

Reestructuración pedagógica de la asignatura lógica y algoritmia para el mejoramiento de la enseñanza en los programas de ingeniería de la Universidad Pontificia Bolivariana - Seccional Bucaramanga

Jairo Bernardo Viola-Villamizar & Diana Teresa Gómez-Forero

Facultad de Ingeniería de Sistemas e Informática, Universidad Pontificia Bolivariana, Bucaramanga, Colombia. jairo.viola@upb.edu.co,
diana.gomez@upb.edu.co

Resumen— Lógica y algoritmia es un curso obligatorio del plan de estudios de los programas de la Escuela de Ingeniería de la Universidad Pontificia Bolivariana Seccional Bucaramanga, donde se enseña al estudiante los fundamentos del pensamiento lógico matemático mediante la lógica de programación. Sin embargo, el desempeño de los estudiantes en la asignatura no era el esperado, debido a la falta de motivación hacia la asignatura. Este artículo, presenta la reestructuración realizada a la asignatura con el fin de mejorar la motivación y desempeño de los estudiantes durante el curso. Para esto se determinaron las causas del bajo desempeño de los estudiantes, a partir de las cuales se planteó la transformación de la asignatura, basado en un enfoque constructivista que relacione la enseñanza teórica, el aprendizaje y las notas del curso. Para esto, se desarrollaron cambios como la unificación de los materiales de clase, los exámenes, del lenguaje de programación empleado, así como la incorporación de estrategias lúdicas para reforzar el aprendizaje de los estudiantes. Además, como instrumento de medida del aprendizaje, se incorporaron herramientas TIC para llevar trazabilidad del desempeño de estudiantes y docentes en conjunto con las calificaciones de la asignatura. Los resultados de esta reforma muestran que los estudiantes tienen una mayor apropiación de las temáticas abordadas, así como una mejor actitud hacia la lógica y algoritmia debido a una evolución de su nivel de aprendizaje, evidenciado en un mejor rendimiento académico a nivel individual y grupal.

Palabras Clave— lógica y algoritmia; modelo constructivista; taxonomía SOLO; Python; aprendizaje y enseñanza; estrategias lúdicas; Moodle

Recibido: 10 de julio de 2018. Revisado: 30 de julio de 2018. Aceptado: 3 de septiembre de 2018.

Pedagogic restructuring of the logic and algorithms course for better teaching in the engineering programs of the Universidad Pontificia Bolivariana - Seccional Bucaramanga

Abstract— Logic and algorithm is a mandatory course on the engineering school of the Universidad Pontificia Bolivariana Seccional Bucaramanga, which teaches the fundamentals of the logical-mathematical thinking through programming. However, the student's performance is not suitable due to a lack of motivation for the course. This paper presents the pedagogical restructuring of the class to improve the student's motivation and performance. The lack of motivation causes was analyzed to propose a course with a constructivist focus based on the SOLO taxonomy, which relates the theory, learning, and grades with the learning level of the students. Many changes were introduced in the course as new learning goals, a new programming language, ludic resources for the students learning reinforcement, and IT tools for the students' performance tracking. Obtained results show that students assimilate better the logic mathematical thinking, as well as they reach a new learning level evidenced in a better individual and group performance.

Keywords— logic and algorithms; Python, teaching and learning; SOLO taxonomy; ludic; Moodle

1 Introducción

Un ingeniero en formación necesita aprender un conjunto de conocimientos y habilidades básicas para dar solución a los problemas dentro de su disciplina. Algunos de estos problemas, tales como el modelado y simulación de procesos, el planteamiento de problemas de optimización, el diseño asistido por computador, el diseño de planes y flujos de trabajo para obras civiles y gestión de proyectos entre otras, son situaciones que un ingeniero enfrentara durante su ejercicio profesional. Sin embargo, aunque todas estas habilidades son necesarias para el ingeniero, se requiere de un modelo de pensamiento lógico-matemático que las encadene, para lograr abstraer los problemas del mundo real o formal y representarlos mediante estructuras lógicas y matemáticas básicas que permitan obtener una solución a esa situación problema.

Sin embargo, ¿en qué momento se adquiere este pensamiento lógico-matemático? Aunque este pensamiento se comienza a desarrollar desde la educación primaria y se mantiene en permanente desarrollo en la educación secundaria, superior e incluso después, son asignaturas como lógica y algoritmia, o lógica de programación, o programación de computadores I, o algunos otros cursos con nombres similares dentro del cada universidad donde se refuerza este modelo de pensamiento para los ingenieros en formación. Pero ¿qué hace la diferencia para que algunos ingenieros hayan logrado adquirir este modelo de pensamiento mientras otros estudiantes solamente expresan que no tiene relación con su disciplina?

Debido a esto, la Escuela de Ingeniería de la Universidad Pontificia Bolivariana Seccional Bucaramanga (UPB), se mantiene atenta al impacto de sus asignaturas en la formación de sus estudiantes, y por supuesto que la asignatura de Lógica y Algoritmia está sujeta a este tipo de cuestionamientos. En el año 2015, se realizó una evaluación de la asignatura [1], la cual evidenció debilidades que reducían el impacto esperado en el aprendizaje de esta asignatura y que fueron analizadas y sometidas a un proceso de reestructuración [2,3]. En este artículo se presenta el proceso de reestructuración pedagógica

Como citar este artículo: Viola-Villamizar, J.B. and Gómez-Forero, T., Reestructuración pedagógica de la asignatura lógica y algoritmia para el mejoramiento de la enseñanza en los programas de ingeniería de la Universidad Pontificia Bolivariana Seccional Bucaramanga. Educación en Ingeniería, 14(27), pp. 41-47, Agosto 2018 - Febrero, 2019.

de la asignatura lógica y algoritmia para la escuela de ingenierías de la UPB basada en la teoría constructivista, que facilite el cambio de actitud de los estudiantes hacia la asignatura, creando un modelo mental que permita organizar, la percepción de los contenidos de la asignatura y su importancia enfocada en cada campo de la ingeniería.

Para esto, inicialmente se realizó un análisis de la asignatura, el cual evidencio las debilidades en la enseñanza teórica del curso, las cuales dificultaban el aprendizaje de la asignatura por parte de los estudiantes, obteniendo un bajo desempeño académico reflejado en las notas. Posteriormente, basado en este análisis, se plantea la reestructuración de la asignatura, la cual es soportada en el modelo constructivista y apoyada en la taxonomía SOLO (Structure of observed Learning Outcomes) planteada por [1], la cual busca un alineamiento constructivo entre los objetivos de enseñanza del curso, las actividades que desarrolla el alumno y los instrumentos de evaluación utilizados para medir el desempeño de los estudiantes y docentes en el curso. Posteriormente, se describe la reestructuración de la asignatura, la cual se basa en la unificación de contenidos, la renovación del syllabus, la estandarización de un lenguaje de programación, la inclusión de herramientas didácticas para el aprendizaje de los conceptos básicos, y la inclusión de herramientas TIC para la medición de la trazabilidad del curso por parte de estudiantes y docentes. Después, se presentan los resultados de la reestructuración, los cuales han sido evaluado durante cuatro semestres. Como índice de desempeño cuantitativo del curso se emplearon las notas obtenidas en dos grupos diferentes de lógica y algoritmia.

El artículo se encuentra estructurado de la siguiente forma. En la Sección II se presenta el análisis de la asignatura. En la Sección III, se presenta el modelo constructivista basado en la taxonomía SOLO. La sección IV muestra los cambios que hicieron parte de la reestructuración de la asignatura. En la Sección V se muestra y discute el efecto en términos de notas como consecuencia de los cambios implementados en la asignatura. Finalmente se presentan las conclusiones.

2 Asignatura lógica y algoritmia

En esta sección se presenta una descripción de la asignatura lógica y algoritmia de la UPB y el análisis realizado en 2015 acerca de las debilidades pedagógicas de la asignatura.

2.1 Descripción del curso

El curso de Lógica y Algoritmia, en la Universidad Pontificia Bolivariana Seccional Bucaramanga, es coordinado desde la Facultad de Ingeniería de Sistemas e Informática. [2,3]. La intensidad semanal en horas de la asignatura corresponde a una hora de teoría, y tres horas de laboratorio lo que equivale dos créditos académicos. Las clases teóricas se imparten de forma magistral, y los laboratorios son utilizados para llevar a la práctica los conceptos vistos en las clases teóricas. Los laboratorios presentan una infraestructura de 20 computadores que se renuevan cada dos años, y se garantiza un computador por estudiante. Las aulas, así como los laboratorios cuentan con videobeam para proyección del material audiovisual. Hasta el primer semestre de 2015, el lenguaje de programación utilizado

fue C++, luego de una reforma el lenguaje pasó a ser Python [4].

La metodología es orientada por competencias del saber, del ser y del hacer. Asimismo, las clases son presenciales y se utilizan como estrategias de enseñanza charlas magistrales, laboratorios (los cuales incluyen pre-informes, desarrollo de actividades, e informes finales), indagación en el contexto externo al aula de clase, ejercicios, analogías, proyecto de aula, foros, entre otros. Se cuenta con la plataforma Moodle como apoyo a la presencialidad, herramienta que soporta el diseño instruccional de la asignatura y que permite tener el material audiovisual utilizado por el docente, lecturas y ejercicios complementarios, buzones para el envío del trabajo en el aula y el trabajo independiente. La calificación de la asignatura se obtiene a partir de cinco notas: seguimiento del primer corte (que la componen quices, tareas y trabajos), examen parcial, seguimiento del segundo corte, examen final y laboratorio (que lo conforman todos los trabajos prácticos desarrollados semana a semana en las horas de laboratorio).

2.2 Análisis del curso

Finalizando el primer semestre de 2015, se realizó un análisis de la asignatura, evaluando el desempeño de los estudiantes y el impacto de la asignatura en su proceso de formación. A nivel de aprendizaje, una debilidad consiste en que al enfocar lógica y algoritmia desde la perspectiva de la programación para enseñar el pensamiento lógico matemático, los estudiantes consideraban que la asignatura no era parte importante de su plan de estudios, lo cual producía una reticencia a aprender la teoría básica y desarrollar los ejercicios prácticos del curso.

Además, muchos de los ejemplos utilizados en la enseñanza de los conceptos básicos de asignatura, eran enfocados desde una estricta perspectiva matemática que no se encuentra aplicada y contextualizada dentro del contexto de cada programa de ingeniería, lo cual desmotiva el aprendizaje por parte de los estudiantes.

Por otra parte, desde la perspectiva de enseñanza de la asignatura, cada docente tenía su propio material de exposiciones, y se carecía de uniformidad en las expresiones técnicas, y en la complejidad de los ejercicios de clase y exámenes. Así mismo, el lenguaje de programación utilizado era C++, y su sintaxis, la definición de tipos de datos, el uso de punteros, entre otros detalles representaba para muchos estudiantes un obstáculo adicional que no les facilita enfocarse en el pensamiento lógico matemático sino en la sintaxis del lenguaje de programación.

Un reflejo de estas debilidades se ve en las calificaciones de los estudiantes. La Fig. 1 representa mediante diagrama de cajas y bigotes el comportamiento típico de un curso de lógica y algoritmia anterior al proceso de reestructuración. Este curso está conformado por 30 estudiantes de diferentes programas de ingeniería de la UPB. Como se puede observar, la primera caja de izquierda a derecha muestra las notas obtenidas durante el primer corte que en promedio se aproximan a 4.2 con una variación menor a tres décimas por encima y por debajo. La siguiente caja corresponde a la nota examen parcial, allí más del 50% de los estudiantes cuenta con una calificación superior a

4.0 y la nota más baja estuvo por encima de 3.4. La tercera caja que representa la nota de laboratorio muestra que todos los estudiantes obtuvieron notas por encima de 4.2 hasta 4.7. La caja morada, que representa las notas del segundo corte, se observa que las notas oscilaron entre 3.2 a 3.9. Finalmente, la caja cian que representa la nota del examen final muestra que más del 50% de los estudiantes tienen una calificación inferior a 1.0, a excepción de algunos casos particulares. Es importante resaltar que este comportamiento es similar para casi todos los cursos de lógica y algoritmia anteriores a la reestructuración.

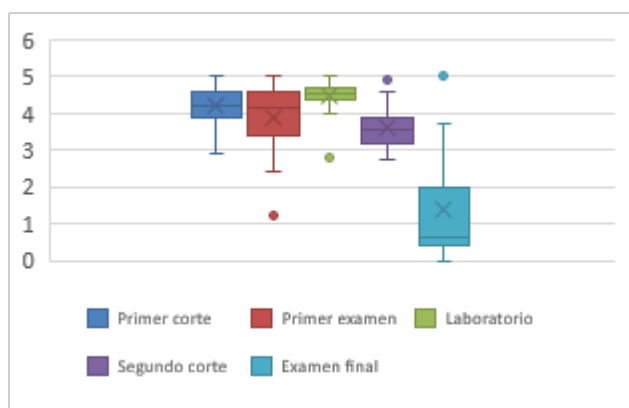


Figura 1. Calificaciones de lógica y algoritmia en un curso del segundo semestre de 2015.

Fuente: Autores

3 Reestructuración del curso de lógica y algoritmia

De acuerdo con estos resultados y las debilidades identificadas en la sección II, la facultad de ingeniería de sistemas e informática propuso la reestructuración pedagógica de la asignatura lógica y algoritmia para mejorar la apropiación del conocimiento por parte de los estudiantes, así como mejorar las prácticas de enseñanza, basados en la taxonomía SOLO y un modelo constructivista.

3.1 Taxonomía SOLO

De acuerdo con [1], el profesor es una parte fundamental en el proceso de aprendizaje y enseñanza de un curso universitario, ya que es quien planifica y estructura los contenidos teóricos de un curso, establece las actividades de aprendizaje que los estudiantes deben adquirir dentro del proceso formativo y crea un ambiente adecuado dentro del aula de clases que permita el aprendizaje de los estudiantes.

Además, [1] propone que el aprendizaje es un ejercicio constructivista, en el cual los estudiantes son sometidos a un conjunto de actividades de aprendizaje que llevan al estudiante hacia nuevas estructuras de pensamiento o simplemente se limiten a una acumulación de conocimientos. Por lo tanto, se requiere que el profesor defina claramente sus objetivos de aprendizaje antes de realizar su ejercicio docente.

Para lograrlo, la taxonomía SOLO es una herramienta que permite establecer los resultados esperados del proceso de aprendizaje dentro de un ambiente universitario y cuenta con cinco niveles de conocimiento de acuerdo con los objetivos de aprendizaje [1,5].

El primer nivel preestructural, es el mas bajo de todos y corresponde a la insuficiencia de conocimiento que se presenta cuando el estudiante no ha cumplido con los objetivos de aprendizaje del curso, lo cual le impide dar solución a una situación problema. El segundo nivel conocido como uniestructural, es aquel donde los estudiantes pueden identificar un procedimiento sencillo y dar respuesta a este único aspecto de la situación problema sin ser capaces de ir más allá de ese procedimiento. El tercer nivel o multiestructural, el estudiante es capaz de identificar varios procedimientos relevantes y dar una respuesta estructurada para cada uno de estos aspectos en forma secuencial.

El cuarto nivel relacional, el estudiante muestra una comprensión global de la situación y brinda una respuesta integrada que relaciona todos los aspectos y procedimientos.

El quinto nivel o nivel de abstracción extendida brinda respuestas basadas en un proceso de teorización, generalización, que permiten formular hipótesis y reflexionar en aspectos que van más allá del problema a resolver.

3.2 Objetivos de aprendizaje

En el caso de lógica y algoritmia, de acuerdo con el análisis realizado en la sección II, se puede decir que debido a las debilidades como la falta de motivación o la falta de enfoque disciplinar de la asignatura hacia cada uno de los programas de ingeniería, los estudiantes se encuentran en su mayoría entre los niveles de aprendizaje preestructural y uniestructural presentados en la taxonomía SOLO.

Ya que se desea que los estudiantes se encuentren dentro de los niveles relacional y de abstracción extendida, es necesario reformular los objetivos de aprendizaje de la asignatura, que orienten el ejercicio de reestructuración. Para esto, el propósito del curso se enfocó más en fortalecer el pensamiento lógico-matemático, que en dominar comandos de un lenguaje de programación [4]. Teniendo en cuenta que la UPB cambio su modelo pedagógico al modelo basado en competencias, las nuevas competencias cognitivas del curso buscan que el estudiante:

- Abstrae problemas del mundo real y formal, los analiza desde la perspectiva de sus entradas, proceso y salidas, y los modela a través de diagramas de flujo y pseudocódigo”.
- “Entiende los diagramas de flujo y el pseudocódigo como representación de situaciones del mundo real y formal”.
- “Comprende la correspondencia entre un lenguaje simbólico como el de los diagramas de flujo y los comandos propios de un lenguaje computacional de alto nivel”.

Por otro lado, desde las competencias actitudinales se forma para que el estudiante:

- “Actúe responsablemente frente a aspectos de corrección en los algoritmos y soluciones computacionales propuestas”

En las competencias pragmáticas se busca que el estudiante:

- “Elabore diagramas de flujo o pseudocódigo que den cuenta de la formalización algorítmica de problemas concretos y abstractos certificados a través de técnicas de

validación y comprobación, en diversos campos de aplicación de la ingeniería”.

- “Transforme diagramas de flujo y pseudocódigo a lenguaje de programación de alto nivel, que respondan con la semántica, gramática y sintaxis de este, a partir de la elaboración del código fuente enmarcado en principios de seguridad, la ejecución satisfactoria de un conjunto de pruebas y el dominio de ambientes integrados de desarrollo”.

Además, el contenido del curso se concentró en el pensamiento lógico matemático combinado con el pensamiento algorítmico, enfatizando en las fases del proceso de la programación (definición del problema, análisis, diseño, codificación), el concepto de algoritmo, y sus formas de representación, los tipos de datos, expresiones, operadores y sus jerarquías, las estructuras de control en los diagramas de flujo y pseudocódigo como entradas, salidas, estructuras secuenciales, condicionales, repetitivas, modulares, arreglos unidimensionales y bidimensionales, las técnicas de validación y comprobación. Finalmente, un lenguaje de programación para llevar a código las estructuras anteriores.

3.3 Elección de Python como lenguaje de programación

Originalmente el lenguaje de programación empleado en lógica y algoritmia fue C++, debido a que es un lenguaje de propósito general, muy popular, el de mayor dominio por parte de los docentes de la asignatura. Sin embargo, la sintaxis estricta del lenguaje se convierte en una dificultad para los principiantes quienes realizan un desgaste innecesario tratando de dominar la sintaxis, más que privilegiando el desarrollo de la algoritmia. La popularidad de Python con propósito educativo se debe a que es un lenguaje de programación débilmente tipado, que se asemeja al inglés, posee lectura y escritura simple, lo que lo hace ideal para principiantes en el mundo de la programación que buscan reforzar el pensamiento lógico matemático. En la Fig. 2 se muestra que Python es el cuarto lenguaje más empleado en universidades para el aprendizaje de la programación.

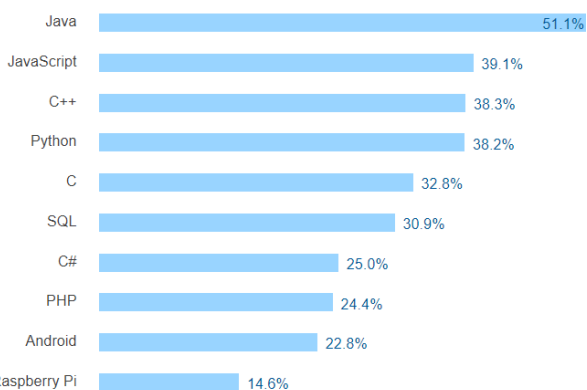


Figura 2. Lenguajes más empleados en la academia de acuerdo con StackOverflow en 2016.

Fuente: [6] Stack Overflow, Stack Overflow survey 2016, [En línea]. Available at: <https://insights.stackoverflow.com/survey/2016>.

Python, no es sólo un lenguaje de propósito académico, sino que posee gran cantidad de bibliotecas que le dan fortalezas para muchos campos de aplicación. Por ejemplo, numpy está orientada a la computación científica, maneja estructuras vectoriales y matriciales; matplotlib se utiliza para representación gráficas. De acuerdo con el último reporte de StackOverflow [6], Python encabeza la lista de lenguajes de programación empleados en minería de datos y procesamiento matemático. La Fig. 3 muestra que supera a lenguajes tradicionales como C++ y especializados en el manejo de datos como R. Además, Python es un lenguaje de programación de código abierto, lo cual posibilita que cualquier persona lo pueda descargar en forma gratuita sin ningún tipo de licencia. Así que los argumentos anteriores condujeron a seleccionar Python como el nuevo lenguaje para Lógica y Algoritmia.

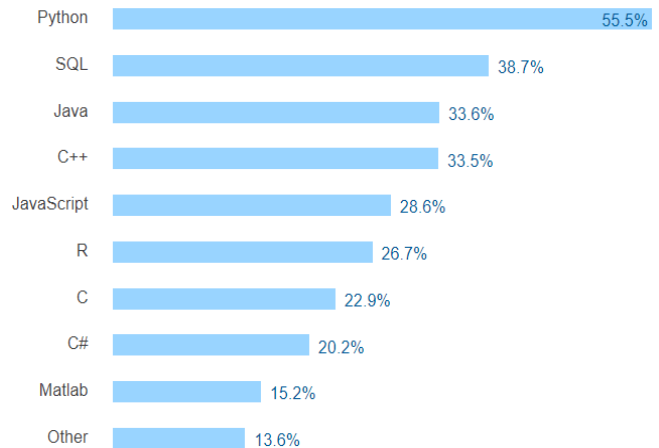


Figura 3. Lenguajes más empleados para minería de datos y procesamiento matemático de acuerdo con StackOverflow en 2016

Fuente: [6] Stack Overflow, Stack Overflow survey 2016, [En línea]. Available at: <https://insights.stackoverflow.com/survey/2016>.

3.4 Estrategias lúdicas y tutoriales

Para superar las deficiencias pedagógicas centradas en métodos y ejemplos tradicionales, se exploraron estrategias lúdicas para la enseñanza de la lógica. Una de ellas es el sitio web Hora del Código [7]. Este sitio permite a los estudiantes introducirse en el pensamiento lógico matemático a través de juegos y actividades.

En la Fig.4 se presenta un ejemplo de uno de los juegos utilizados en la plataforma hora del código. Allí un robot debe alcanzar montones de chatarra, y para ello se le programan una secuencia de pasos; en la medida que avanza tiene que tomar decisiones, y realizar repeticiones, para los cuales involucra estructuras condicionales y cíclicas.

Por otra parte, la plataforma permite al docente crear un curso donde puede registrar el avance de los estudiantes en los temas de programación. En la Fig. 5 se observa un ejemplo de uno de los cursos de lógica y algoritmia donde se puede ver la cantidad de niveles completados por el estudiante, así como las líneas de código que ha utilizado para resolver las situaciones presentadas.



Figura 4. Ejemplo de juego en la Hora de código
Fuente: [7] Code.org, Hora del codigo, [En línea]. Available at: code.org.

Además, con el objetivo de familiarizar al estudiante con el lenguaje de programación Python, que es el soporte práctico del curso para el aprendizaje del pensamiento lógico matemático, se presentan a los estudiantes los tutoriales de Codecademy.com para el aprendizaje de Python [8]. En la Fig. 6 se muestra el entorno de trabajo, el cual muestra las instrucciones a seguir, un espacio para codificar y una pantalla para visualizar los resultados del ejercicio.

Nombre	Niveles Completados	Líneas de Código
Jose David Ordoñez Leal	100	469
James Méndez Gómez	162	745
Andrés Kamilo Peña Ramírez	68	154
Joan Sebastian riveros lozada	123	453
Andres Kamilo Peña Ramirez	108	420

Figura 5. Desempeño de un curso de lógica y algoritmia en hora de código
Fuente: Autores

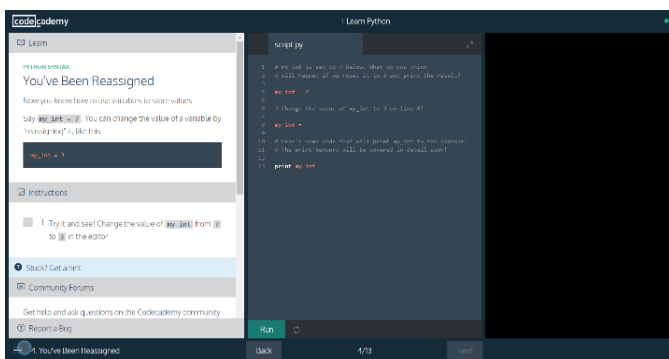


Figura 6. Plataforma de code academy para Python
Fuente: [8] Code academy, [En línea]. Available at: <https://www.codecademy.com/>.

3.5 Herramientas TIC de apoyo a la presencialidad

Tomando en consideración varias características del curso de lógica y algoritmia como el número considerable de grupos debido a la cantidad de estudiantes y los múltiples docentes que imparten el curso, se requiere buscar la uniformidad en el

material base de enseñanza incluyendo las prácticas de laboratorio. Este último es recibido semanalmente, por lo que se requiere de un buzón para la recepción de los exámenes, así como una forma de medir la trazabilidad del progreso de cada grupo para evaluar si todos alcanzan la base mínima esperada de desarrollo del plan de trabajo semestral.

Por lo tanto, se hace imprescindible contar con un entorno virtual de aprendizaje de apoyo a la presencialidad. Y la Universidad desde varios años atrás, encontró en Moodle [9,10] la mejor solución a esta necesidad.

El curso de lógica y algoritmia está configurado en Moodle como un solo módulo con un único diseño instruccional. En la Fig. 7. Se puede observar el diseño de un tema compuesto por cuatro secciones, como se indica en el manual de diseño de la UPB.

La sección de ¿Qué saber?, proyecta el alcance e importancia del tema. En la sección, ¿Qué hacer? Allí los estudiantes encontrarán las actividades a desarrollar, que son propias de cada grupo, y visibles sólo para ellos. La sección ¿Qué recursos usar? Ofrece el material base y unificado para el tema. Aparecen principalmente las presentaciones de cada clase. Finalmente, la sección de material complementario permite que los docentes sugieran lecturas o actividades adicionales que pueden ser particulares para cada grupo.

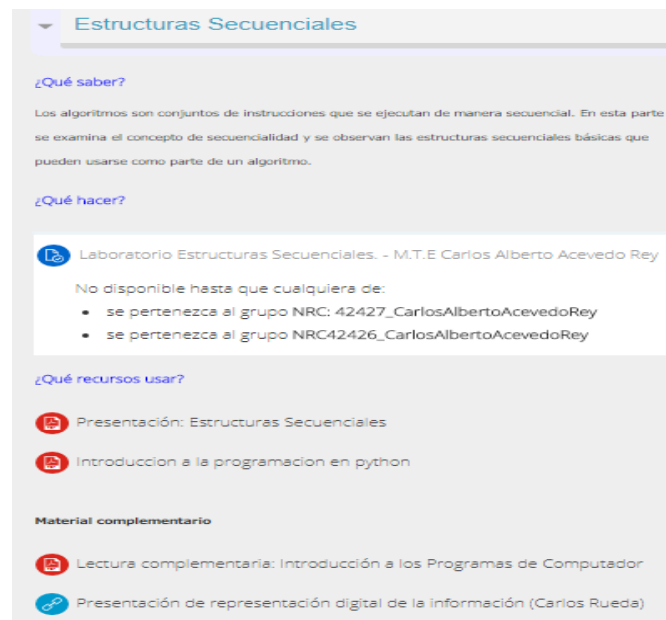


Figura 7. Ejemplo de una unidad temática dispuesta en Moodle
Fuente: Autores

3.6 Exámenes por departamento

Originalmente en cada grupo se desarrollaban los exámenes parcial y final con temarios definidos por sus docentes. Esto tiene ventaja para el estudiante pues se espera una evaluación muy coherente con los temas y ejercicios abordados durante las clases. Sin embargo, al realizar comparaciones entre los niveles de dificultad de los exámenes de los diferentes docentes se encontraron diferencias considerables que terminaban por no favorecer el aprendizaje, el cual se limitaba muchas veces a un nivel de evaluación uniestructural que dejaba indagar al estudiante más allá de la resolución de un ejercicio.

Por ello, se determinó que los exámenes serían los mismos para todos los grupos de lógica y algoritmia. Para ello se demandaba una logística en tiempo, y acceso a los laboratorios, seguridad de los temarios, que garantizara que todos los estudiantes podrían estar tomando el examen a la misma hora, para evitar problemas de fraude en los exámenes.

Con relación a los temarios, las preguntas debían ser enviadas por los docentes y seleccionadas por el coordinador, de acuerdo con diversos criterios de estilos de pregunta (de selección, falso y verdadero, pregunta abierta, interpretación de situaciones para creación de algoritmos, seguimiento de algoritmos o código, mapeo entre algoritmos y código).

4 Resultados de la reestructuración

Con el fin de evaluar la efectividad de la reestructuración de la asignatura de acuerdo con los nuevos objetivos de aprendizaje basados en la taxonomía SOLO, se analizarán en forma cualitativa diferentes aspectos como las opiniones de estudiantes y docentes que tomaron la asignatura. En forma cuantitativa, se estudiarán las notas de los cursos de lógica y algoritmia en los cuales se realizó la reestructuración, así como las estadísticas de usabilidad de la plataforma Moodle del curso.

En forma cualitativa se encontró que, a nivel teórico, se pudo observar que la unificación de materiales de clase y la contextualización de ejemplos y ejercicios mejoro el interés de los estudiantes, así como su comprensión del pensamiento lógico matemático. También, a nivel de aprendizaje el uso de las herramientas de code.org y CodeAcademy contribuyo a mejorar la comprensión del pensamiento lógico matemático y su relación con la programación. Asimismo, el uso de Python como lenguaje de programación permitió a los estudiantes enfocarse mas en el pensamiento lógico matemático en lugar de la sintaxis de la programación. A nivel de evaluación, la estandarización de exámenes permitió unificar los objetivos de evaluación de todos los cursos y enfocar al estudiante hacia la comprensión del pensamiento lógico matemático.

De esta forma, muchos estudiantes, ya son capaces de pasar de un nivel unidimensional a un nivel relacional, en el cual pueden comprender la relación entre los diferentes elementos de un mismo problema y dar una solución, basada en este caso en el pensamiento lógico matemático empleando un algoritmo.

Por otra parte, los estudiantes con mayor interés fueron capaces de pasar a un nivel de abstracción extendida, el cual les permite llevar el pensamiento lógico matemático a nuevas situaciones y ser más críticos ante el desarrollo de soluciones computacionales basadas en el pensamiento lógico matemático.

Además, el uso de Moodle como herramienta de mediación TIC para el seguimiento de la trazabilidad el curso permite a los estudiantes y docentes tener un espacio para el intercambio de información y el seguimiento del desarrollo del curso por parte del coordinador de la asignatura.

Para evidenciar los efectos de la reestructuración en forma cuantitativa se realizó la comparación de notas obtenidas de dos cursos de la asignatura lógica y algoritmia para el periodo 2016-20, los cuales serán denominados grupo 1 y grupo 2, con respecto a las notas obtenidas durante el periodo 2015-10 mostradas en la Fig.1. El grupo 1 corresponde a un grupo de 30 personas de las cuales el 95% corresponde a ingenieros civiles. Por otra parte, el grupo 2 corresponde a un grupo de 35 personas donde el 95% corresponden a ingenieros industriales. Estos

grupos se pueden considerar equivalentes con el grupo mostrado en la Fig.1, ya que los dos grupos tienen estudiantes de la escuela de ingenierías de programas donde, aunque el pensamiento lógico matemático es necesario, la programación produce una reticencia ya los estudiantes consideraban que esta no era parte importante de su plan de estudios.

En la Fig.8a se muestra un diagrama de caja y bigotes con las notas obtenidas por grupo 1 y en la Fig.8b las notas obtenidas por el grupo 2. Como se puede observar, existe un cambio significativo en la distribución de las notas del examen final, así como de la nota del segundo corte para el grupo 1 y el grupo 2 con respecto al grupo presentado en la Fig.1. Particularmente en el caso del examen final, se puede apreciar que más del 75% de las notas obtenidas en el grupo 1 y en el grupo 2 se encuentran por encima de una nota de 2.0, de los cuales el 50% en ambos grupos está alrededor del 3.4, resaltando una mejora significativa con respecto al grupo de referencia donde el 75% se ubicó por debajo de 2.0. Esto evidencia que los estudiantes son más perseverantes en el estudio del curso, lo que hace suponer que obtienen más conocimientos y de una manera más significativa.

Las notas del primer corte y del segundo corte muestran una mejora en el desempeño por parte de los estudiantes, donde el 50% de las notas se encuentran por encima de 3.5. También se puede notar la existencia de algunos puntos atípicos, correspondientes a casos particulares de estudiantes que no representan el comportamiento general del grupo.

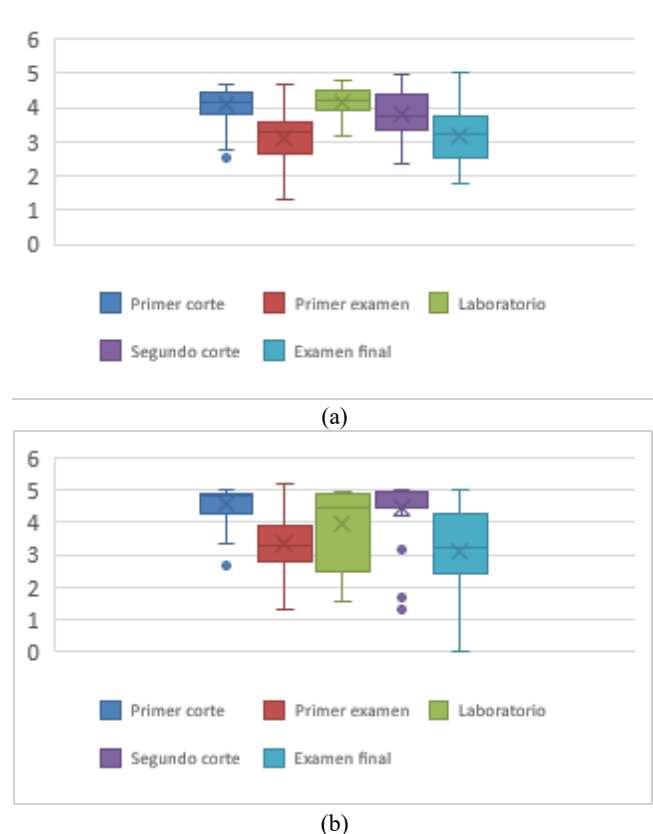


Figura 8. Diagrama de caja y bigotes para las notas (a) del grupo 1 y (b) del grupo 2 de lógica y algoritmia en el periodo 2016-20
Fuente: Autores

Otro indicador cuantitativo empleado para medir el nivel de desempeño de los estudiantes de la asignatura corresponde a las

estadísticas de usabilidad que entrega el curso en Moodle. En la Fig.9 se presentan los tiempos de dedicación promedio y total de los estudiantes y docentes a la plataforma a lo largo de tres semestres (2016-10, 2016-20 y 2017-10). En ellos estudiantes y docentes han totalizado 2341 horas de permanencia con un tiempo promedio por persona de 9 horas 57 minutos. Lo cual resulta favorable en las expectativas de actividad, uso de material unificado, desarrollo de actividades.

Evaluación de la eficacia del Ambiente Virtual de Aprendizaje	Rango de dedicación		
	Días	Hora	Minutos
Tiempo de duración total del Ambiente Virtual de Aprendizaje	156	14	00
Tiempo de dedicación total de los estudiantes y docente		2341	56
Tiempo de dedicación media de los estudiantes y docente	00	9	57

Figura 9. Tiempo de dedicación de los estudiantes y docentes en el uso de la plataforma Moodle del curso lógica y algoritmia.

Fuente: Autores

A partir de los resultados de la reestructuración de la asignatura, es posible decir que la mayoría de los estudiantes, de acuerdo con la taxonomía SOLO, han logrado pasar de un nivel preestructural y unidimensional a los niveles relacional y de abstracción. Esto se evidencia cuantitativamente en un mejor desempeño académico demostrado en un incremento en las calificaciones finales de la asignatura. Sin embargo, en forma conceptual y teórica, estos resultados son un indicador de una mayor apropiación del conocimiento de la asignatura, así como de una reducción considerable en la retención al aprendizaje del pensamiento lógico matemático. Además, la contextualización del pensamiento lógico matemático hacia cada una de las disciplinas de la ingeniería permite a los estudiantes visualizar un modelo de pensamiento que puede ser extendido en cada disciplina como herramienta para la solución de situaciones problema. Por otra parte, desde el punto de vista del aprendizaje, la inclusión de herramientas didácticas, la unificación de contenidos y el uso de un lenguaje de programación más sencillo permitió a los estudiantes dar un enfoque más centrado hacia el pensamiento lógico matemático y la abstracción de problemas del mundo real en lugar de estar atento únicamente a la sintaxis de un lenguaje.

5 Conclusiones

Este artículo presenta el proceso de reestructuración de la asignatura Lógica y Algoritmia de la UPB Bucaramanga, basada en un modelo constructivista soportado por la taxonomía SOLO para evidenciar los objetivos de aprendizaje deseados de la asignatura. Esta transformación se basó en la transformación del pensamiento lógico matemático de los estudiantes de la escuela de ingenierías que se encontraba en un nivel preestructural o unidimensional, a pasar a un nivel relacional o de abstracción extendida. Para esto, se requirió la definición de nuevos objetivos de evaluación, el cambio de lenguaje de programación a Python, la incorporación de material lúdico y tutoriales interactivos, uso de la plataforma Moodle como apoyo a la presencialidad con material unificado, proyectos de aula contextualizados para las carreras de los estudiantes, y exámenes unificados.

Los resultados de esta reestructuración evidenciaron que los estudiantes pasaron de un nivel preestructural a un nivel de relacional en la mayoría de los casos, lo cual se evidencia en forma cuantitativa en una mejora en las notas de la asignatura. Sin embargo, desde el punto de vista teórico, elementos como la unificación de contenidos y la inclusión de herramientas didácticas permitió a los estudiantes disminuir su retención hacia el curso y cambiar su nivel de conocimiento de forma efectiva, acercándose hacia los objetivos de enseñanza del curso. Desde el aprendizaje, se puede ver que hubo una mayor apropiación del conocimiento, así como una comprensión global del pensamiento lógico matemático aplicado desde el punto de vista de la programación. Cabe resaltar, que este proceso de reestructuración se mantiene en una constante revisión, para garantizar al estudiante no solo un ambiente adecuado de clase, sino un espacio para continuar con el desarrollo del pensamiento lógico matemático, buscando siempre un nivel de conocimiento relacional o de abstracción que le brinde herramientas para resolver problemas dentro de las disciplinas de cada estudiante de la escuela de ingenierías de la UPB Bucaramanga.

Referencias

- [1] Biggs, J., Calidad de la educación universitaria, Madrid: Narcea, 2005, ISBN 10:8427713983.
- [2] UPB Seccional Bucaramanga, Informe de notas de lógica y algoritmia, semestre 2015-10, Floridablanca, 2016.
- [3] UPB Seccional Bucaramanga, Informe final de notas de lógica y algoritmia del semestre 2016-20, Floridablanca, 2016.
- [4] UPB, Syllabus de Lógica y Algoritmia, 2015.
- [5] Huerta, M.P., Los niveles de Van Hiele y la taxonomía SOLO: un análisis comparado, una integración necesaria, Investigación Didáctica, 17(2), pp. 291-309, 1999.
- [6] Stack Overflow, Stack Overflow survey 2016, [online]. [Último acceso: 2017-08-12]. Available at: <https://insights.stackoverflow.com/survey/2016>.
- [7] Code.org, Hora del código, [En línea]. [Último acceso: 28-07-2017]. Disponible en: code.org.
- [8] Code academy, [online]. [Último acceso: 28 07 2017]. Available at: <https://www.codecademy.com/>.
- [9] Moodle, Moodle, [online]. Available at: https://docs.moodle.org/all/es/Acerca_de_Moodle.
- [10] UPB, UPB Virtual. BL Bucaramanga, 2017. [En línea]. Disponible en: <https://bl.bucaramanga.upb.edu.co/>.
- [11] JAVAWORLD of IDG, Python bumps off Java as top learning language, [online]. [Último acceso: 2017-08-12]. Available at: goo.gl/aj6PRw.
- [12] UPB, Diseño instruccional de la asignatura lógica y algoritmia, UPB, Bucaramanga, 2017.

D.T. Gómez-Forero, es Ing. de Sistemas y MSc. en Informática de la Universidad Industrial de Santander (Colombia). MSc. en Ciencias de la Computación de la Universidad de Oklahoma (USA). Trabaja para la Universidad Pontificia Bolivariana Seccional Bucaramanga. Su área de interés en investigación es la Ingeniería de Software y la Informática Educativa. ORCID: 0000-0001-5054-4864

J.B. Viola-Villamizar, es Ing. Electrónico y MSc en Ingeniería Electrónica de la Universidad Pontificia Bolivariana Seccional Bucaramanga, Colombia. Sus áreas de investigación son el control fraccionario, la robótica industrial, el procesamiento de imágenes y el reconocimiento de patrones. ORCID: 0000-0003-2739-5383