

Antonio López García

**Experiencias de Uso del
Pseudocódigo y Java en la
Enseñanza de Programación en
Ciclos Formativos y Bachillerato**

Número 2021-02

Serie de Informes Técnicos DLSI1-URJC

ISSN 1988-8074

**Departamento de Lenguajes y Sistemas Informáticos I
Universidad Rey Juan Carlos**

Índice

1. Introducción	5
2. Trabajos relacionados	6
3. Pseudocódigo y carga cognitiva.....	7
4. La lengua materna	9
5. Motivación y resultados	10
6. Descripción del estudio	10
6.1. Sujetos de estudio	11
6.2. Tratamientos	12
6.3. Variables dependientes	12
6.4. Procedimiento	13
7. Resultados.....	13
8. Conclusiones.....	18
9. Referencias.....	18

Experiencias de Uso del Pseudocódigo y Java en la Enseñanza de Programación en Ciclos Formativos y Bachillerato

Antonio López García¹

¹LITE – Laboratorio de Tecnologías de la Información en la Educación
Universidad Rey Juan Carlos, Madrid, España
a.lopezg.2018@alumnos.urjc.es

Abstract. Con este trabajo, se intenta ayudar y contribuir, a que el Pseudocódigo mantenga un lugar especial en la educación, y se utilice aún más de lo que se ha hecho hasta ahora, no solo con los estudiantes iniciales de informática en universidades, sino también con los estudiantes de Bachillerato y Formación Profesional. Se pretende exponer la aparente eficacia y beneficios de usar el Pseudocódigo en los inicios de la enseñanza en programación de ordenadores, frente a iniciar directamente en un lenguaje de alto nivel como Java. Se plantea como objetivo de trabajo, un estudio preliminar del impacto del uso del Pseudocódigo. Aunque el Pseudocódigo se usa frecuentemente en la enseñanza universitaria, parece que su uso es menos frecuente en Bachillerato y Formación Profesional. Las posibles ventajas del uso del Pseudocódigo, pueden ser, su sintaxis sencilla y el uso de la lengua materna, y consecuentemente puede dar lugar a una menor carga cognitiva. Este estudio se ha realizado con cuatro grupos, un total de 36 estudiantes. Tres grupos aprendieron Java directamente, necesitando una media de 35 horas para alcanzar un nivel aceptable sobre fundamentos de programación. Un grupo aprendió con Pseudocódigo, alcanzando en 20 horas un nivel aceptable sobre fundamentos de programación. Creemos, que esto indica, que el Pseudocódigo, podría ayudar, a un aprendizaje más eficiente de los fundamentos de programación, en los niveles de Bachillerato y Formación Profesional. Queda por analizar el impacto del Pseudocódigo en la transición a un lenguaje de programación real.

Keywords: Pseudocódigo, lenguaje de alto nivel, educación, carga cognitiva, lengua materna, informática.

1 Introducción

Podemos utilizar el Pseudocódigo de forma sistemática, sobre todo al inicio de la enseñanza en lenguajes de programación. El hecho de que a un alumno, un concepto

abstracto se le presente de una forma más fácil, puede significar una mejor asimilación de conceptos, una mayor motivación, y consecuentemente, una mejora en los resultados.

Podría decirse, que el Pseudocódigo es una descripción de un algoritmo, que se puede expresar en cualquier lengua existente [1]. Podría considerarse, un lenguaje neutral que nos proporciona un puente a múltiples lenguajes [2]. La programación de ordenadores es un ámbito amplio y abstracto, que presentado de una forma más simple, a través del Pseudocódigo, parece recomendable al principio.

El Pseudocódigo nos puede proporcionar, principalmente dos ventajas fundamentales. Por un lado, la programación en lengua materna hace que el entorno sea más amigable y asequible, reduciendo la carga cognitiva para el estudiante [3]. Por otro lado, la simplicidad de sintaxis puede facilitar la comprensión. Y tal vez podremos así, alcanzar más rápidamente la asimilación de los conceptos pretendidos. Otro aspecto muy importante del Pseudocódigo, es la posibilidad de portabilidad de código hacia cualquier lenguaje de programación.

Seguidamente, en la sección 2 hacemos referencia a trabajos relacionados, en la sección 3 mostramos la relación entre Pseudocódigo y carga cognitiva, en la sección 4 estudiamos la lengua materna, en la sección 5 la motivación, en la sección 6 presentamos el caso de estudio de investigación, en la sección 7 mostramos los resultados, en la sección 8 conclusiones y en la sección 9 referencias bibliográficas.

2 Trabajos Relacionados

En esta sección analizamos, una serie de trabajos relacionados, que conviene destacar.

Es frecuente encontrar estudios, que pretenden desvelar que lenguaje es más apropiado emplear con los estudiantes iniciales de programación de ordenadores. En el artículo publicado en 2014 por T. Koulouri, S. Lauria y R. D. Macredie [4], se evalúan diferentes enfoques en la enseñanza de la introducción a la programación. Se investigaron los efectos de tres factores: la elección del lenguaje de programación, la capacitación en resolución de problemas y el uso de la evaluación formativa en el aprendizaje de la programación. El estudio se realizó durante cuatro años, utilizando métricas objetivas y cuantitativas. Concluyendo que el uso de un lenguaje sintácticamente simple como (Python), en lugar de uno más complejo como (Java), facilitó el aprendizaje de los conceptos de programación a los estudiantes iniciales. En este estudio, sus autores parten de la idea, de que enseñar programación a principiantes es una tarea compleja [5]. Y esto se ve reflejado en los datos estadísticos de las últimas décadas. La industria demanda y está necesitada en gran medida de profesionales informáticos. Los alumnos que se inician en la informática, en mayor medida los alumnos de primer y segundo curso de Informática, abandonan en un alto porcentaje los estudios. En otras palabras, sufren un desencanto y un elevado porcentaje no consigue graduarse. Se pone de manifiesto que introducir a principiantes en lenguajes de programación como Python está ganando terreno frente a otros lenguajes con una sintaxis y gramática más compleja como Java y C++ [6]. Python ofrece potencia, claridad de sintaxis, y un entorno de desarrollo simple, por lo

que puede parecer más apropiado para iniciarse en la programación, frente a las otras opciones.

En 2006. L. Mannila , M. Peltomäki y T. Salakoski [7], presentan los resultados de un estudio que consta de dos partes. En primer lugar se analizan 60 programas escritos por programadores iniciales de entre 16 y 19 años después de su primer curso de programación, en Java y Python. El objetivo es encontrar dificultades independientes del lenguaje utilizado y las que se originen por el propio lenguaje. Los alumnos que aprenden Java, encuentran más problemas, en general, de sintaxis y de lógica. El lenguaje Python, se promociona como un lenguaje más simple, por lo que parece lógico que los alumnos tengan menos problemas. En segundo lugar, se estudia la transición de un lenguaje "simple" a uno más "avanzado", siguiendo a ocho estudiantes, que aprendieron programación en Python antes de pasar a Java. Los resultados sugieren que un lenguaje simple da lugar a menos errores de sintaxis y menos errores de lógica. Podría pensarse que aprender un lenguaje simple, podría ser algo inútil, y luego se tendría que aprender algo nuevo, pero no se reveló ninguna desventaja de haber aprendido a programar en un lenguaje sencillo y pasar a uno más complejo. Esto sugiere que se puede utilizar un lenguaje simple al introducir la programación como una habilidad general, y a la vez proporcionar así habilidades básicas, fundamentos básicos en programación a futuros profesionales, el Pseudocódigo puede ser una buena opción.

Resulta de interés, el artículo de Orlando Arboleda [8], publicado en 2011. Presenta un nuevo lenguaje de programación con estructuras similares al Pseudocódigo, se describe su compilador y posterior traductor a programas equivalentes en Java. También hace alusión a herramientas para apoyar la enseñanza y programar en Pseudocódigo, permitiendo visualizar los resultados, tales como PSeInt [9].

F. Huari Evangelista y P. José Novara [10], fomentan como herramienta para dar a conocer a la comunidad estudiantil que se inicia en la programación, la aplicación PSeInt, y de esta forma interpretar un algoritmo escrito en Pseudocódigo. Se pone a disposición de los estudiantes, la posibilidad de entender cómo trabaja un programa, operar con algoritmos y visualizar datos reales sin necesidad de utilizar un lenguaje de alto nivel. Y el estudiante, no necesita conocer ningún lenguaje de programación.

3 Pseudocódigo y carga cognitiva

La capacidad de nuestro cerebro, aunque muy extensa, es limitada, la carga cognitiva hace referencia a la cantidad de cuestiones que este puede procesar simultáneamente, o son necesarias para producir un trabajo. Es muy importante para los profesores, conocer esta teoría, y realizar unas actuaciones, en consonancia con esta, en su trabajo.

La teoría de la carga cognitiva, fue desarrollada en los años 80 y 90, por el psicólogo-investigador educativo australiano John Sweller [11], y según él, tenemos una memoria de trabajo y una memoria a largo plazo. La memoria de trabajo, de capacidad limitada a 7 ± 2 elementos simultáneos [12], o aún más limitada como concluyen otros autores [13], incluye subcomponentes parcialmente independientes,

para tratar con material auditivo/verbal e información visual/o tridimensional. La capacidad del cerebro para procesar información nueva, simultáneamente es limitada. Si queremos que se produzca un aprendizaje de forma eficiente, cuyos contenidos se almacenen de forma correcta, no podemos sobrecargar la memoria de trabajo. Si la sobrecargamos entraríamos en una dinámica negativa de aprendizaje, estaríamos perdiendo el tiempo. La memoria de trabajo, es una memoria volátil, asimilado a lo que pensamos en el momento, donde pequeñas cantidades de información se almacenan durante un pequeño periodo de tiempo.

El Pseudocódigo, nos permite iniciarnos en los fundamentos de la programación de ordenadores, y es posible, que al estar escrito en lengua materna, y al componerse de una sintaxis sencilla, permita, reducir la carga cognitiva para el estudiante. El estudiante podría asimilar más fácilmente los conceptos, y en menor tiempo, y esto podría implicar, que el alumno esté más comprometido con el aprendizaje. El alumno podría aumentar su motivación al obtener logros, y cerciorarse de que puede avanzar, y como consecuencia obtener un mayor rendimiento académico [14]. La abstracción y complejidad de un lenguaje de alto nivel, al estar escrito en otra lengua y por su sintaxis compleja, puede suponer una barrera al principio. Parece lógico, que si no sobrecargamos al estudiante, con demasiados elementos al principio, podemos conseguir la disponibilidad de su cerebro, de su memoria de trabajo, para aprender cosas nuevas.

Según Sweller, en cuanto a la memoria a largo plazo, tiene una capacidad ilimitada [11], aquí grandes cantidades de información se almacenan de forma permanente o semipermanente, en forma de esquemas o estructuras. Estos contenidos de la memoria a largo plazo, nos permiten percibir, analizar, pensar y tratar situaciones. A medida que asimilamos esquemas, conseguimos una automatización del proceso, de manera, que los esquemas anteriores nos sirven de base, para construir conocimiento. Puede ser que el Pseudocódigo, nos facilite la construcción de conocimiento a los estudiantes iniciales.

La carga cognitiva, hace referencia a la cantidad de actividad de la memoria de trabajo en un determinado instante [15]. Diversos autores consideran, que la carga cognitiva total es la suma de 3 tipos diferentes de carga [16], la intrínseca, extrínseca y pertinente (o relevante). La intrínseca, es una carga necesaria, que guarda relación con la complejidad o dificultad, del contenido de lo que se pretende enseñar. Por lo que, para un estudiante con conocimientos previos en la materia, puede resultarle menor. La carga cognitiva extrínseca, guarda relación con el contenido en sí, que se pretende enseñar, por lo que parece recomendable, no maximizar esta carga, para no saturar la memoria de trabajo. La carga cognitiva pertinente, es la inherente al propio proceso de aprendizaje, de construcción de esquemas, y transferencia de información a la memoria a largo plazo, por lo que es relevante para el aprendizaje.

Para que tenga lugar el aprendizaje de conceptos, los contenidos informativos, deben permanecer en la memoria de trabajo, hasta que se haya procesado el tiempo suficiente para pasar a la memoria a largo plazo. Es en la memoria a largo plazo, donde se guardan los esquemas, los cuales, quedarán disponibles, almacenados, y llegaríamos a una automatización de estos esquemas, que estarían en disposición de usar. Durante toda la vida estaremos adquiriendo esquemas y aprendiendo. A medida que estamos almacenado esquemas en la memoria a largo plazo, de una manera

adecuada, estaríamos, aliviando los procesos de información en la memoria de trabajo.

Frecuentemente en educación se usa, la división de problemas en partes, para facilitar el aprendizaje. Es decir, resolver un problema difícil, dividiéndolo en partes más simples, tantas veces como sea necesario, hasta que la solución sea clara, y la solución del problema principal se construye con las soluciones encontradas, tratadas individualmente. En informática, esto es usado de forma metódica. Con la programación estructurada, un programa complejo, siempre lo dividimos en subprogramas más pequeños, lo que nos puede proporcionar, claridad y efectividad. En educación, con el Pseudocódigo, cuando partimos de conceptos simples, o más entendibles, o ya asimilados, estamos reduciendo la carga cognitiva del alumnado, y favoreciendo un aprendizaje eficaz, es decir, partimos de lo simple hacia lo complejo [17].

En el ámbito de la informática educativa, la simplicidad y sencillez del Pseudocódigo, permite al individuo una mayor disponibilidad de su memoria de trabajo para el aprendizaje de esquemas, destinar más recursos a esto y desarrollar de una forma consistente la memoria a largo plazo.

4 La lengua materna

El Pseudocódigo proporciona, un entorno más amigable y ameno para el alumnado, un ambiente más simple y asequible para la asimilación de conceptos [17], y una de estas razones, muy importante, es la ventaja, de que lo desarrollamos en lengua materna [3]. La lengua materna guarda una relación directa con la capacidad que tiene el alumno de prosperar en sus estudios.

En 1990 se realiza un estudio [18], en el que sus autores, establecen una relación entre el fracaso escolar y el lenguaje. Realizado en 13 colegios públicos de EGB (educación primaria), los autores concluyen que el lenguaje está íntimamente relacionado con el rendimiento escolar y que el lenguaje natural es determinante en el rendimiento escolar, por lo que sería posible elaborar un programa de intervención, para evitar el fracaso escolar.

En un estudio realizado en 2014 en Nueva York [19], se presenta una investigación en la que se identifican los factores que influyen en el rendimiento académico de los hijos de inmigrantes mexicanos que residen en Nueva York. Sus autores concluyen, que los alumnos con mayor conocimiento del idioma Inglés obtienen un mejor rendimiento académico. Se aprecia claramente, como el factor idioma influye directamente en el rendimiento escolar.

En 2017, un estudio ambientado en Ceuta [20], expone que la lengua materna influye más que el género, en el rendimiento académico del alumnado, y los alumnos que tienen una lengua materna diferente del castellano tienen un rendimiento más bajo.

En un estudio presentado en 2009, sobre rendimiento en aprendizaje de idiomas y carga cognitiva [3], sus autores concluyen que el grado de ansiedad hacia un idioma

extranjero, incurre en una mayor carga cognitiva, y da lugar a peores resultados académicos.

Ponemos así de manifiesto, la importancia que la lengua materna tiene en la educación, y la relación que guarda con los resultados académicos de los estudiantes. Los estudiantes prosperan y obtienen mayores logros, utilizando la lengua materna, con menos sobrecargas.

5 Motivación y resultados

Gracias a la simplicidad y flexibilidad del Pseudocódigo, podemos conseguir en clase un ambiente más amigable y motivador. Los alumnos pueden comprender los algoritmos más rápidamente, entender los procedimientos, obtener logros, y claro, pueden estar más comprometidos y motivados [14]. Por todo esto, la clase sería más productiva. Pretendemos mostrar en este apartado, la importancia de la motivación en el alumnado y la consecuente mejora en resultados, para lo que analizaré, algunos estudios, que me han parecido representativos.

Un estudio sobre motivación para mejorar el rendimiento de los alumnos en la materia de matemáticas [21], revela y manifiesta la importancia de la motivación de los alumnos y su repercusión en resultados. El autor concluye, que poniendo los alumnos a reflexionar y hacer las cosas por convicción, no por obligación, conlleva a éxito.

En 2017, M.L. Navarro [14], presenta un trabajo en el que pone en práctica una serie de estrategias motivadoras, obteniendo buenos resultados, los alumnos han experimentado una mejora significativa en el aprendizaje y resultados. La autora, subraya la importancia de la motivación dentro del aula con una consecuencia inmediata de obtener una mejora en los resultados.

6 Descripción del estudio

En esta sección se describen los sujetos, la metodología, y las variables que luego se medirán en los 4 grupos estudiados.

El estudio, se realiza a lo largo de 3 cursos académicos, en diferentes trimestres, con alumnos de enseñanzas medias de institutos españoles, de Formación Profesional y Bachillerato, en diferentes cursos y niveles educativos.

Con 3 grupos, iniciamos, el estudio en lenguaje de programación de alto nivel Java, y con Pseudocódigo en el cuarto grupo.

Tendremos en cuenta los elementos integradores de cada grupo, como son motivaciones parecidas, grupos de edades similares, capacidades académicas en general similares, y especialmente el hecho de que ningún grupo ha tenido previamente contacto con la programación de ordenadores. Podemos apreciar en algunos grupos, algunos elementos diferenciadores, como diferentes cursos, variado número de alumnos, diferentes circunstancias personales. El estudio se ha realizado de forma secuencial, a continuación mostramos una descripción de los sujetos de cada grupo.

6.1 Sujetos de estudio

El estudio se realiza en 4 institutos diferentes. En dos grupos, con alumnos de 2º curso de Bachillerato, un grupo con alumnos de 1º de Formación Profesional, FP, Grado Superior en Informática, y un cuarto grupo con alumnos de 1º de Formación Profesional, FP, Grado Medio en Informática.

GRUPO 1. Se trata de un grupo de 9 alumnos, de 2º curso de Bachillerato, con edades comprendidas entre 17 y 19 años. La asignatura es “TIC2”, Tecnología de la Información y Las Comunicaciones 2, es elegida de forma optativa. Estos alumnos han elegido la asignatura de forma voluntaria, lo que puede implicar una mayor motivación y compromiso con esta.

No habían tenido ningún contacto con la programación de ordenadores, y este 2º trimestre está destinado al lenguaje de programación Java.

Son alumnos de Bachillerato, responsables y trabajadores, no se pierde tiempo en clase, y predomina el bienestar y buen entendimiento.

GRUPO 2. Se trata de un grupo de 24 alumnos de 1er curso de Formación Profesional, Grado Superior en Informática, con edades comprendidas entre 18 y 29 años. La asignatura es “Programación de Ordenadores”, impartida en lenguaje Java. El estudio se realiza en el 1er trimestre, al inicio del curso escolar.

Existe un alto grado de implicación entre el alumnado, casi todos tienen intención de enfocar su futura vida profesional hacia la Informática. No han tenido contacto con la programación de ordenadores, aunque hay algún alumno que ha adquirido conocimientos de forma autodidacta.

Todos mantiene un alto interés y atención a las clases, muy buen ambiente en clase, los alumnos están altamente motivados. Todos participan activamente en el transcurso y desarrollo de las clases.

GRUPO 3. Este grupo está compuesto por 6 alumnos, de 17 y 18 años, la asignatura que se imparte es “TIC2”, Tecnología de la Información y las Comunicaciones 2, asignatura que han elegido de forma optativa.

Son alumnos de 2º curso de Bachillerato, trabajadores, no se pierde tiempo en clase, y predomina el bienestar y buen entendimiento. Hay tres alumnos que manifiestan su intención de proseguir estudios en informática.

No habían tenido contacto con la programación de ordenadores, y este 2º trimestre del curso, está destinado al lenguaje de programación Java.

GRUPO 4. Se trata de un grupo de 7 alumnos de 1er curso de Formación Profesional, Grado Medio en Informática, con edades comprendidas entre 17 y 21 años. El estudio se realiza en el 3er trimestre del curso.

Existe un alto grado de motivación entre el alumnado, la mayor parte del alumnado tiene intención de enfocar su futura vida profesional hacia la Informática. Previamente, no habían tenido contacto con la programación de ordenadores. Todos mantienen un alto interés y atención a las clases, muy buen ambiente en clase, los alumnos están altamente motivados. Todos participan activamente en el transcurso y desarrollo de las clases.

A continuación se explican los tratamientos.

6.2 Tratamientos

Se realiza un seguimiento personalizado de cada alumno, evaluando continuamente sus progresos y aprendizajes.

- *Aprendizaje de Java*

Con respecto a los alumnos que se le ha enseñado directamente la programación estructurada en Java (bloque de fundamentos de programación: conceptos básicos, sintaxis de Java, tipos de datos y operadores, estructuras básicas de control de flujo, condicionales y bucles), se ha seguido el mismo método para todos los grupos.

Sobre cada concepto estudiado se realizan ejercicios de ejemplo y ejercicios a desarrollar por el alumno.

Después de explicar cada concepto, se exponen ejemplos de utilización, y posteriormente los alumnos realizarán algunos ejercicios de ejemplo. Utilizamos un compilador de JAVA en línea, para no tener necesidad de instalar ningún software.

- *Aprendizaje de Pseudocódigo*

Tenemos un grupo de 7 alumnos, a los que se les ha enseñado directamente la programación estructurada en Pseudocódigo (bloque de fundamentos de programación: conceptos básicos, sintaxis de Pseudocódigo, tipos de datos y operadores, estructuras básicas de control de flujo, condicionales y bucles).

Sobre cada concepto estudiado se realizan ejercicios de ejemplo y ejercicios a desarrollar por el alumno.

Tras explicar cada concepto, se exponen ejemplos de utilización, y posteriormente los alumnos realizarán ejercicios de ejemplo. Utilizamos la herramienta PSeInt [11] para realizar los algoritmos en Pseudocódigo y visualizar los resultados de salida del algoritmo.

6.3 Variables dependientes

Vamos a tener en cuenta el tiempo empleado por el conjunto de la clase, en alcanzar el nivel idóneo de comprensión y aprendizaje deseado en fundamentos de programación: conceptos básicos, sintaxis de Java, tipos de datos y operadores, estructuras básicas de control de flujo, condicionales y bucles.

Necesitamos un instrumento de medida (véase TABLE I.), una forma de medir el aprendizaje, y es mediante una rúbrica comprensiva, con una escala de 0 a 5, considerando 5 el grado más alto de comprensión y 0, ninguna comprensión, que utilizamos la misma para cada alumno, donde puntuamos el nivel comprensivo que alcanzan. Tras recibir las actividades resueltas de cada alumno y trabajar directamente con ellos, calificamos su nivel.

En el momento que hemos llegado a ver y practicar, todos los contenidos del bloque inicial de fundamentos de programación, y además se ha alcanzado el nivel de aproximadamente 4, de puntuación media, en las rúbricas comprensivas de todos los alumnos de la clase, es cuando la clase tiene, a nivel general, un nivel idóneo de comprensión. Si no se alcanza el nivel de comprensión y aprendizaje, habrá que volver atrás y repetir ejercicios y explicaciones hasta alcanzarlo.

TABLE I. RÚBRICA COMPRENSIVA

N	Detalle
5	Comprensión total de la actividad, todos los elementos requeridos
4	Comprensión, con un alto grado de elementos requeridos
3	Comprensión parcial, con algunos de los elementos requeridos
2	Algo de comprensión, pero no se incluyen elementos requeridos
1	No se comprende la actividad que se ha planteado
0	No se realiza nada

6.4 Procedimiento

Durante 3 cursos académicos, y con los grupos antes mencionados, trataremos de medir el tiempo invertido, empleado por estos grupos de alumnos, en conocer, aprender, un primer bloque de fundamentos de programación, con un nivel idóneo.

Es necesario por parte del alumno, ir esquematizando, asimilando y comprendiendo los conceptos, para poder avanzar. Tras dar la explicación y proponer ejercicios de ejemplo, los alumnos realizan actividades que entregan resueltas. Además solventamos dudas a los alumnos que tienen dificultades.

7 Resultados

En este apartado, presentamos los resultados obtenidos para cada grupo.

- **Grupo 1**

Conocer y aprender los fundamentos de programación en Java, ocupa muchas horas de estudio. Tras la realización de una evaluación continua (véase TABLE II.), es necesario un tiempo de 40 horas de estudio-aprendizaje y realización de ejercicios, hasta que se alcanza un nivel idóneo en conocimientos. 3 alumnos han alcanzado una comprensión total, 4 alumnos una comprensión alta y 2 alumnos una comprensión parcial.

TABLE II. GRUPO 1

Nº DE ALUMNOS, 2º Bachillerato TIC2	Compresión total de la actividad. Incluye todos los elementos requeridos.	Compresión, con un alto grado de elementos requeridos en la actividad.	Compresión parcial, con algunos de los elementos requeridos.	Algo de comprensión, pero no se incluyen los elementos requeridos.	No se comprendió la actividad planteada.	No se realiza nada de nada.	Nivel Medio de Comprensión y Aprendizaje.	Tiempo Empleado en Horas.
3	5						4,11	40
4		4						
2			3					

- **Grupo 2**

Practicamos evaluación continua y es necesario un tiempo de 31 horas de estudio-aprendizaje y realización de ejercicios (véase TABLE III.), hasta que se alcanza un nivel idóneo de conocimientos, en fundamentos de programación estructurada en Java. 9 alumnos han alcanzado una comprensión total, 7 alumnos una comprensión alta, 6 alumnos una comprensión parcial, y 2 alumnos algo de comprensión.

TABLE III. GRUPO 2

Nº DE ALUMNOS, 1º FP Grado Superior en Informática	Comprensión total de la actividad. Incluye todos los elementos requeridos.	Comprensión, con un alto grado de elementos requeridos en la actividad.	Comprensión parcial, con algunos de los elementos requeridos.	Algo de comprensión, pero no se incluyen los elementos requeridos.	No se comprendió la actividad planteada.	No se realiza nada de nada.	Nivel Medio de Comprensión y Aprendizaje.	Tiempo Empleado en Horas.
9	5						3,96	31
7		4						
6			3					
2				2				

- **Grupo 3**

Habiendo realizado evaluación continua, es necesario un tiempo de 34 horas de estudio-aprendizaje y realización de ejercicios (véase TABLE IV.), hasta que alcanzan un nivel idóneo en conocimientos, en fundamentos de programación estructurada en Java. 2 alumnos han alcanzado una comprensión total, 3 alumnos una comprensión alta y 1 alumno una comprensión parcial.

TABLE IV. GRUPO 3

Nº DE ALUMNOS, 2º Bachillerato TIC2	Compresión total de la actividad. Incluye todos los elementos requeridos.	Compresión, con un alto grado de elementos requeridos en la actividad.	Compresión parcial, con algunos de los elementos requeridos.	Algo de comprensión, pero no se incluyen los elementos requeridos.	No se comprendió la actividad planteada.	No se realiza nada de nada.	Nivel Medio de Comprensión y Aprendizaje.	Tiempo Empleado en Horas.
2	5						4,17	34
3		4						
1			3					

- **Grupo 4**

Comenzamos a enseñar los conceptos y estructuras básicas de la programación estructurada, en Pseudocódigo.

Hemos practicado una evaluación continua, y alcanzar un nivel idóneo en conocimientos sobre fundamentos de programación estructurada en Pseudocódigo, nos ocupan menos tiempo de estudio y comprensión que en los anteriores grupos. Empleamos un tiempo de 20 horas (véase TABLE V.). 3 alumnos han alcanzado una comprensión total, 2 alumnos una comprensión alta y 1 alumno algo de comprensión.

TABLE V. GRUPO 4

Nº DE ALUMNOS, 1º FP Grado Medio en Informática	Compresión total de la actividad. Incluye todos los elementos requeridos.	Compresión, con un alto grado de elementos requeridos en la actividad.	Compresión parcial, con algunos de los elementos requeridos.	Algo de comprensión, pero no se incluyen los elementos requeridos.	No se comprendió la actividad planteada.	No se realiza nada de nada.	Nivel Medio de Comprensión y Aprendizaje.	Tiempo Empleado en Horas.
3	5						4	20
2		4						
1				2				

8 Conclusiones

El Pseudocódigo es uno de los métodos más utilizados para el diseño de algoritmos, y es independiente del lenguaje de programación que se vaya a utilizar. Al estar escrito en lengua materna (castellano en nuestro caso), su sintaxis sencilla y claridad de código, puede facilitar la comprensión, y hacen de él una herramienta indispensable para la educación, para la enseñanza de la programación.

Los grupos que ha aprendido con Java, han empleado un tiempo medio de 35 horas, frente a las 20 horas del grupo de Pseudocódigo. Como conclusión, resulta muy llamativa, la reducción del tiempo necesario, empleando el Pseudocódigo, para la comprensión del bloque de fundamentos de programación: conceptos básicos, sintaxis, tipos de datos y operadores, estructuras básicas de control de flujo, condicionales y bucles.

El efecto pedagógico del uso del Pseudocódigo en idioma materno, para el aprendizaje de la programación, puede ser un efecto positivo. Al parecer, la asimilación por parte del alumnado, de los mecanismos, de los fundamentos de programación, se produce más rápida y eficientemente a través del Pseudocódigo. Puede ser una manera más amena, de abordar el estudio de los citados contenidos, frente a empezar directamente con un lenguaje de programación de alto nivel como Java. Parece ser, que con el Pseudocódigo, el alumno experimenta una menor sobrecarga mental.

Aunque es cierto, que algún grupo era muy numeroso, y consecuentemente, puede llevar a una disminución de la atención personalizada por parte del profesor. Los alumnos, a los que hemos enseñado Pseudocódigo de entrada, lo han afrontado sin mucha dificultad, y probablemente se enfrentarían a cualquier lenguaje de programación de alto nivel, con mayor facilidad, habiendo adquirido una base en programación.

En definitiva, el Pseudocódigo parece ser una herramienta indispensable, que conviene tener en cuenta y utilizar, en la enseñanza de la programación estructurada a los alumnos.

No se ha analizado la transición de Pseudocódigo a Java, sería interesante como trabajo futuro, ya que las exigencias sintácticas de Java, pudieran ralentizar el aprendizaje de los estudiantes.

9 Referencias

- [1] N. Bennett, "Introduction to Algorithms and Pseudocode." <https://www.researchgate.net/publication/309410533>, August 2015.
- [2] G. Michaelson, "Teaching Programming with Computational and Informational Thinking." <https://www.researchgate.net/publication/273758977>, March 2015.
- [3] I-Jung Chen, 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.

- [4] T.Koulouri, S. Lauria, R. D. Macredie, "Teaching Introductory Programming: a Quantitative Evaluation of Different Approaches." *ACM Transactions on Computing Education* · December 2014.
- [5] Robins, A., Rountree, J., and Rountree, N. 2003. "Learning and teaching programming: A review and discussion." *Computer Science Education*, 13. 2, 137-172.
- [6] Zelle, J. M. 1999, March. "Python as a first language." In *Proceedings of 13th Annual Midwest Computer Conference*. Vol. 2.
- [7] L. Mannila, M. Peltomäki, T. Salakoski (2006) "What about a simple language? Analyzing the difficulties in learning to program." *Computer Science Education*, 16:3, 211-227.
- [8] O. Arboleda, "Definición del lenguaje de programación EPLOAM para la ejecución de Pseudocódigo y su compilador." *Scientia et Technica*, vol. 2(48), 2011.
- [9] P. Novara. PSeInt. (2020) <http://pseint.sourceforge.net/>
- [10] F. Huari Evangelista and P. José Novara, "Intérprete para probar un programa escrito en Pseudocódigo." *Industrial Data*, vol. 17(1), pp. 101-110. 2014.
- [11] J. Sweller, "Cognitive load theory, learning difficulty, and instructional design," *Learning and Instruction*, vol. 4(4), pp. 295-312, April 1994.
- [12] G.A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, vol. 63(2), pp. 81-97, 1956.
- [13] N. Cowan, "The magical number 4 in short-term memory: A reconsideration of mental storage capacity." *Behavioral and Brain Sciences*, vol. 24(1), pp. 87-114, 1956.
- [14] M.L. Navarro Navarro, "Estrategias de motivación en el aula y su impacto académico." *Universidad de Almería*. 2017.
- [15] G. Cooper, "Research into Cognitive Load Theory and Instructional Design at UNSW." *University of New South Wales, Australia*, December 1998.
- [16] J. Sweller, J.J.G. van Merriënboer, Fred G.W.C. Paas, "Cognitive Architecture and Instructional Design." *Educational Psychology Review*, September 1998.
- [17] J.J.G van Merriënboer, P. A. Kirschner, L. Kester, "Taking the Load Off a Learner's Mind Instructional Design for Complex." *Open University of the Netherlands, Educational Psychologist*, March 2003.
- [18] G. Shum, A. Conde, C. Díaz, F. Martínez and L. Molina, "Lenguaje y rendimiento escolar: Un estudio predictivo," *Comunicación, Lenguaje y Educación*, vol. 2(5), pp. 69-79, 1990.
- [19] J. Palomar Lever, S.I. Montes de Oca Mayagoitia, A.M. Polo Velázquez, A. Victorio Estrada, "Factores explicativos del rendimiento académico en hijos de inmigrantes mexicanos en Nueva York," *Psicología Educativa*, vol. 22(2), pp. 125-133, 2016.
- [20] Q.A. La influencia de la lengua materna en el éxito escolar es determinante. 2017. <https://elfarodeceuta.es>
- [21] R. Análís Carreón "Desarrollo de técnicas de motivación para mejorar el rendimiento de los alumnos en la materia de matemáticas." *Universidad Autónoma de Nuevo León*. 2002.