Secundaria Obligatoria (ESO), Formación Profesional Básica, Técnico Auxiliar, Bachillerato o acceso mediante prueba de acceso.

2.2. Participantes

El alumnado del centro es muy heterogéneo, además de los estudiantes de nacionalidad española, un alto número de ellos proceden de distintos países, como Colombia, Ecuador, Bolivia, Brasil, República Dominicana, Marruecos o Rumanía.

Los estudiantes involucrados en el estudio, cuya participación fue voluntaria, se encontraban en los cursos primero y segundo de las titulaciones anteriormente

¹ Título Profesional Básico en Informática y Comunicaciones - TodoFP | Ministerio de Educación y Formación Profesional. <https://www.todofp.es/gl/que-estudiar/loe/informatica-comunicaciones/informatica-comunicaciones.html>

² Técnico en Sistemas Microinformáticos y Redes - TodoFP | Ministerio de Educación y Formación Profesional. <https://www.todofp.es/que-estudiar/loe/informatica-comunicaciones/sistemas-microniformaticos-redes.html>

mencionadas: FPB INF y SMR. Generalmente estos estudiantes están motivados por la profesión informática y la mayoría pretenden continuar sus estudios e incluso la realización de un Ciclo Formativo de Grado Superior en Informática. Así, los estudiantes están involucrados con la informática, pero en el ámbito académico, los estudiantes de FPB INF y primer curso de SMR no han tenido contacto previo con la programación de ordenadores. Los cuatro grupos tienen un nivel de inglés medio.

La población inicial de participantes constaba de 91 estudiantes, pero sólo 58 han participado de forma efectiva en el estudio. Sus edades están comprendidas entre 15 y 25 años. Aunque hay algunos estudiantes que no son españoles, llevan ya tiempo suficiente en España para ser considerados como hispanohablantes. Los estudiantes están divididos en 4 grupos, coincidiendo con los dos cursos de ambas titulaciones (véase resumen en la Tabla 1):

- GRUPO 1. Se trata de un grupo de 16 estudiantes (de un total de 22) de primer curso de FPB INF, con edades comprendidas entre 15 y 18 años). Son estudiantes que han sido propuestos para la realización del ciclo y acceden de forma voluntaria, la mayoría sienten atracción por la informática.
- GRUPO 2. Este es un grupo de 13 estudiantes (de un total de 18) de segundo curso de FPB INF, con edades comprendidas entre 17 y 18 años. Son estudiantes que han realizado primero de FPB INF y fueron propuestos para la realización del ciclo, accediendo de forma voluntaria, algunos de ellos pretenden seguir sus estudios en informática realizando un grado medio.
- GRUPO 3. Este grupo está compuesto por 18 estudiantes (de un total de 31) de primer curso de SMR, con edades comprendidas entre 17 y 24 años. Son estudiantes muy motivados por la informática, con una clara vocación profesional con ella, que han accedido con una titulación previa, ESO o FPB, y de forma voluntaria, con vocación de dedicarse profesionalmente a la informática e incluso la realización de un grado superior en informática.
- GRUPO 4. Este grupo está compuesto por 11 estudiantes (de un total de 20) de segundo curso de SMR, con edades comprendidas entre 18 y 25 años. También tienen una fuerte motivación por la informática. El estudio se realiza finalizando el curso escolar, por lo que los estudiantes de este grupo han recibido algunas nociones de programación y uso de scripts.

Es importante resaltar que los estudiantes no reciben nociones de programación para la titulación de FPB INF y primer curso de SMR. Mientras que los estudiantes de segundo curso de SMR tienen un cierto contacto con estructuras típicas de programación debido al aprendizaje de scripts.

Grupo	Titulación	Curso	Nº Estudiantes
Grupo 1	FPB INF	1º	16
Grupo 2	FPB INF	2º	13
Grupo 3	SMR	1º	18
Grupo 4	SMR	2º	11

Tabla 1 – Resumen de características de los grupos

2.3. Variables e instrumento

El objetivo del estudio es valorar la comprensión por el estudiante de una serie de algoritmos básicos similares, utilizando palabras clave en pseudocódigo natural, estándar e inglés. Pretendemos evaluar el grado de comprensión que alcanzan los estudiantes.

Nuestro instrumento son tres encuestas anónimas (A, B, C), una para cada pseudocódigo: natural (véase Anexo A), estándar español (véase Anexo B) e inglés (véase Anexo C). Los cuestionarios tienen las mismas preguntas, pero las respuestas se encuentran en distinto orden. La principal diferencia entre los cuestionarios está en los algoritmos, que pese a tener el mismo significado computacional, en cada encuesta se utilizan palabras clave específicas para cada uno de los tres pseudocódigos.

Cada encuesta cuenta con seis preguntas, sobre algoritmos básicos de pseudocódigo referentes a instrucciones sobre entrada por teclado, salida por pantalla, control del flujo de ejecución con selección y bucles. Cada pregunta presenta tres opciones de respuesta, compuestas por tres algoritmos básicos de pseudocódigo, donde el estudiante tiene que elegir una, por tanto, la variable dependiente sería el nivel de comprensión alcanzado, de un rango de valores de cero hasta seis. Cada pregunta de la encuesta vale un punto.

2.4. Procedimiento

El estudio se plantea como una actividad complementaria y se realiza en horas de tutoría de clase, dentro del horario habitual, por lo que no supone un incremento de carga lectiva para los estudiantes. El profesor tutor de cada curso es el encargado de realizar la encuesta a sus estudiantes, en total contamos con 4 tutores diferentes.

Al comienzo, el profesor tutor de cada grupo da una breve explicación sobre la forma de realización de la encuesta y seguidamente las reparte de manera aleatoria, garantizando un equilibrio en la cantidad de respuestas para cada tipo de encuesta. Cada estudiante realiza solo una encuesta, la que le ha sido asignada referente a pseudocódigo natural, estándar o inglés.

3. Resultados

En este apartado mostramos los resultados de la encuesta tanto desde un punto de vista global como individual de cada grupo. La duración media de la realización de la encuesta en los 4 grupos ha sido de 26 minutos. Siendo la duración mínima 24 minutos y la máxima 27.

Hemos obtenido las puntuaciones medias para cada grupo y para cada encuesta a partir de la nota media individual de cada estudiante (véase Tabla 2). En la Figura 1 podemos observar un gráfico con las puntuaciones medias de los grupos.

A continuación, se describen los resultados para cada grupo y encuesta:

- En el Grupo 1 la puntuación más alta se ha obtenido para la encuesta de comprensión en pseudocódigo natural y estándar (2,8), el pseudocódigo inglés (2) ha quedado en tercer lugar.

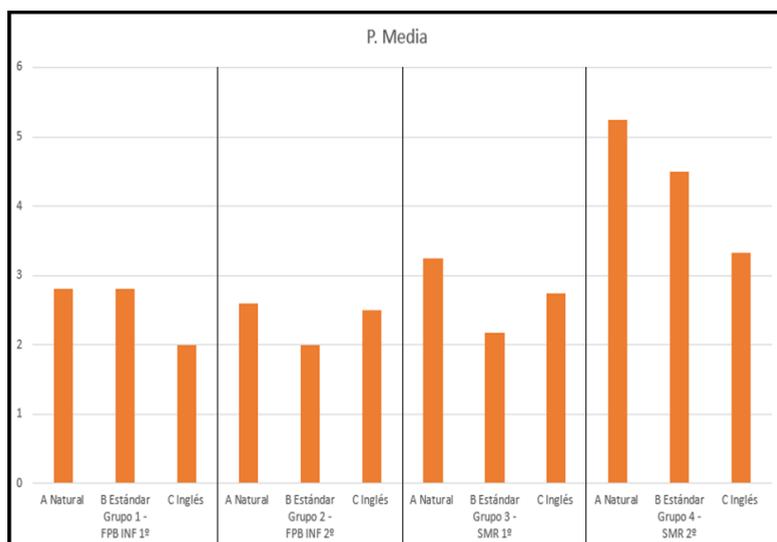


Figura 1 – Puntuaciones medias de los grupos

- En Grupo 2 la encuesta de pseudocódigo natural ocupa el primer lugar (2,6), en segundo lugar, el pseudocódigo inglés (2,5) y en tercer lugar el pseudocódigo castellano (2).
- En el Grupo 3 el pseudocódigo natural vuelve a ocupar el primer lugar (3,25), en segundo lugar, el pseudocódigo inglés (2,75) y en tercer lugar el pseudocódigo castellano (2,17).
- Y terminamos con el Grupo 4, donde el pseudocódigo natural destaca en primer lugar (5,25), seguido del castellano (4,5) y en último lugar el inglés (3,33).

Grupo	Encuesta	Nº Estudiantes	P. Media
Grupo 1 FPB INF 1º	A Natural	5	2,8
	B Estándar	5	2,8
	C Inglés	6	2
Grupo 2 FPB INF 2º	A Natural	5	2,6
	B Estándar	4	2
	C Inglés	4	2,5
Grupo 3 SMR 1º	A Natural	8	3,25
	B Estándar	6	2,17
	C Inglés	4	2,75
Grupo 4 SMR 2º	A Natural	4	5,25
	B Estándar	4	4,5
	C Inglés	3	3,33

Tabla 2 – Puntuaciones medias de los grupos

La Tabla 3 muestra las puntuaciones medias globales para cada encuesta y el total de estudiantes encuestados. La puntuación más alta se ha obtenido para la encuesta de comprensión en pseudocódigo natural (3,36), seguida del pseudocódigo estándar (2,79) y ha quedado en tercer lugar el pseudocódigo inglés (2,53).

Encuesta	Nº Estudiantes	P. Media
A Pseudocódigo Natural	22	3,36
B Pseudocódigo Estándar	19	2,79
C Pseudocódigo Inglés	17	2,53

Tabla 3 – Puntuaciones globales

4. Conclusiones y Limitaciones del Estudio

Este estudio tiene un espíritu preliminar puesto que se realiza en un solo centro educativo, con un determinado contexto social y educativo y con una población en un principio era numerosa ($N = 91$), pero que debido al carácter voluntario del estudio, siendo además final de curso, se vio reducida ($N = 58$). Los resultados se analizan desde un punto de vista cuantitativo (medias y promedios). En general, los resultados, excepto para el Grupo 4, evidencian un nivel de comprensión medio-bajo, en torno a 3 puntos sobre 6 (véase Figura 1). Era previsible que el Grupo 4, al estar integrado por estudiantes que ya han tenido un cierto contacto previo con la programación y uso de scripts, obtenga una mejor puntuación ya que son estudiantes con una mayor comprensión del procedimiento computacional.

Los resultados parecen indicar una mayor facilidad de comprensión para el pseudocódigo natural, ya que ha quedado en primer lugar en todos los grupos, salvo en el Grupo 1. En este caso concreto tiene la misma puntuación que el pseudocódigo estándar, ambos por encima del pseudocódigo en inglés. A nivel global, aunque las diferencias no son estadísticamente significativas, el pseudocódigo natural se sitúa por encima de los otros dos. Estos resultados no son definitivos, ya que repetiremos el estudio a más estudiantes. No obstante, el juego de palabras de pseudocódigo natural parece ayudar y facilitar la comprensión de los estudiantes de programas escritos con ese pseudocódigo. Parece lógico que una reducción en la carga cognitiva para el estudiante, una menor sobrecarga mental puede suponer una mayor disponibilidad del estudiante para aprender nuevos conceptos. Por ello, creemos que es posible que la simplicidad y naturalidad del pseudocódigo pueda favorecer el aprendizaje en estudiantes noveles de programación. Así ayudaría a la comprensión de conceptos iniciales de forma menos costosa permitiendo que los estudiantes dediquen más esfuerzo a aspectos más complejos, como sería el paso a un lenguaje de programación ya de alto nivel.

5. Trabajos Futuros

Estos resultados nos parecen interesantes y nos animan a continuar con el estudio del pseudocódigo centrado en el estudiante, el pseudocódigo natural. Así, creemos que sería conveniente extender al estudio a un mayor número de estudiantes y a otros centros con diferentes contextos. Pretendemos también realizar un estudio posterior

y poder medir la capacidad de aprendizaje de los estudiantes con la gramática natural de pseudocódigo, el tiempo empleado en aprender los fundamentos de programación, y contrastar resultados con algún lenguaje de alto nivel. En un análisis sobre entornos de programación para principiantes (Laura-Ochoa & Bedregal-Alpaca, 2021), a los estudiantes no les resultó fácil aprender a programar con PSeInt. Nosotros utilizaremos la herramienta MultiPseudo (López-García & Urquiza-Fuentes, 2023) para la enseñanza de la programación, que permite cambiar el idioma de las palabras clave del lenguaje fuente.

Anexo A - Cuestionario Tipo A - Pseudocódigo Natural

A -- COMPRENSIÓN DE PSEUDOCÓDIGO -- edad: curso:

Elegir la respuesta correcta, el pseudocódigo correcto, en concordancia con el enunciado del ejercicio.

01. Sabiendo que A y B son dos lugares donde podemos almacenar datos numéricos y CAD un lugar donde podemos almacenar datos alfanuméricos (números y letras).

Nos Aparece por pantalla un mensaje "Escriba un ..." y después entraremos por teclado un valor numérico para A, de la misma forma otro para B, y un valor alfanumérico(cadena) para CAD.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mostrar "Escriba un número:" Mostrar A Mostrar "Escriba un número:" Mostrar B Mostrar "Escriba una cadena:" Mostrar CAD	Mostrar "Escriba un número:" Introducir A Mostrar "Escriba un número:" Introducir B Mostrar "Escriba una cadena:" Introducir CAD	Mostrar "Escriba un número:" Introducir A Mostrar "Escriba un número:" Introducir B Mostrar "Escriba una cadena:" Mostrar CAD

02. Sabiendo que A, B, SUMA son lugares donde podemos almacenar datos numéricos.

Nos aparece un mensaje por pantalla "Ingrese ...". Entramos por teclado el valor de A y de la misma forma para B. En SUMA almacenamos la suma de A+B. Visualizamos en pantalla los resultados.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mostrar "ingrese A:" Introducir A Mostrar "ingrese B:" Introducir B SUMA=(A+B) Mostrar "La suma de A + B es:", SUMA	Mostrar "ingrese A:" Mostrar A Mostrar "ingrese B:" Mostrar B SUMA=(A+B) Mostrar "La suma de A + B es:", SUMA	Mostrar "ingrese A:" Introducir A Mostrar "ingrese B:" Introducir B SUMA=(SUMA+B) Mostrar "La suma de A + B es:", SUMA

03. Sabiendo que EDAD es un lugar donde podemos almacenar datos numéricos.

Vemos un mensaje por pantalla que nos dice "Introduzca edad:". Entramos EDAD por teclado. Visualizar en pantalla si es mayor o menor de edad. Aclaración de simbología, mayor o igual que (\geq) y menor o igual que (\leq).

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mostrar "Introduzca edad:" Introducir E Si EDAD \leq 18 Mostrar "Es mayor" SiNo Mostrar "Es menor de edad"	Mostrar "Introduzca edad:" Introducir EDAD Si EDAD \leq 18 Mostrar "Es mayor" SiNo Mostrar "Es menor de edad"	Mostrar "Introduzca edad:" Introducir EDAD Si EDAD \geq 18 Mostrar "Es mayor" SiNo Mostrar "Es menor de edad"

A -- COMPRENSIÓN DE PSEUDOCÓDIGO -- edad: curso:

04. Tenemos A y B que son dos lugares donde podemos almacenar datos numéricos y CAD un lugar donde podemos almacenar datos alfanuméricos (números y letras).

Nos aparece un mensaje "Introduzca un número:" Entramos por teclado un número entero que guardamos en A. Nos aparece un mensaje "Introduzca una palabra:" Entramos por teclado una palabra o cadena que guardamos en CAD. Visualizar en pantalla la palabra repetida, tantas veces como indique el número. Por ejemplo, si el número es 4 y la palabra es "mía", de la ejecución de código resultaría: mía mía mía mía

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mostrar "Introduzca un número:" Introducir A Mostrar "Introduzca una palabra:" Introducir CAD Iniciando X=1 Hasta A hacer: Mostrar CAD	Mostrar "Introduzca un número:" Mostrar "Introduzca una palabra:" Introducir CAD Iniciando X=1 Hasta A hacer: Mostrar CAD	Mostrar "Introduzca un número:" Introducir A Mostrar "Introduzca una palabra:" Introducir CAD Iniciando X=A Hasta CAD hacer: Mostrar CAD

05. Tenemos P, que es un lugar donde podemos almacenar datos alfanuméricos (números y letras). Inicialmente guardamos un valor en P de cadena vacía (nada). De forma cíclica, primero vemos por pantalla un mensaje que nos dice "Escribe una palabra:" Entramos por teclado una palabra que guardamos en P, el programa la imprime en pantalla. Cuando introducimos la palabra "exit" por teclado, el programa no hace nada y termina la ejecución del mismo. *Aclaración de simbología, desigual (\neq).*

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>P=""</p> Siempre Que P \neq "exit" Hacer: Mostrar "Escribe una palabra:" Introducir P Mostrar P	<p>P=""</p> Siempre Que P="exit" Hacer: Mostrar "Escribe una palabra:" Introducir P Mostrar P	<p>P="exit"</p> Siempre Que P \neq "exit" Hacer: Mostrar "Escribe una palabra:" Introducir P Mostrar P

06. A es un lugar donde podemos almacenar número enteros. De forma cíclica, vemos en pantalla un mensaje que nos dice "Escribe un número:" entramos por teclado un número de tipo entero, el programa imprime en pantalla el valor entrado. Cuando introducimos el valor 9, terminamos la ejecución del mismo.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Repetir Mostrar "Escribe un número:" Introducir A Mostrar A Hasta Que A \neq 9	Repetir Mostrar "Escribe un número:" Mostrar A Hasta Que A \neq 9	Repetir Mostrar "Escribe un número:" Introducir A Mostrar A Hasta Que A=9

Anexo B - Cuestionario Tipo B - Pseudocódigo Estándar

B -- COMPRENSIÓN DE PSEUDOCÓDIGO -- edad: curso:

Elegir la respuesta correcta, el pseudocódigo correcto, en concordancia con el enunciado del ejercicio.

01. Sabiendo que A y B son dos lugares donde podemos almacenar datos numéricos y CAD un lugar donde podemos almacenar datos alfanuméricos (números y letras).

Nos Aparece por pantalla un mensaje "Escriba un ..." y después entraremos por teclado un valor numérico para A, de la misma forma otro para B, y un valor alfanumérico(cadena) para CAD.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Escribir "Escriba un número:" Leer A Escribir "Escriba un número:" Leer B Escribir "Escriba una cadena:" Leer CAD	Escribir "Escriba un número:" Leer A Escribir "Escriba un número:" Leer B Escribir "Escriba una cadena:" Escribir CAD	Escribir "Escriba un número:" Escribir A Escribir "Escriba un número:" Escribir B Escribir "Escriba una cadena:" Escribir CAD

02. Sabiendo que A, B, SUMA son lugares donde podemos almacenar datos numéricos.

Nos aparece un mensaje por pantalla "Ingrese ...". Entramos por teclado el valor de A y de la misma forma para B. En SUMA almacenamos la suma de A+B. Visualizamos en pantalla los resultados.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Escribir "ingrese A:" Escribir A Escribir "ingrese B:" Escribir B SUMA=(A+B) Escribir "La suma de A + B es:", SUMA	Escribir "ingrese A:" Leer A Escribir "ingrese B:" Leer B SUMA=(SUMA+B) Escribir "La suma de A + B es:", SUMA	Escribir "ingrese A:" Leer A Escribir "ingrese B:" Leer B SUMA=(A+B) Escribir "La suma de A + B es:", SUMA

03. Sabiendo que EDAD es un lugar donde podemos almacenar datos numéricos.

Vemos un mensaje por pantalla que nos dice "Introduzca edad:". Entramos EDAD por teclado. Visualizar en pantalla si es mayor o menor de edad. Aclaración de simbología, mayor o igual que (\geq) y menor o igual que (\leq).

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Escribir "Introduzca edad:" Leer EDAD Si EDAD \leq 18 Escribir "Es mayor" SiNo Escribir "Es menor de edad"	Escribir "Introduzca edad:" Leer EDAD Si EDAD \geq 18 Escribir "Es mayor" SiNo Escribir "Es menor de edad"	Escribir "Introduzca edad:" Leer E Si EDAD \leq 18 Escribir "Es mayor" SiNo Escribir "Es menor de edad"

B -- COMPRENSIÓN DE PSEUDOCÓDIGO -- edad: curso:

04. Tenemos A y B que son dos lugares donde podemos almacenar datos numéricos y CAD un lugar donde podemos almacenar datos alfanuméricos (números y letras).

Nos aparece un mensaje "Introduzca un número:" Entramos por teclado un número entero que guardamos en A. Nos aparece un mensaje "Introduzca una palabra:" Entramos por teclado una palabra o cadena que guardamos en CAD. Visualizar en pantalla la palabra repetida, tantas veces como indique el número. Por ejemplo, si el número es 4 y la palabra es "mía", de la ejecución de código resultaría: mía mía mía mía

□	□	□
Escribir "Introduzca un número:" Escribir "Introduzca una palabra:" Leer CAD Para X=1 Hasta A hacer: Escribir CAD	Escribir "Introduzca un número:" Leer A Escribir "Introduzca una palabra:" Leer CAD Para X=A Hasta CAD hacer: Escribir CAD	Escribir "Introduzca un número:" Leer A Escribir "Introduzca una palabra:" Leer CAD Para X=1 Hasta A hacer: Escribir CAD

05. Tenemos P, que es un lugar donde podemos almacenar datos alfanuméricos (números y letras). Inicialmente guardamos un valor en P de cadena vacía (nada). De forma cíclica, primero vemos por pantalla un mensaje que nos dice "Escribe una palabra:" Entramos por teclado una palabra que guardamos en P, el programa la imprime en pantalla. Cuando introducimos la palabra "exit" por teclado, el programa no hace nada y termina la ejecución del mismo. Aclaración de simbología, desigual (≠).

□	□	□
P="" Mientras P="exit" Hacer: Escribir "Escribe una palabra:" Leer P Escribir P	P="exit" Mientras P ≠ "exit" Hacer: Escribir "Escribe una palabra:" Leer P Escribir P	P="" Mientras P ≠ "exit" Hacer: Escribir "Escribe una palabra:" Leer P Escribir P

06. A es un lugar donde podemos almacenar número enteros. De forma cíclica, vemos en pantalla un mensaje que nos dice "Escribe un número:" entramos por teclado un número de tipo entero, el programa imprime en pantalla el valor entrado. Cuando introducimos el valor 9, terminamos la ejecución del mismo.

□	□	□
Repetir Escribir "Escribe un número:" Escribir A Hasta Que A ≠ 9	Repetir Escribir "Escribe un número:" Leer A Escribir A Hasta Que A=9	Repetir Escribir "Escribe un número:" Leer A Escribir A Hasta Que A ≠ 9

Anexo C - Cuestionario Tipo C - Pseudocódigo Inglés

C -- COMPRENSIÓN DE PSEUDOCÓDIGO -- edad: curso:

Elegir la respuesta correcta, el pseudocódigo correcto, en concordancia con el enunciado del ejercicio.

01. Sabiendo que A y B son dos lugares donde podemos almacenar datos numéricos y CAD un lugar donde podemos almacenar datos alfanuméricos (números y letras).

Nos Aparece por pantalla un mensaje "Escriba un ..." y después entraremos por teclado un valor numérico para A, de la misma forma otro para B, y un valor alfanumérico(cadena) para CAD.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Write "Escriba un número:" Read A Write "Escriba un número:" Read B Write "Escriba una Cadena:" Write CAD	Write "Escriba un número:" Write A Write "Escriba un número:" Write B Write "Escriba una Cadena:" Write CAD	Write "Escriba un número:" Read A Write "Escriba un número:" Read B Write "Escriba una Cadena:" Read CAD

02. Sabiendo que A, B, SUMA son lugares donde podemos almacenar datos numéricos.

Nos aparece un mensaje por pantalla "Ingrese ...". Entramos por teclado el valor de A y de la misma forma para B. En SUMA almacenamos la suma de A+B. Visualizamos en pantalla los resultados.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Write "ingrese A:" Read A Write "ingrese B:" Read B SUMA=(SUMA+B) Write "La suma de A + B es:", SUMA	Write "ingrese A:" Read A Write "ingrese B:" Read B SUMA=(A+B) Write "La suma de A + B es:", SUMA	Write "ingrese A:" Write A Write "ingrese B:" Write B SUMA=(A+B) Write "La suma de A + B es:", SUMA

03. Sabiendo que EDAD es un lugar donde podemos almacenar datos numéricos.

Vemos un mensaje por pantalla que nos dice "Introduzca edad:". Entramos EDAD por teclado. Visualizar en pantalla si es mayor o menor de edad. Aclaración de simbología, mayor o igual que (\geq) y menor o igual que (\leq).

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Write "Introduzca edad:" Read EDAD If EDAD \geq 18 Write "Es mayor" Else Write "Es menor de edad"	Write "Introduzca edad:" Read E If EDAD \leq 18 Write "Es mayor" Else Write "Es menor de edad"	Write "Introduzca edad:" Read EDAD If EDAD \leq 18 Write "Es mayor" Else Write "Es menor de edad"

C -- COMPRENSIÓN DE PSEUDOCÓDIGO -- edad: curso:

04. Tenemos A y B que son dos lugares donde podemos almacenar datos numéricos y CAD un lugar donde podemos almacenar datos alfanuméricos (números y letras).

Nos aparece un mensaje "Introduzca un número:" Entramos por teclado un número entero que guardamos en A. Nos aparece un mensaje "Introduzca una palabra:" Entramos por teclado una palabra o cadena que guardamos en CAD. Visualizar en pantalla la palabra repetida, tantas veces como indique el número. Por ejemplo, si el número es 4 y la palabra es "mía", de la ejecución de código resultaría: mía mía mía mía

□	□	□
Write "Introduzca un número:" Read A Write "Introduzca una palabra:" Read CAD For X=A To CAD Do: Write CAD	Write "Introduzca un número:" Read A Write "Introduzca una palabra:" Read CAD For X=1 To A Do: Write CAD	Write "Introduzca un número:" Write "Introduzca una palabra:" Read CAD For X=1 To A Do: Write CAD

05. Tenemos P, que es un lugar donde podemos almacenar datos alfanuméricos (números y letras). Inicialmente guardamos un valor en P de cadena vacía (nada). De forma cíclica, primero vemos por pantalla un mensaje que nos dice "Escribe una palabra:" Entramos por teclado una palabra que guardamos en P, el programa la imprime en pantalla. Cuando introducimos la palabra "exit" por teclado, el programa no hace nada y termina la ejecución del mismo. *Aclaración de simbología, desigual (≠).*

□	□	□
P="exit" While P ≠ "exit" Do: Write "Escribe una palabra:" Read P Write P	P="" While P ≠ "exit" Do: Write "Escribe una palabra:" Read P Write P	P="" While P="exit" Do: Write "Escribe una palabra:" Read P Write P

06. A es un lugar donde podemos almacenar número enteros. De forma cíclica, vemos en pantalla un mensaje que nos dice "Escribe un número:" entramos por teclado un número de tipo entero, el programa imprime en pantalla el valor entrado. Cuando introducimos el valor 9, terminamos la ejecución del mismo.

□	□	□
Repeat Write "Escribe un número:" Read A Write A Until A=9	Repeat Write "Escribe un número:" Read A Write A Until A ≠ 9	Repeat Write "Escribe un número:" Write A Until A ≠ 9

Agradecimientos

Agradecemos la colaboración con la ejecución de este trabajo a la Dirección del centro, a la Jefatura del Departamento de Informática, a los profesores participantes en la encuesta de los cursos 2º FPB INF, 1º SMR Y 2º SMR, y a todo el alumnado implicado. Este trabajo ha sido cofinanciado por la Universidad Rey Juan Carlos, con el proyecto puente M2614 y la Comunidad de Madrid, con el proyecto e-Madrid-CM (P2018/TCS-4307), también cofinanciado por los fondos estructurales (FSE y FEDER).

Referencias

- Beltrán, J., Sánchez, H. & Rico, M. (2021). Aprendizaje divertido de programación con Gamificación. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 41(3), 17-33. 2021. <https://doi.org/10.17013/RISTI.41.17-33>
- Chen, I. & Chang, C. (2009). 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. <https://doi.org/10.25115/ejrep.v7i18.1369>
- Coblenz, M., Aldrich, J., Myers, B.A., & Sunshine, J. (2018). Interdisciplinary Programming Language Design. *The Art, Science and Engineering of Programming*, 3. <https://doi.org/10.22152/programming-journal.org/2019/3/16>
- Condor, E.E. & De la Cruz, M.A. (2020). Algoritmos Resueltos con Python. *Editorial EIDEC*. <https://doi.org/10.34893/6kbn-5a63>
- Joyanes, L. (2008). *Fundamentos de Programación Algoritmos, Estructura de Datos y Objetos*. 4ª Ed. McGraw-Hill.
- Karthikeyan, C. (2017). Evaluative Study on Consequences of Cognitive and Sensory Overload: A Socio - Psychological Perspective. *International Journal of Management, IT & Engineering*, 7(4):300-326.
- Laura-Ochoa, L., & Bedregal-Alpaca, N. (2021). Análisis de entornos de programación para el desarrollo de habilidades del pensamiento computacional y enseñanza de programación a principiantes. *RISTI - Revista Ibérica de Sistemas y Tecnologías de la Información*, e43(7), 533-548.
- López-García, A., & Urquiza-Fuentes, J. (2022). First approach to the design of a natural pseudo-code language: the case of keywords. *International Symposium on Computers in Education (SIIE), Coimbra, Portugal*. <https://doi.org/10.1109/SIIE56031.2022.9982351>
- López-García, A., & Urquiza-Fuentes, J. (2023). MultiPseudo, lenguaje de programación en pseudocódigo multi-idioma. *Actas de las Jenui* (8), 391-394.
- Palomar, J., Montes de Oca, S.I., Polo, A. M., & Victorio, A. (2016). Factores explicativos del rendimiento académico en hijos de inmigrantes mexicanos en Nueva York. *Psicología Educativa*, 22(2), 125-133.