



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Evaluación del impacto en la motivación en el aprendizaje de los estudiantes de programación de computadores mediante el uso de una herramienta de evaluación automática

Hernán Darío Lozano Rojas

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2022

Evaluación del impacto en la motivación en el aprendizaje de los estudiantes de programación de computadores mediante el uso de una herramienta de evaluación automática

Hernán Darío Lozano Rojas

Tesis presentada como requisito parcial para optar al título de:
Magíster en Ingeniería - Ingeniería de Sistemas y Computación

Director:

Felipe Restrepo Calle, Ph.D.

Codirector:

Jhon Jairo Ramírez Echeverry, Ph.D.

Línea de Investigación:

Computación Aplicada - Educación en Ingeniería

Grupo de Investigación:

Programming Languages and Systems - PLaS

Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2022

Quiero agradecer primeramente al que hace todo posible en mi vida, a Dios por la sabiduría en cada momento, por las fuerzas para afrontar cada reto presente y por ser la luz que guía cada paso que doy, sé que todo lo puedo tomado de su mano y mi corazón estará agradecido eternamente por sus bendiciones.

A mis padres, Hernán Lozano y Yubieth Rojas, por ser mi voz de aliento, por nunca dejar de creer en mí y por motivarme aún cuando creía imposible llegar hasta aquí, por impulsarme cada día a ser mejor y por enseñarme el significado del verdadero amor.

A mi abuelita, Aurora Rojas, porque con cada oración, con cada café lleno de amor y cada beso me lleno de las fuerzas para culminar este desafío, por consentirme durante toda mi vida de la forma más maravillosa que existe.

A mi hermano, Wilmer Lozano, por ser mi mayor motivación y más grande orgullo, por enseñarme lo que es la verdadera disciplina, esfuerzo, constancia y dedicación.

Y a ti, mi Luz en el camino, por ser mi gran motivo para lograrlo todo en esta vida.

Agradecimientos

En primer lugar, quiero manifestar mi enorme agradecimiento a mi director Felipe Restrepo Calle, Ph.D. y a mi codirector Jhon Jairo Ramírez Echeverry, Ph.D. por la paciencia con la que aportaron a este proceso con todo su conocimiento, por cada aporte y palabras de motivación durante cada etapa de este recorrido. Agradecer por cada espacio en el que a través de su sabiduría y conocimientos instruyeron mi camino y esta investigación de manera satisfactoria. Gracias igualmente por el tiempo invertido durante cada una de las reuniones y por la disposición brindada a lo largo de este camino.

Agradecer de igual manera, al grupo de investigación PLaS, a cada profesor y cada uno de los compañeros que lo integran, por sus aportes en cada reunión y las sugerencias que permitieron mejorar esta investigación.

A la Fundación Universitaria UCompensar, por formarme, creer en mi y permitirme crecer y aprender en esta vocación de docente. Al decano de la Facultad de Ingeniería Nelson Felipe Rosas Jiménez, por su valiosa confianza y por sus palabras de apoyo. Al director Paul Alexander Díaz Montaña, por su gran apoyo, ayuda y disposición durante este trayecto y permitirme los espacios requeridos para llevar a cabo este estudio. A todos mis compañeros que hicieron parte a través de sus palabras de ayuda y aliento e hicieron posible culminar esta tesis.

A la mejor profesora del mundo, Judy Marcela Moreno Ospina, por enseñarme el verdadero valor de esta tan bonita vocación, por motivarme a empezar este reto y creer en que lo lograría, por enseñarme a confiar en mí, por cada palabra de fuerza, ánimo e inspiración que permitieron avanzar de enorme manera este documento, y por ayudarme a crecer en cada etapa de mi vida, mil y mil gracias.

Y a todos los que de una u otra manera hicieron parte de este camino y aportaron a culminar de manera satisfactoria esta tesis.

Resumen

Título: Evaluación del impacto en la motivación en el aprendizaje de los estudiantes de programación de computadores mediante el uso de una herramienta de evaluación automática

La enseñanza y aprendizaje en el contexto educativo universitario ha necesitado a través del tiempo de múltiples esfuerzos por parte de los estudiantes y docentes. Por un lado, los estudiantes están ante las dificultades propias de cualquier área de estudio: la disminución de motivación por aprender a lo largo de las asignaturas, las falencias en los conocimientos previos, los malos hábitos de estudio o la falta de estrategias para determinar buena dedicación de tiempo a las actividades académicas. Por otro lado, los docentes se enfrentan a diversas dificultades a la hora de determinar las estrategias de aprendizaje y los mecanismos apropiados para contribuir a la motivación de los estudiantes durante toda la asignatura, buscando que se cumpla con el propósito de garantizar un aprendizaje completo y significativo. En el aprendizaje de la programación de computadores los estudiantes se enfrentan a diferentes desafíos, por lo cual, el docente busca diversas estrategias y metodologías innovadoras para motivar el aprendizaje de los estudiantes. Esto implica que hoy en día se integren herramientas computacionales con el propósito de facilitar la apropiación del conocimiento en los estudiantes. Cada vez son más las estrategias y metodologías educativas que incorporan herramientas computacionales de evaluación automática de código fuente, de tipo juez online, para las asignaturas de programación de computadores ya que permiten: motivar a los estudiantes y promover su interés en las diferentes temáticas a través de la práctica y la ejercitación de las habilidades lógicas de programación, incentivar la participación y la atención en las sesiones de clase y mejorar el rendimiento académico. Se ha evidenciado a través de la literatura que integrar estas herramientas en las asignaturas de programación ha tenido efectos positivos en la mejora del rendimiento, el desarrollo de habilidades lógicas y la participación. Sin embargo, aún se evidencia la falta de investigaciones que determinen si la motivación, como factor importante del aprendizaje, se ve o no afectada por la intervención de estas herramientas en la metodología de las clases de programación de computadores.

Por tal razón, la presente investigación centra sus esfuerzos en evaluar el impacto en la motivación en el aprendizaje de los estudiantes al utilizar una herramienta de evaluación automática de código fuente en la asignatura de programación de computadores. El estudio se llevó a cabo a través de un diseño de investigación cuasiexperimental, en el cual se comparó el nivel de la motivación en el aprendizaje de los estudiantes divididos en dos grupos. Por un lado, el grupo experimental, en el cual se integró la herramienta de evaluación automática en la metodología de la clase. Por otro lado, el grupo control, en el cual se optó por seguir una metodología tradicional en este tipo de asignaturas y no se integró ninguna herramienta de evaluación automática. La comparación de los niveles de motivación se dio por medio de un Pre-Test y Post-Test con el instrumento MSQI-Colombia. Adicional a esto, en el grupo experimental, se aplicó una encuesta con respuestas abiertas, lo que permitió obtener las diferentes opiniones y percepciones sobre la herramienta de evaluación automática a su vez sobre la integración de estas estrategias en la metodología de la clase.

A través de los datos recolectados se realizó un análisis estadístico respecto a la variación de la motivación en los grupos tanto experimental como control. Por una parte, los datos cuantitativos obtenidos a través del MSLQ-Colombia no evidenciaron diferencias significativas en la motivación por aprender de los estudiantes del grupo experimental en el cual se integró la herramienta de evaluación automática. Por otra parte, los datos cualitativos permitieron contrastar el impacto hallado en el análisis cuantitativo. Los datos de la encuesta de percepción aplicada permitieron evidenciar un aumento en los niveles relacionados con las creencias sobre el control del aprendizaje. Estos resultados permiten entender de mejor manera los efectos de la integración de herramientas de evaluación automática en la motivación por aprender de los estudiantes de las asignaturas de programación de computadores.

Palabras clave: herramientas de evaluación, motivación en el aprendizaje, programación de computadores

Abstract

Title: Evaluation of the impact on the learning motivation of computer programming students using an automatic assessment tool

Teaching and learning in the university educational context has required over time multiple efforts on the part of students and teachers. On the one hand, students are faced with the difficulties typical of any area of study: the decrease in motivation to learn throughout the subjects, the lack of previous knowledge, poor study habits or the lack of strategies to determine good time dedication to academic activities. On the other hand, teachers face several difficulties when determining learning strategies and appropriate mechanisms to contribute to the motivation of students throughout the subject, seeking to fulfill the purpose of ensuring a complete and meaningful learning. In learning computer programming, students face different challenges, so the teacher seeks various strategies and innovative methodologies to motivate student learning. This implies that nowadays computational tools are integrated with the purpose of facilitating the appropriation of knowledge in students. There are more and more educational strategies and methodologies that incorporate computational tools for automatic assessment of source code, of the online judge type, for computer programming subjects since they allow: motivating students and promoting their interest in the different topics through the practice and exercise of logical programming skills, encouraging participation and attention in class sessions and improving academic performance. It has been evidenced through the literature that integrating these tools in programming subjects has had positive effects on performance improvement, development of logic skills and participation. However, there is still a lack of research to determine whether or not motivation, as an important factor in learning, is affected by the intervention of these tools in the methodology of computer programming classes.

For this reason, the present research focuses its efforts on evaluating the impact on students' learning motivation when using an automatic source code assessment tool in computer programming classes. The study was carried out through a quasi-experimental research design, in which the level of learning motivation of students divided into two groups was compared. On the one hand, the experimental group, in which the automatic assessment tool was integrated into the classroom methodology. On the other hand, the control group, which chose to follow a traditional methodology in this type of subjects and did not integrate any automatic assessment tool. The comparison of motivation levels was made by means of a Pre-Test and Post-Test with the MSLQ-Colombia instrument. In addition to this, in the experimental group, a survey with open answers was applied, which allowed to obtain the different opinions and perceptions about the automatic assessment tool as well as about the integration of these strategies in the classroom methodology.

Through the data collected, a statistical analysis was carried out regarding the variation of motivation in both experimental and control groups. On the one hand, the quantitative data obtained through the MSLQ-Colombia did not show significant differences in the motivation to learn of the students in

the experimental group in which the automatic assessment tool was integrated. On the other hand, the qualitative data allowed contrasting the impact found in the quantitative analysis. The data from the perception survey showed an increase in the levels related to beliefs about the control of learning. These results allow a better understanding of the effects of the integration of automatic assessment tools on the learning motivation of students in computer programming subjects.

Keywords: assessment tools, computer programming, learning motivation

Esta tesis de maestría se sustentó el 17 de Noviembre de 2022 a las 08:00 am, y fue evaluada por los siguientes jurados:

Paul Alexander Díaz Montaña, Mba.

Afiliación: Facultad de Ingeniería, Fundación Universitaria UCompensar.

Elizabeth León Guzmán, Ph.D.

Afiliación: Departamento de Ingeniería de Sistemas e Industrial, Universidad Nacional de Colombia, Sede Bogotá.

Contenido

Resumen	VII
Abstract	IX
Lista de Figuras	XIII
Lista de Tablas	XV
1. Introducción	1
1.1. Motivación	1
1.2. Trabajos relacionados	3
1.3. Problema de investigación	5
1.4. Objetivos	6
1.4.1. Objetivo General	6
1.4.2. Objetivos Específicos	6
1.5. Estructura de la tesis	7
2. Marco conceptual y trabajos relacionados	8
2.1. Aprendizaje en ciencias de la computación	8
2.2. Motivación en el aprendizaje	11
2.3. Metodologías de aprendizaje a través de evaluación automática de código fuente	15
2.4. Estrategias para la autorregulación del aprendizaje de la programación	18
2.5. Trabajos relacionados	20
3. Selección de la herramienta de evaluación automática	24
3.1. Antecedentes de herramientas de evaluación automática	24
3.2. Selección de la herramienta de evaluación automática	32
3.3. VPL: Virtual Programming Lab	36
4. Diseño del estudio	41
4.1. Metodología planteada	41
4.2. Participantes	47
4.3. Integración de la herramienta de evaluación automática en la asignatura	47
4.3.1. Estructura curricular de la asignatura	48

4.3.2.	Configuración de las actividades a desarrollar dentro de la herramienta de evaluación automática	50
4.3.3.	Relación entre los contenidos temáticos de la asignatura y las actividades de VPL	52
4.4.	Cronograma de actividades	53
4.5.	Variables a medir	56
4.5.1.	Motivación en el aprendizaje	56
4.5.2.	Percepciones sobre la herramienta	56
4.6.	Instrumentos de recolección de datos	57
4.6.1.	Motivated Strategies for Learning Questionnaire - Colombia (MSLQ - Colombia)	57
4.6.2.	Encuesta de percepción sobre la herramienta VPL	63
5.	Resultados experimentales	67
5.1.	Motivación en el aprendizaje	68
5.1.1.	Grupo control - Ingeniería de Telecomunicaciones	68
5.1.2.	Grupo experimental - Ingeniería de Sistemas	72
5.2.	Percepciones de los estudiantes	74
5.2.1.	Categoría: Motivación	76
5.2.2.	Categoría: Evaluación automática	84
6.	Discusión de resultados	98
7.	Conclusiones y trabajos futuros	104
7.1.	Conclusiones	104
7.2.	Publicación	105
7.3.	Trabajos futuros	106
A.	Anexo: Ítems de la escala de motivación en el aprendizaje del instrumento MSLQ-Colombia	107
	Bibliografía	110

Lista de Figuras

- 2-1. Modelo de innovación docente propuesto por Álvarez *et al* (2008) 10

- 3-1. Componentes de VPL 37
- 3-2. Actividad del curso “Algoritmos y Programación” en VPL 38
- 3-3. Configuración de casos de prueba en VPL 38
- 3-4. Lista de entregas de una actividad del curso “Algoritmos y Programación” en VPL . . . 39
- 3-5. Similaridad de código fuente de una actividad del curso “Algoritmos y Programación”
identificado por VPL 39

- 4-1. Metodología propuesta de diseño del estudio 42
- 4-2. Opciones de la escala de Likert del cuestionario MSLQ-Colombia 61
- 4-3. Opciones de la escala de Likert de la encuesta de percepción sobre VPL 65

- 5-1. Respuestas obtenidas con el MSLQ-Colombia en el Pre-Test del grupo control del
Programa de Ingeniería de Telecomunicaciones 69
- 5-2. Respuestas obtenidas con el MSLQ-Colombia en el Pos-Test del grupo control del
Programa de Ingeniería de Telecomunicaciones 70
- 5-3. Respuestas obtenidas con el MSLQ-Colombia en el Pre-Test del grupo experimental
del Programa de Ingeniería de Sistemas 72
- 5-4. Respuestas obtenidas con el MSLQ-Colombia en el Pos-Test del grupo experimental
del Programa de Ingeniería de Sistemas 73
- 5-5. Resultado del análisis cualitativo a partir de la encuesta de percepción - Categorías
Motivación y Evaluación automática 76
- 5-6. Respuestas obtenidas con la Escala de Likert en la encuesta de percepción de VPL en
el grupo experimental de Ingeniería de Sistemas 77
- 5-7. Resultado del proceso de análisis cualitativo. Categoría “Motivación” 78
- 5-8. Resultado del proceso de análisis cualitativo. Categoría “Evaluación automática” 85
- 5-9. Respuestas obtenidas sobre las características de VPL con mayor aceptación 97

Lista de Tablas

2-2.	Continuación de Constructos de la motivación en el aprendizaje	15
2-3.	Comparación de las metodologías y técnicas para la enseñanza de la programación a partir de herramientas de evaluación. Adaptado de Kanika <i>et al</i> (2020)	17
2-4.	Categorías de autorregulación del aprendizaje de la programación adaptadas de Garcia <i>et al</i> (2018)	19
2-5.	Herramientas de evaluación automática en cursos de programación	20
3-2.	Criterios de selección de la herramienta	34
4-1.	Diseño de la primera de sesión de clase del experimento	43
4-2.	Diseño de sesiones de clase durante el experimento	44
4-5.	Continuación de cronograma de actividades	55
4-6.	Continuación de Sub-escalas de motivación en el aprendizaje del instrumento MSLQ-Colombia	60
4-7.	Continuación de Encuesta de Percepción sobre la herramienta VPL	64
5-1.	Número de estudiantes que participaron, aprobaron las actividades, el número de entregas por actividad y el promedio de entregas por estudiante en cada actividad para el curso Programación de Computadores	67
5-2.	Resultados estadísticos del Pre-test y Post-test del grupo control - Ingeniería de Telecomunicaciones	71
5-3.	Resultados estadísticos del Pre-test y Post-test del grupo experimental - Ingeniería de Sistemas	74
5-4.	Conceptos en la categoría "Motivación" definidos a partir de las preguntas abiertas de los en la encuesta de percepción	79
5-5.	Continuación de Conceptos e indicadores en la categoría "Evaluación automática" definidos a partir de las preguntas abiertas de los en la encuesta de percepción	88
5-5.	Continuación de Conceptos e indicadores en la categoría "Evaluación automática" definidos a partir de las preguntas abiertas de los en la encuesta de percepción	89
5-5.	Continuación de Conceptos e indicadores en la categoría "Evaluación automática" definidos a partir de las preguntas abiertas de los en la encuesta de percepción	90
A-1.	Sub-escala motivacional: Valoración de la Tarea	107
A-2.	Sub-escala motivacional: Orientación hacia metas intrínsecas	107

A-3.	Sub-escala motivacional: Orientación hacia metas extrínsecas	108
A-4.	Sub-escala motivacional: Creencias sobre control del aprendizaje	108
A-5.	Sub-escala motivacional: Expectativas de auto-eficacia en el rendimiento	108
A-6.	Sub-escala motivacional: Expectativas de auto-eficacia en el aprendizaje	109
A-7.	Sub-escala motivacional: Ansiedad en los procesos de evaluación	109

1. Introducción

1.1. Motivación

En el campo de la ingeniería, principalmente en los programas académicos relacionados a las ciencias de la computación, una de las competencias fundamentales a desarrollar es la programación de computadores y/o algoritmos (Alves *et al.*, 2020). Las asignaturas que trabajan dichas competencias por lo general reciben el mismo nombre. En ellas las habilidades lógicas y racionales juegan un factor indispensable a la hora de adquirir destrezas para resolver problemas en cualquier área de estudio, incluso en otras carreras del campo de la ingeniería tales como: electrónica, civil, mecatrónica, industrial, eléctrica, telecomunicaciones, telemática, entre otras; convirtiéndose en la base para asignaturas de la misma línea en semestres posteriores (Muñoz *et al.*, 2012; Sun *et al.*, 2020).

En estas asignaturas los estudiantes adquieren competencias para: desarrollar programas simples que cumplan con tareas previas, comprendiendo sus funcionalidades y características; definir funciones simples comprendiendo sus parámetros y argumentos; realizar pruebas y depuraciones a programas; elegir las iteraciones y las construcciones condicionales apropiadas para alguna tarea específica (ACM y IEEE., 2009; ACM y IEEE., 2013; ACM y IEEE., 2017). Las mismas competencias que se demandan en el personal a contratar en el campo laboral, y que según varios reportes, representan unas de las habilidades más necesarias en los últimos años. Por tanto, se hace imperativo que los profesionales las apropien (Elliott., 2017).

Lo anterior representa un reto para las instituciones educativas respecto a las mejoras en sus procesos pedagógicos, los cuales deben darse teniendo en cuenta dos factores: por una parte, el estudiante debe desarrollar la capacidad de resolver una problemática dada a partir del uso de diferentes conceptos y estrategias propias de la materia; por otro lado, la intervención que realiza el docente es vital para lograr el éxito que se espera, lo que demanda que aborde diferentes metodologías y estrategias pedagógicas, las cuales permitan facilitar dicho proceso (Bennedsen y Caspersen., 2019; Byrne y Lyons., 2001). Sin embargo, este reto no es fácil de abordar, normalmente surgen una gran cantidad de problemáticas tanto en el proceso de aprendizaje como en el proceso de la enseñanza.

Por un lado, el estudiante puede presentar una serie de inconvenientes que dificultan su comprensión en este campo. Problemáticas tales como: la falta de motivación, pocas habilidades de pensamiento lógico y computacional, dificultad para la resolución de problemas, bajos niveles de pensamiento matemático harán que este proceso sea más complejo para algunos estudiantes (Jiménez-Toledo *et al.*,

2019; Khramova *et al.*, 2019).

Por otro lado, el docente, como rol valioso en la consolidación de este conocimiento, puede presentar diversos inconvenientes a la hora de la enseñanza. La falta de actualización de conocimientos, las didácticas inapropiadas, pedagogía basada en situaciones demostrativas, uso de recursos desactualizados del mercado o la falta de retroalimentación pueden conllevar desde unas bases insuficientes en habilidades de programación hasta malas prácticas a la hora de desarrollar programas (Chibizova., 2018).

Por ende, se debe contar con metodologías y estrategias pedagógicas innovadoras que faciliten este proceso de aprendizaje (Bosse y Gerosa., 2017). Las herramientas computacionales y tecnológicas son una gran opción, puesto que, adoptando formas novedosas de aprendizaje, fomentan la motivación y atención de cada estudiante estos cursos (González Jaimes *et al.*, 2018; Tavares *et al.*, 2017).

Existen diferentes estrategias pedagógicas basadas en herramientas computacionales para el aprendizaje/enseñanza de programación de computadores (Darejeh y Salim., 2016; Guerrero *et al.*, 2015). Entre estas estrategias se destacan las siguientes:

- **Ambientes de aprendizaje:** Los ambientes de aprendizaje en el contexto de la educación en programación de computadores y el uso de herramientas tecnológicas son aquellos espacios virtuales en donde se da la interacción del docente y los estudiantes con el propósito de la enseñanza y el aprendizaje (Paredes-Daza y Sanabria-Becerra., 2015). Esta interacción se puede dar a través de diferentes elementos de los ambientes de aprendizaje tales como: videos, tutoriales, imágenes, documentos, foros, recursos didácticos, entre otros.
- **Ambientes gamificados:** Los ambientes gamificados son espacios tecnológicos apoyados por plataformas web, aplicaciones de escritorio o ambientes de aprendizaje que integran elementos, dinámicas, interacción y mecánicas de los juegos, tales como: medallas, recompensas, tableros de puntuación, narrativa, temporalización, progreso, trabajo colaborativo, competitividad y metas u objetivos puntuales (Escamilla *et al.*, 2016). Esto con el fin de incrementar la motivación y participación de los estudiantes además de facilitar el aprendizaje de la programación permitiendo captar la atención de los estudiantes a través del reto o desafío generando así una experiencia emocionantes por medio de ésta.
- **Ambientes/Herramientas de evaluación automática:** Los ambientes de evaluación automática son herramientas que mediante entornos web, aplicaciones instalables en IDEs, aplicaciones de escritorio o ambientes de aprendizaje permiten, por un lado, automatizar el proceso de calificación de código fuente, evaluando la solución enviada frente a unos casos de prueba planteados por el docente. Por otro lado, permiten que los estudiantes puedan practicar a partir de múltiples ejercicios, con la posibilidad de ejecutar su solución conociendo si cumple con lo esperado, y recibiendo a través de la herramienta una retroalimentación rápida de su solución (Guerrero *et al.*, 2015).

Teniendo en cuenta que uno de los factores más importantes en la programación corresponde a la rápida y completa retroalimentación de los ejercicios, las herramientas de evaluación automática resultan particularmente importantes (Verdú *et al.*, 2012). Estas herramientas sirven como apoyo para el aprendizaje autorregulado de los estudiantes. Este aprendizaje hace referencia al proceso que conduce al estudiante a controlar su propia comprensión, comportamiento y motivación con el objetivo de alcanzar las metas de aprendizaje, lo cual resulta esencial cuando se intentan aprender estas habilidades de programación (Ramírez-Echeverry *et al.*, 2018). Sin embargo, las herramientas de evaluación automática difieren en muchos aspectos. Los lenguajes de programación que soporta, el ambiente de ejecución, el propósito y tipo de evaluación, la aplicación en diversos ámbitos, la ejecución de código fuente, el proceso de anti-plagio, la comparación de código pueden ser algunos de estos aspectos. Por otra parte, a través de diversos estudios se ha validado la mejora en los procesos de enseñanza de la programación mediante este tipo de herramientas, identificando factores claves para la comprensión de las habilidades en los estudiantes. Algunos de los factores de mejora con el uso de las herramientas de evaluación automática son los siguientes: ayudan a comprender los problemas antes de implementar soluciones, facilitan la evaluación formativa mediante una oportuna retroalimentación y aportan a la corrección semántica y sintáctica de las soluciones brindadas por los estudiantes (Gordillo., 2019; Restrepo-Calle *et al.*, 2019).

Las estrategias pedagógicas que integran herramientas tecnológicas en los procesos de aprendizaje en muchos casos buscan motivar al estudiante en su proceso de formación. En el contexto del aprendizaje autorregulado se habla de la motivación en el aprendizaje como un proceso de construcción psicológica que realiza un estudiante individual y que toma la forma de las interpretaciones y valoraciones situacionales (Järvelä y Niemivirta., 2001). Esta motivación está compuesta por características como: el interés de realizar un entrega, las expectativas de éxito del proceso de aprendizaje, las expectativas respecto al proceso de evaluación, la ansiedad ante la evaluación, la valoración para la tarea, entre otras (Ramírez-Echeverry *et al.*, 2018).

Teniendo en cuenta todo lo anterior, las herramientas de evaluación automática de tareas de programación podrían ser un posible factor motivacional en el proceso de aprendizaje de los estudiantes de programación de computadores.

1.2. Trabajos relacionados

Uno de los problemas fundamentales a la hora de enfrentar el proceso de enseñanza y aprendizaje de la programación son las formas en las que los estudiantes perciben su proceso de aprendizaje y las distintas formas que tienen los docentes de evaluar el conocimiento y las habilidades adquiridas por ellos. Teniendo en cuenta esto, las herramientas de evaluación automática se integran para poder apoyar este proceso con el fin de motivar al estudiante y facilitar la forma en la que adquiere habilidades de la programación. Según Gupta y Gupta (2018), el propósito de estas herramientas en el

aprendizaje es fortalecer las habilidades de los estudiantes prestando especial atención a los estudiantes novatos o principiantes. Estas herramientas han sido altamente utilizadas en los campos de la programación por sus resultados en la motivación del estudiante por aprender y en la manera en que la retroalimentación, como una de las características esenciales de estas herramientas, aporta al estudiante a conocer sus errores en las soluciones propuestas (Ullah *et al.*, 2018). Sin embargo, también se han empezado a estudiar los efectos de estas herramientas en otros cursos tales como: sistemas inteligentes, inteligencia artificial y aprendizaje de máquina (González-Carrillo *et al.*, 2021).

Al tener presente que la programación de computadores requiere de un desarrollo mental lógico para su comprensión, es importante que estas herramientas permitan: la medición de superación a nivel individual, la solución de problemas de complejidad incremental, la búsqueda de soluciones tanto individual como en equipo, el desarrollo apropiado de modelos mentales y la asignación de tareas, trabajos o entregas frecuentes y cortas (Vennila *et al.*, 2004). Teniendo en cuenta que las primeras etapas del aprendizaje en la programación implican una carga cognitiva bastante alta, más aún cuando no se ha tenido una experiencia previa con la programación, es importante que todo aprendizaje dentro de estas herramientas permita tener la capacidad de manejar la complejidad de los ejercicios de forma gradual al permitir una personalización y creación de los mismos (Sweller., 1988).

Al integrar estas herramientas en el ámbito académico se puede obtener una mejora significativa en los procesos de enseñanza y aprendizaje, puesto que aumentan la motivación de los estudiantes para completar correctamente las actividades gracias a la capacidad que brindan estas herramientas en la retroalimentación formativa y la posibilidad de reenviar las soluciones entendiendo que quizás no se logran solucionar completamente en el primer intento (Ullah *et al.*, 2018). A través de múltiples investigaciones se ha evidenciado el aporte que pueden lograr estas herramientas en el proceso formativo de la programación. En el estudio de Ullah *et al* (2018) se profundiza en los efectos que tiene la integración de herramientas de evaluación automática en los estudiantes de programación principiantes o de primeros semestres en el contexto universitario. En esta investigación se estudian los diversos tipos de herramientas de evaluación automática indicando que según el tipo de enfoque del análisis que brinde la herramienta (estático, dinámico o híbrido) puede conllevar diversas acciones a realizar por parte de los docentes y de los estudiantes. En el estudio realizado por Cardoso, Marques *et al* (2020) se puede identificar unas mejoras significativas en el proceso de aprendizaje de los estudiantes al integrar herramienta de evaluación automática dentro del curso de algoritmos. En este se determinó un grupo experimental en el cual, a partir de actividades, se apoyaba la enseñanza de la temática de programación y, mediante una encuesta, obtuvieron la percepción de los estudiantes. En esta encuesta encontraron que los estudiantes perciben el uso de la herramienta como una gran ayuda para la resolución de ejercicios de programación, para apoyo al proceso de aprendizaje y para el fortalecimiento de la enseñanza.

Por otro lado, el estudio de realizado por Restrepo-Calle *et al* (2019) permite identificar los factores importantes para los estudiantes en cuanto a la integración de estas herramientas. En el estudio se

indica que la retroalimentación de la herramienta incorporada ayudó a guiar a los estudiantes en su proceso de aprendizaje en el contexto de la programación, por lo cual la herramienta incorporada facilitó la evaluación formativa puesto que brindaba a los estudiantes una retroalimentación antes incluso de obtener una calificación cuantitativa, es decir, una nota. Igualmente, se indica que tras el uso de la herramienta la retroalimentación obtenida por los estudiantes es más completa que si tan solo se utiliza un compilador de código fuente, incluyendo además una corrección sintáctica de las soluciones ingresadas por los estudiantes o una corrección semántica.

Finalmente, en el estudio realizado por Rubio-Sánchez *et al* (2014) se analiza la percepción de los estudiantes al usar una herramienta de evaluación automática para actividades de programación. En este estudio se indica que para obtener un mayor aporte al proceso de aprendizaje la herramienta que se integre debe permitir brindar una buena retroalimentación que permite a los estudiantes estar al tanto de los errores en su proceso formativo. Sin embargo, no se conoce con claridad el impacto que generó en la motivación de aprender de los estudiantes la integración de la herramienta de evaluación. De igual manera, no se tiene claridad del impacto al integrar herramientas con detección de plagio, planteado como un trabajo futuro el poder determinar el efecto de dicha integración y si aquello ayuda a despertar la motivación y el interés de los estudiantes por aprender.

1.3. Problema de investigación

Las herramientas de evaluación automática que permiten evaluación formativa y/o sumativa, resultan ser un factor motivacional importante en las habilidades de programación de los estudiantes novatos (Ullah *et al.*, 2018). Hoy en día las estrategias pedagógicas basadas en estas herramientas computacionales tienen una gran acogida en los procesos de enseñanza y aprendizaje. Resultan tener un impacto significativo en la motivación de los estudiantes gracias a diferentes mecanismos, tales como: la retroalimentación durante el proceso de evaluación, la experiencia de concursos y competición, la disponibilidad de prácticas en todo momento, la práctica frecuente, la verificación de errores detectados en el proceso. Todo ello genera un gran apoyo al proceso para la comprensión y adquisición de habilidades lógicas en el campo de la programación de computadores (Ion *et al.*, 2019).

Muchos investigadores en el campo de la educación en ingeniería han integrado estas herramientas de evaluación automática en cursos de programación. Sin embargo, su enfoque se ha centrado en la experiencia de estas implementaciones desde el punto de vista tecnológico (Brito y Goncalves., 2019; Croft y England., 2019; Shivam *et al.*, 2019).

En investigaciones anteriormente citadas, se evidencian claras mejoras en cuanto a la motivación de los estudiantes y su mejora en los procesos de aprendizaje sobre la programación de computadores a través de la evaluación y retroalimentación propias de estas herramientas (Janičić y Marić., 2020;

Krugel *et al.*, 2020; Sun *et al.*, 2019). Sin embargo, muchas de estas integraciones han centrado sus esfuerzos en verificar el efecto de la herramienta desde el enfoque de la enseñanza y la mejora en el proceso docente más que en el proceso de aprendizaje por parte de los estudiantes (Gomes y Mendes., 2015; Gordillo., 2019; Wunsche *et al.*, 2019). Además, la mayoría de este tipo de estudios carece de diseños experimentales que permitan validar los resultados y el impacto de estas herramientas en la motivación de los estudiantes. Por esto, varios autores sugieren realizar la implementación e integración de estas herramientas en un ambiente académico para la validación del efecto generado en la percepción, motivación y el rendimiento de los estudiantes de programación (Benotti *et al.*, 2018; Restrepo-Calle *et al.*, 2018).

Por ende, no se puede determinar, en relación a las investigaciones analizadas, si los resultados aportados evidencian mejoras en la motivación del estudiante más allá de los efectos propios de estas integraciones. Si estos resultados y discusiones pudieran corroborarse experimentalmente, aportarían para entender de mejor manera el impacto que pueden llegar a generar estas herramientas en la motivación de los estudiantes de primeros semestres en el área de la programación de computadores. En esta investigación se desea comprender el impacto que genera la integración de una herramienta de evaluación automática en la motivación de los estudiantes, a través de métodos mixtos de investigación que permitan validar los resultados. Teniendo en cuenta que este aprendizaje, guiado a partir de evaluación y retroalimentación, permite mejorar considerablemente los aspectos motivacionales y las destrezas en programación programación, tiene sentido profundizar en este tipo de estudio y análisis (Gupta y Gupta., 2018).

En este contexto, en esta tesis de maestría se pretende contestar la siguiente pregunta de investigación:

- ¿Cuál es el impacto en la motivación en el aprendizaje de los estudiantes a través de la integración de una herramienta de evaluación automática en un curso de programación de computadores?

1.4. Objetivos

1.4.1. Objetivo General

Evaluar el impacto en la motivación en el aprendizaje de los estudiantes de programación de computadores mediante la integración de una herramienta de evaluación automática.

1.4.2. Objetivos Específicos

1. Seleccionar una herramienta de evaluación automática que apoye el proceso de aprendizaje de la programación de computadores mediante una revisión de literatura.

2. Integrar la herramienta de evaluación automática seleccionada para apoyar el proceso de aprendizaje en un curso de programación de computadores.
3. Diseñar un experimento para evaluar el impacto de la herramienta de evaluación automática en la motivación en el aprendizaje de los estudiantes en un curso de programación.
4. Caracterizar el impacto en la motivación en el aprendizaje de los estudiantes generado por medio de la integración de una herramienta de evaluación automática para el aprendizaje de programación de computadores.

1.5. Estructura de la tesis

Esta tesis se encuentra estructurada de la siguiente manera:

- En el Capítulo 1 se puede encontrar toda la introducción al trabajo investigativo realizado, teniendo en cuenta la motivación, trabajos relacionados, planteamiento del problema y los objetivos planteados para este estudio.
- En el Capítulo 2 se presenta el marco conceptual y los trabajos relacionados en detalle, teniendo en cuenta conceptos e investigaciones relacionadas con el uso de herramientas de evaluación automática en el aprendizaje de la programación de computadores.
- En el Capítulo 3 se realiza la selección de la herramienta de evaluación automática para el apoyo en el proceso de aprendizaje de la programación, buscando identificar diferentes herramientas y realizando un análisis comparativo que determine la selección de dicha herramienta.
- En el Capítulo 4 se presenta el diseño experimental de manera detallada que se llevó a cabo durante la realización de esta investigación. Describiendo la integración y configuración de la herramienta de evaluación automática en un curso de programación de computadores y evidenciando el plan acción para el experimento y la recolección de datos posteriores a este.
- En el Capítulo 5 se evidencian los resultados de los análisis cuantitativos y cualitativos del experimento.
- En el Capítulo 6 se realiza la discusión de los resultados experimentales.
- En el Capítulo 7 se dan a conocer las conclusiones, publicaciones realizadas en el proceso de investigación y los posibles trabajos futuros.

2. Marco conceptual y trabajos relacionados

En este capítulo se profundizará en cada una de las temáticas fundamentales y necesarias para el desarrollo del presente trabajo de investigación, en el marco de la motivación en el aprendizaje. En el campo de la educación en ingeniería se ven diversas variantes que alteran la forma tradicional del aprendizaje, teniendo en cuenta procesos lógicos, formación sistemática de conceptos, pensamiento algorítmico asociado a la resolución de problemas y demás formas racionales de abordar las diferentes situaciones del aprendizaje. Este aprendizaje resulta ser significativo si el estudiante se involucra en su proceso y encuentra interés en las temáticas planteadas. Es por esto que se ve la necesidad de despejar la temática del aprendizaje desde el campo de las ciencias de la computación y cómo este aprendizaje tiene algunos aspectos que lo diferencian de otras áreas de estudio. A partir de allí se busca comprender el paradigma de aprendizaje significativo enmarcado en el campo propiamente de la ingeniería, en el cual el proceso de aprendizaje de la programación conlleva una importancia relevante y una base sólida para su proceso formativo y está orientado desde la motivación en el aprendizaje y la evaluación del aprendizaje. Posteriormente se abordan la motivación en el aprendizaje con el fin de analizar los factores que permiten al estudiante mejorar su proceso de formación, teniendo en cuenta mejoras en las estrategias pedagógicas para este fin. Después se analizan las metodologías del aprendizaje a partir de la evaluación del aprendizaje con herramientas de evaluación automática. Lo que conlleva a comprender como la autorregulación en el aprendizaje aprovecha las herramientas de evaluación automática para fortalecer las habilidades de la programación. Finalmente, se abordan trabajos relacionados con la integración de herramientas de evaluación automática en clases de programación.

2.1. Aprendizaje en ciencias de la computación

El aprendizaje se contempla como el proceso mediante el cual el estudiante o participante del proceso formativo adquiere diversos conocimientos, destrezas y habilidades (Pérez Pino *et al.*, 2017). Este proceso integra variadas técnicas, teorías, modelos, metodologías y estilos para permitir que el aprendizaje sea significativo y aplicable al campo de acción propio de cada participante. Si bien, el aprendizaje a través de las últimas décadas ha empezado a depender en mayor medida de la integración de tecnologías que permitan apoyar la comprensión y evaluación de dichos conocimientos adquiridos, no en todos los campos puede llevarse a cabo bajo los mismos parámetros, ni se puede contar con las mismas estrategias tecnológicas para el proceso de formación.

En el área de las ciencias de la computación, los procesos cognitivos que deben realizar los estudiantes se enfocan en la lógica, la racionalización y la sistematización como factores fundamentales para desarrollar las habilidades requeridas en este campo (R. Contreras *et al.*, 2020). Proceso mismo que requiere que el participante no solo capte y memorice la información si no también realice otros procesos cognitivos tales como: comprender, aplicar, analizar y sintetizar (Monereo., 2014). Esto implica que el estudiante participe de manera activa en este proceso de formación y comprenda que ser autónomo le permite conocer su propio proceso cognitivo y controlar así su propio aprendizaje (Barros Barrios *et al.*, 2008). Por lo cual es necesario que el estudiante desarrolle un conjunto de procesos cognitivos tales como: selección, organización, estructuración de la información, integración de conocimientos previos y aplicación en situaciones reales (Dalfaro *et al.*, 2008). Es por esto que en este campo, el docente o profesor tiene un rol fundamental que va más allá de transmitir información. El docente tiene que participar activamente en la formación, aportando en el proceso de construcción de conceptos, y tomando el rol de mediador entre el estudiante y su camino al conocimiento (Del Valle., 2004).

A lo largo de los años se han investigado distintas estrategias para aportar a los estudiantes de ingeniería conocimientos sólidos a través de un proceso de aprendizaje acorde a las necesidades de los estudiantes, a la realidad de la industria y a las diferentes herramientas tecnológicas que participan en el proceso como mediación tecnológica. En el campo de las ciencias de la computación uno de los paradigmas educativos para concebir el aprendizaje y que ha empezado a generar cambios en procesos metodológicos es el aprendizaje significativo (Álvarez *et al.*, 2008). Para Álvarez *et al.* (2008), este paradigma permite concebir el aprendizaje como todo un proceso de construcción de significado, por lo tanto, el estudiante no se limita a obtener el conocimiento, sino que lo construye. De igual manera, busca que el estudiante sea mucho más activo en su proceso formativo con el fin de que éste sea más autónomo, autorregulado, conozca sus propios procesos cognitivos y desarrolle un buen control del aprendizaje. Según Moreira (2017), este es un proceso activo, en el cual el estudiante no se puede limitar únicamente a aprender mecánicamente o de memoria, sino que debe realizar un serie de actividades con el fin de comprenderlos y asimilarlos de forma significativa en sus estructuras cognitivas de forma organizada. De acuerdo con Beltrán (1993), para obtener un aprendizaje significativo el estudiante no se puede limitar a repetir actividades o materiales de apoyo informático, pues ese aprendizaje será simplemente basado en la repetición; por otra parte, si el estudiante organiza y elabora la información, además de controlar el aprendizaje a partir de la comprensión y la calidad del tiempo de estudio, éste será un aprendizaje significativo.

Álvarez *et al.* (2008) plantea un modelo que permite centrar los esfuerzos en obtener un aprendizaje significativo a partir de variables como la motivación para aprender por parte de los estudiantes y la evaluación en el aprendizaje como se observa en la Figura 2-1. Posee, como proceso intermediación, los medios que se requieren para que la evaluación permita generar interés y motivación en el proceso de aprendizaje.

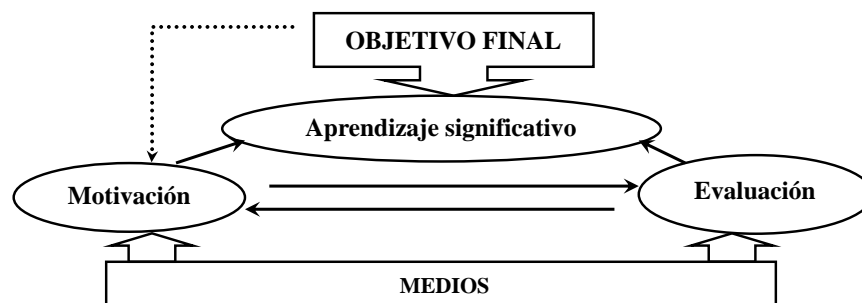


Figura 2-1.: Modelo de innovación docente propuesto por Álvarez et al (2008)

Para el proceso de motivación en el aprendizaje, dentro del modelo planteado por Álvarez et al (2008), se plantea la necesidad de brindar apoyo en el proceso a partir de actividades que permitan captar la atención de los estudiantes por aprender la temática propuesta por el docente. En el estudio se indica la necesidad de motivación, puesto que al encontrar estudiantes que carecen de una adecuada motivación conllevará a que el estudiante tarde más tiempo en ponerse a realizar actividades, tendrá menos concentración, estudiará la temática con menor frecuencia y de forma más general, y se rendirá ante las primeras dificultades encontradas (Tapia., 2001). Álvarez et al (2008) propone para la motivación por aprender lo siguiente: tratar de plantear los temas a través de una determinada situación, sorprendente, novedosa y basada en prácticas aplicadas en el entorno real, relacionar los temas nuevos con los contenidos anteriores y proporcionar espacios propicios para la solución de problemas y apoyo en retroalimentación por parte del docente.

Por otra parte, esta motivación se debe integrar y relacionar con un proceso de evaluación del aprendizaje propicio para analizar los conocimientos y habilidades adquiridas por el estudiante sin disminuir el interés o sentir algún tipo de frustración en su aprendizaje. Álvarez et al (2008) plantea que hay algunas diferencias entre lo que es evaluación en el aprendizaje, indicando que para algunos docentes se trata de realizar mediciones y para otros se consolida como la comprobación de objetivos y resultados de aprendizaje. Sin embargo, E. Contreras (2004) propone la siguiente definición de evaluación: “evaluar es un proceso que implica obtener una buena información respecto a los dominios de los alumnos (objetivos, conocimientos, aptitudes, habilidades, comportamientos, etc), establecer juicios de valor (aceptable, adecuado, bien, suficiente, etc) y tomar decisiones (admitir, aprobar, recomendar, promocionar, liberar, convalidar, etc)”.

Álvarez et al (2008) indica que en el proceso evaluativo se deben identificar tres fases con objetivos diferenciados en cada una de estas. En la primera fase se determinará cuál es el tipo de información necesaria por lo cual es importante saber qué se quiere evaluar. En la segunda fase se recopila la información por lo cual es importante conocer las técnicas de recopilación con los instrumentos adecuados y aplicarlos. Finalmente, en la última fase se procede a evaluar la información y tomar decisiones a

partir de esto para posteriormente divulgar con los participantes los resultados obtenidos del proceso.

Se pueden diferenciar los tipos de evaluación partiendo de distintos criterios. Por lo cual es normal encontrar literatura que hable de: evaluación formal o informal, evaluación de resultados, procesos o autorregulación, autoevaluación y coevaluación, evaluación con referencia a normas o criterios o evaluación diagnóstica, formativa y sumativa (Álvarez *et al.*, 2008). Es en esta última tipología de evaluación a la que más énfasis se ha brindado en el contexto de ciencias de la computación. En la Tabla 2-1 se pueden evidenciar las diferencias entre los tres tipos de evaluación: diagnóstica, sumativa y formativa.

Tabla 2-1.: Características de la evaluación diagnóstica, formativa y sumativa

Evaluación diagnóstica	Evaluación sumativa	Evaluación formativa
Proceso de identificación de estilos y motivaciones de aprendizaje y/o conocimientos previos de los estudiantes con el fin de plantear estrategias de mejora (Grover., 2021)	Permite determinar si los estudiantes han alcanzado o no y hasta qué punto los objetivos educativos propuestos mediante una calificación/nota (Gallardo., 2021)	Permite brindar información al estudiante sobre cómo desarrolla, mejora, y/o corrige sus actividades académicas (Gallardo., 2021)
<ul style="list-style-type: none"> ● Recoge información sobre los conocimientos previos ● Permite adaptar la programación de la asignatura ● Fuente de información para los alumnos ● Aprender a aprender ● Anticipar el contenido de la asignatura ● Estimular mediante una implicación más activa 	<ul style="list-style-type: none"> ● Se usa al final de una fase educativa ● Decisiones respecto a la nota final impuesta al alumno ● Sistema tradicional ● Función educativa: recoge información sobre el progreso y nivel de aprendizaje 	<ul style="list-style-type: none"> ● Durante todo el proceso de aprendizaje ● Ofrece realimentación al alumno (<i>feedback</i>) ● Es difícil de realizar: se confunde con la sumativa e implica un cambio en la metodología

Fuente: Adaptado de Álvarez *et al* (2008).

En cuanto a la recopilación de la información sobre el aprendizaje, se pueden disponer de múltiples instrumentos. Sin embargo, la selección de uno u otro dependerá de los objetivos planteados. Estos instrumentos pueden ser pruebas objetivas, pruebas abiertas, pruebas orales, observación, evaluación de trabajos y otros métodos. En la evaluación de trabajos y otros métodos es donde se encuentran instrumentos más acordes a las ciencias de la computación con instrumentos tales como: proyectos, utilización de software informático, escalas de autoevaluación y autoinformes (Álvarez *et al.*, 2008).

2.2. Motivación en el aprendizaje

A través del tiempo la investigación en educación informática ha estado orientada a comprender los requerimientos cognitivos en los estudiantes y las necesidades pedagógicas de los docentes. Según Lishinski y Yadav (2019), los factores cognitivos y pedagógicos más investigados han sido principalmente la resolución de problemas en el contexto de la educación informática, el pensamiento crítico y la capacidad matemática, encontrando así que algunos de estos factores están relacionados con resultados de aprendizaje en los estudiantes de computación. Sin embargo, factores de tipo no cognitivo, tales como la motivación y las actitudes, no han llamado la misma atención de los investigadores en

educación. Si bien, estos factores no están tan asociados a los contenidos de cursos de informática o computación, son importantes para el éxito en el aprendizaje de los estudiantes y los resultados propios de su formación.

Lishinski y Yadav (2019) indican que la motivación es uno de los factores no cognitivos más importantes para lograr el éxito del proceso de aprendizaje de los estudiantes de áreas relacionadas con la computación y la ingeniería. En la investigación se asocia la motivación como un componente psicológico que conlleva a las personas a tener un mayor compromiso y persistan en su comportamiento respecto a la meta que se planteen.

Según Loui y Borrego (2019), la motivación es un componente de la psicología que se encarga de estudiar y explicar las razones por las que las personas inician y mantienen comportamientos frente a alguna actividad o meta. En el ámbito de la educación en ingeniería la motivación puede explicar los comportamientos de los estudiantes respecto a las actividades académicas y su persistencia a lo largo del tiempo frente al programa de formación estudiado. Lishinski y Yadav (2019) de igual manera determinan la motivación como un conjunto de modelos teóricos relacionados pero distintos que se conforma de diferentes constructos tales como: autoeficacia, la orientación a metas y la autorregulación.

Para Wardani *et al* (2020) la motivación es aquel impulso que permite dirigir u organizar su comportamiento. En cuanto a la motivación en el aprendizaje lo percibe como una fuerza que impulsa a los estudiantes a generar, garantizar y proporcionar dirección respecto a las actividades de aprendizaje con el fin de lograr los objetivos esperados. Entre mayor sea la motivación que tenga el estudiante, mayor serán los logros y los resultados obtenidos. En esta investigación se indican elementos que pueden influir en la motivación para el aprendizaje, tales como: las aspiraciones, la capacidad del estudiante para aprender, las condiciones de aprendizaje y el entorno de aprendizaje. En este estudio se identifican la motivación en el aprendizaje como motivación intrínseca y extrínseca.

En el estudio de Aristika *et al* (2021) se indica que la motivación en el aprendizaje es aquello que provoca el aprender en los estudiantes y garantiza la continuidad de las actividades de aprendizaje planteadas según determinados objetivos. Se denota como aquello que conlleva al estudiante a obtener mejores resultados y experiencias generando un interés del estudiante por aprender. En este estudio se plantean cinco dimensiones de la motivación en el aprendizaje de los estudiantes: la persistencia en el aprendizaje, la resiliencia ante la adversidad, el interés y la agudeza de atención en el aprendizaje, la excelencia en el aprendizaje y la independencia en el aprendizaje.

Para Ramírez-Echeverry (2017) la motivación es aquel componente que lleva al estudiante a un buen desempeño académico, en el cual el docente está apoyando este proceso de autorregulación. Sin embargo, es fundamental que el estudiante tenga presente su nivel de motivación, manejando su interés, su eficacia y su ansiedad. En este contexto académico, el estudiante se debe plantear metas claras

acordes a la percepción de si mismo y las capacidades con las que cuenta para lograr las metas planteadas en el proceso de aprendizaje.

Según Ibrahim y Nat (2019), se asocia a la motivación con las experiencias y consecuencias específicas, por lo cual los aspectos extrínsecos resultan ser una fuerte influencia en los estudiantes. En el ámbito de la enseñanza y el aprendizaje en temáticas orientadas a la tecnología, la motivación resulta ser un buen mediador activo que permite a los docentes integrar estrategias para lograr incrementar el interés en los estudiantes. Por otra parte, en el estudio se considera que la motivación se establece como la interacción de un estudiante con una situación en particular. Este proceso de motivación mantiene el esfuerzo para lograr una meta planteada y se determina que el proceso funciona de manera efectiva cuando las necesidades personales no están en conflicto con los objetivos previstos. En este estudio se propone como factores intrínsecos, el aprendizaje, actitudes y virtudes del docente y por factores extrínsecos: la interacción con la tecnología, la sobrecarga académica, el ambiente institucional y la interacción con los estudiantes.

Azmi *et al* (2018) indican que en el proceso de programación las percepciones de los estudiantes suelen ser negativas porque lo consideran difícil, lo que conlleva a que los estudiantes que han tenido experiencias previas con la programación encuentren mayor motivación en el proceso de aprendizaje y los estudiantes principiantes encuentren menor motivación ya que encontrarán algunas temáticas complejas para entenderlas. En este estudio se plantea que para el aprendizaje de la programación los estudiantes con poca motivación preferirán optar por el enfoque de prueba y error, en vez de mejorar las habilidades para la resolución de problemas. En muchos otros casos los estudiantes con baja motivación tenderán a darse por vencidos más fácilmente y evitarán la programación de computadores en sus planes futuros.

El término “motivación”, tal como lo expresan Lishinski y Yadav (2019), cubre múltiples teorías y modelos que están directamente relacionados con los determinantes afectivos de los comportamientos o conductas, incluyendo características como la metas, los deseos y las creencias en el estudiante.

En el estudio de Ramírez-Echeverry *et al* (2016) a partir de Pintrich *et al* (1991) se conciben como constructos de la motivación en el aprendizaje: la ansiedad ante los exámenes, el valor de la tarea, la autoeficacia orientada hacia el aprendizaje, la autoeficacia orientada hacia el rendimiento, las creencias sobre el control del aprendizaje, la motivación extrínseca y la motivación intrínseca. En la Tabla 2-2 se pueden observar las definiciones de cada una de estos.

Tabla 2-2.: Constructos de la motivación en el aprendizaje

Constructos de la motivación en el aprendizaje
<p>Ansiedad ante los exámenes</p> <p>Para Seijo Galán <i>et al</i> (2020) la ansiedad ante los exámenes se explica como la tendencia o predisposición de algún estudiante a reaccionar con una alta ansiedad en situaciones de evaluación de sus aptitudes o conocimientos, asociado con el rendimiento académico y las consecuencias para su autoestima.</p> <p>Esta puede ser anticipatoria si se evidencia al momento de estudio o antes del proceso evaluativo, incluso en el momento de realización de la evaluación. Para este último caso se puede llegar a producir una ansiedad tan elevada que podría provocar un bloqueo en su capacidad para demostrar y evidenciar sus conocimientos en el examen.</p>
<p>Valor de la tarea</p> <p>La definición de este constructo, según Azmi <i>et al</i> (2018), es el grado de interés personal que muestra un estudiante por una actividad, tarea o trabajo específico, teniendo en cuenta la relevancia, utilidad e importancia que este mismo le otorga a su proceso formativo.</p> <p>Para Pintrich <i>et al</i> (1991) el valor de la tarea es el nivel de importancia, utilidad o interés que el estudiante le atribuye a una determinada actividad.</p>
<p>Autoeficacia</p> <p>Para Loui y Borrego (2019) la autoeficacia es la creencia de un individuo en su capacidad para culminar o terminar una determinada tarea, trabajo o actividad con el fin de lograr una meta deseada.</p> <p>Según Lishinski y Yadav (2019) se define la autoeficacia como la confianza de que un estudiante puede completar con éxito un determinado resultado. Esto implica el esfuerzo que esta dispuesto a invertir con el fin de cumplir con las metas, objetivos o retos, y las habilidades que se adquieran para superar las adversidades en el proceso.</p> <p>Azmi <i>et al</i> (2018) determina la autoeficacia como la creencia de los estudiantes sobre el control de su aprendizaje y la tasa de éxito que tendrán al realizar determinada tarea. Law <i>et al</i> (2010) indica que la autoeficacia está relacionada con el rendimiento académico y el control de aprendizaje ante determinada tarea. Por un lado, las expectativas de autoeficacia en el rendimiento se relacionan con la capacidad y confianza que tiene el estudiante frente al factor de éxito en el proceso evaluativo de alguna materia en el contexto académico. En cuanto a las expectativas de autoeficacia en el aprendizaje se asocian con la valoración que hace un alumno acerca de su capacidad de comprender los conceptos o temas planteados en la asignatura o de adquirir las habilidades y competencias deseadas.</p>

Tabla 2-2.: Continuación de Constructos de la motivación en el aprendizaje

Constructos de la motivación en el aprendizaje
Creencias sobre el control del aprendizaje
El constructo de creencias sobre el control del aprendizaje está relacionado con la certeza que tiene el estudiante en que el proceso de aprendizaje dependerá de su propio esfuerzo y no de aspectos externos tales como el ambiente de aprendizaje, la estructura curricular, la dificultad de las actividades en la asignatura o el profesor (Pintrich <i>et al.</i> , 1991).
Motivación extrínseca
Según Wardani <i>et al</i> (2020) la motivación extrínseca en el aprendizaje es el deseo por realizar alguna actividad a partir de un estímulo externo. Para Wardani <i>et al</i> (2020) no se cuenta como un verdadero sentimiento o deseo de aprender por parte de los estudiantes, puesto que los alumnos realizan actividades para lograr objetivos o metas fuera de las actividades de aprendizaje. En Pintrich <i>et al</i> (1991) se define la motivación extrínseca como los factores externos por los que el estudiante desea aprender, sean becas, premios, reconocimiento, entre otros.
Motivación intrínseca
Para Wardani <i>et al</i> (2020) la motivación intrínseca es el deseo de actuar a partir de un factor interno, es decir, no existe ningún factor externo que conlleve al estudiante a aprender. Los estudiantes que están motivados intrínsecamente desean participar en actividades de aprendizaje porque sienten la necesidad de cumplir alguna meta y quieren lograr algún objetivo en su proceso formativo. Los estudiantes que están motivados intrínsecamente en las actividades de formación aprenden mejor que los estudiantes que están extrínsecamente motivados.

Fuente: Elaboración propia

2.3. Metodologías de aprendizaje a través de evaluación automática de código fuente

Las herramientas de evaluación automática cada vez son más usadas en el proceso de enseñanza y aprendizaje de la programación puesto que permiten potencializar las habilidades lógicas de los estudiantes mediante ejercicios, práctica constante y retos. Las metodologías de aprendizaje a partir de herramientas de evaluación automática apoyan la construcción de una experiencia realista para el estudiante, integrándolo en ambientes competitivos; a su vez que permite la práctica autodidacta, lo que genera en el estudiante ser partícipe y apropiarse de su propio proceso de aprendizaje. Estas

herramientas permiten que con la retroalimentación los estudiantes puedan descubrir sus principales dificultades de aprendizaje y, al mismo tiempo, comprendan cuáles son los errores que no permiten solucionar los diversos problemas de programación planteados. Para Ullah *et al* (2018), el apoyo en clases de programación de las herramientas de evaluación automática permite mejorar las habilidades y destrezas de programación de los estudiantes, a la vez que apoyan la motivación por aprender y permiten el apoyo durante el proceso de enseñanza mediante la retroalimentación inmediata de estas herramientas.

En el estudio realizado por Keuning *et al* (2016) se indica que la retroalimentación inmediata de las herramientas de evaluación automática ayudan en el proceso de aprendizaje de los estudiantes, puesto que permite señalar errores en las soluciones enviadas, además de enfatizar en la importancia de las pruebas para el proceso formativo de la programación. Sousa Silva (2022) indica en su estudio que las ventajas de integrar herramientas de evaluación automática son que permite aumentar el interés de los estudiantes por la programación y fortalecer las habilidades y el rendimiento general. Skalka *et al* (2019) indica que estas herramientas de evaluación automática ayudan a los estudiantes en su proceso de aprendizaje, pues permiten trabajar o practicar en cualquier momento y lugar, teniendo a partir de la retroalimentación inmediata un camino por el cual seguir en caso de encontrarse errores a sus soluciones. Según Verdú *et al* (2012), la integración de estas herramientas en la clase de programación aumentó la satisfacción de los estudiantes comparado con los que no hicieron uso de esa, determinando así que el nivel de satisfacción no depende de sus niveles de habilidades informáticas.

Teniendo presentes los diferentes desafíos y retos a lo largo de los cursos de programación en el proceso de enseñanza y aprendizaje, se han planteado distintas soluciones que soportan el proceso de formación de los estudiantes. Una de las soluciones que han abordado distintos autores es la integración de componentes de la evaluación automática, tales como: la ejecución de código fuente, la evaluación a partir de casos de prueba, la retroalimentación inmediata, la calificación rápida, la posibilidad de evaluar en distintos lenguajes de programación y la detección de plagio en las actividades académicas. Estos son algunos de los componentes que permiten aportar al incremento en procesos exitosos de aprendizaje de los estudiantes de programación.

Dentro de las soluciones basadas en componentes de las herramientas de evaluación automática, se pueden identificar: sistemas de programación visual, aprendizaje basado en juegos, programación en parejas y colaborativa, programación de robots y sistemas de evaluación automática.

En el estudio realizado por Kanika *et al* (2020) se presentan técnicas para apoyar las metodologías de aprendizaje en programación de computadores a partir de la integración de herramientas tecnológicas. Como se observa en la Tabla 2-3 se evidencian estas técnicas junto con las ventajas y los desafíos planteados por los autores. Por una parte, la programación visual permite integrar en el proceso de aprendizaje de la programación a los estudiantes principiantes pues pueden evidenciar el proceso de forma clara; sin embargo, este requiere tiempo para configurarse e implementarse; al igual que los

Tabla 2-3.: Comparación de las metodologías y técnicas para la enseñanza de la programación a partir de herramientas de evaluación. Adaptado de Kanika *et al* (2020)

Técnica	Ventajas	Desafíos
Programación visual	Se puede usar de manera efectiva para presentar a los estudiantes con antecedentes no informáticos a la programación.	El desarrollo de un entorno de desarrollo integrado requiere mucho tiempo y esfuerzo.
Aprendizaje basado en juegos	Se puede personalizar según los requisitos de estudiantes individuales.	El desarrollo del entorno de programación requiere mucho tiempo y esfuerzo.
Programación en parejas y colaborativa	Los estudiantes obtienen comentarios detallados y oportunos que les permiten escribir código de mayor calidad.	Encontrar la composición adecuada para una pareja o un equipo es un desafío.
Programación de robots	Se puede utilizar con bastante eficacia para motivar a los estudiantes a aprender a programar.	La disponibilidad de robots es una limitación.
Sistemas de evaluación automática	Aumenta la escalabilidad del curso y la posibilidad de ejecución de código fuente con retroalimentación.	Es difícil evaluar automáticamente ciertos aspectos de la programación.

ambientes basados en juego, que de igual manera requieren una planificación acordada dentro de las clases para poder obtener los resultados deseados. La programación por pares por su parte permite integrar a los estudiantes en un aprendizaje colaborativo, el cual permite que se comparta el conocimiento y se ayuden a fortalecer habilidades apoyado por otro estudiante; sin embargo, esto juega un doble factor y es encontrar el equilibrio de las parejas a fin de que ambos aporten lo necesario para la construcción de las soluciones por partes iguales. La programación con robots generalmente resulta ser motivante para los estudiantes pues se puede evidenciar de forma física los resultados de la aplicación de código fuente; sin embargo, la implementación es más compleja en los lenguajes de programación de hardware, los ambientes tienen que ser controlados y la limitante de cantidad de dispositivos en grupos grandes también juega un factor clave en la decisión de integrar esta técnica dentro de las clases de programación.

Finalmente, Kanika *et al* (2020) expresa que los sistemas de evaluación automáticos son los que se usan en los cursos de programación puntuales en la investigación, teniendo presente que con mayor cantidad de estudiantes se hace difícil el proceso de verificación y comentarios por parte de los docentes. Si bien las herramientas de evaluación automática tienden a ocupar un gran esfuerzo en la configuración inicial de las actividades, posterior a esto permiten apoyar los procesos de aprendizaje contando con componentes como la retroalimentación, calificación automática, posibilidad de reenvíos en las soluciones y soporte de sintaxis para los estudiantes.

2.4. Estrategias para la autorregulación del aprendizaje de la programación

La autorregulación en el aprendizaje es el proceso, dentro de la teoría sociocognitiva, que desarrolla el estudiante con el fin de controlar su propia cognición, su conducta y la motivación con el ánimo de alcanzar unos objetivos propios de aprendizaje (Ramírez-Echeverry *et al.*, 2018). Zimmerman (1989) expresa que la autorregulación es un proceso autodirigido con el fin de que los estudiantes conviertan sus habilidades mentales en las habilidades necesarias para el aprendizaje. En el área de la educación en ingeniería es fundamental conocer cómo desarrollar programas informáticos (desde el diseño hasta la implementación de algoritmos), la cual es una competencia indispensable. Otras competencias claras en este proceso son analizar como funcionan los programas, conocer lenguajes de programación, desarrollar todas las fases del software o elegir la estructura de datos más apropiada para la problemática. Cada una de estas actividades requiere de enormes esfuerzos cognitivos, tales como la resolución de problemas utilizando lo visto en clase o estudiar y practicar extraclase. En este sentido, Ramírez-Echeverry *et al* (2018) indican que para aprender sobre programación, el estudiante tiene que comprender temáticas complejas que requieren esfuerzos cognitivos, estas pueden ser actividades en el momento de la clase y continuar a partir del trabajo extracurricular o prácticas extraclases.

El aprendizaje autorregulado es interpretado a través de distintos modelos. De igual manera, existen diferentes enfoques sobre lo que compone y de cómo interactuará el proceso de aprendizaje autorregulado. El marco general para el aprendizaje autorregulado indica que los estudiantes pueden regular su cognición, motivación, afecto, conducta y contexto mientras llevan a cabo su proceso de aprendizaje. García *et al* (2018) indica que las categorías de autorregulación en el aprendizaje son: autoevaluación, organización y transformación, establecimiento de objetivos y planeación, búsqueda de información, almacenamiento de registros y monitoreo, entorno físico, auto consecuencias, ensayo y memorización, búsqueda de asistencia social, revisión de registros y otros. En la Tabla 2-4 se pueden observar las definiciones de las categorías de aprendizaje autorregulado, así como la estrategia de aprendizaje a utilizar en cada categoría según García *et al* (2018) a partir de Zimmerman (1989).

Es por esto por lo que al integrar este tipo de modelo pedagógico en las clases es fundamental contar con herramientas que permitan al estudiante llevar a cabo de manera autónoma su proceso de aprendizaje a su vez que se aumenta la motivación. En la categoría de “Autoevaluación” se encontrarían las herramientas de evaluación automática que a través de casos de prueba permiten al estudiante comprobar sus soluciones con unos casos planteados por el docente. Sin embargo, la categoría “Otros”, en la cual el comportamiento de aprendizaje esta impulsado por factores externos, podrían contener estas herramientas de evaluación, aunque la categoría está más relacionada al refuerzo del aprendizaje por parte del estudiante de manera complementaria.

Tabla 2-4.: Categorías de autorregulación del aprendizaje de la programación adaptadas de Garcia *et al* (2018)

Categoría	Definición	Estrategias relacionadas
Autoevaluación	Evaluaciones iniciadas por los estudiantes sobre la calidad o el progreso de su trabajo validando los ejercicios de programación	Evaluar el programa a partir de casos de prueba
Organización y transformación	Construcción de planes de diseño antes de la codificación de ejercicios de programación	Escribir algoritmos mediante pseudocódigo y diagramas de flujo de datos (DFD)
Establecimiento de objetivos y planeación	El alumno determina los objetivos de programación y aparta un tiempo para desarrollar los ejercicios de programación para una práctica, taller o examen final	Programar los tiempos de estudio
Búsqueda de información	El estudiante se esfuerza por encontrar información propicia para su tarea, para comprender mejor los temas de programación, por ejemplo, través del uso de bases de conocimientos en línea	Realizar consulta en plataformas de foros como Stackoverflow o buscadores de información como Google
Almacenamiento de registros y monitoreo	El alumno conserva la información sobre el material de clase para su posterior revisión	Organizar la información y los materiales de apoyo a partir de etiquetas
Entorno físico	El estudiante indaga cuál herramienta seleccionar con la que se le facilite el proceso de aprendizaje	Utilizar herramientas como IDEs (Entornos de Desarrollo Integrados) que permiten al estudiante ajustar el entorno digital o herramientas virtuales
Autoconsecuencias	Recompensas o castigos imaginarios otorgados por el éxito o el fracaso durante el proceso de aprendizaje	Uso de herramientas que promueven la competencia para obtener premios o castigos imaginarios.
Ensayo y memorización	El estudiante se esfuerza por memorizar los aspectos temáticos de la programación a partir de la práctica	Apoyarse de contenido o recursos audiovisuales para practicar la programación
Búsqueda de asistencia social (búsqueda de ayuda)	Esfuerzos del estudiante para solicitar respuestas a preguntas de compañeros, profesores y otras personas a través de bases de conocimientos sociales	Utilizar foros en línea para encontrar respuestas o soluciones
Revisión de registros	Esfuerzos del estudiante para releer apuntes, registros o logs de programación, exámenes o materiales de aprendizaje para prepararse para la clase o exámenes adicionales	Revisar materiales de estudio como libros, blogs, foros en línea y otros sitios web.
Otros	Comportamiento de aprendizaje impulsado por personas influyentes externas, como maestros, padres o agentes digitales	Apoyo a través de: juegos, herramientas de aprendizaje, material interactivo, tutorías externas, vídeos, entre otros.

Es por esto, que se tiene una amplia relación entre la autorregulación en el aprendizaje y la integración de herramientas de evaluación automática en las metodologías de clase, puesto que estas permiten que el estudiante regule su proceso de cognición a su vez que encuentra interés en las temáticas mediante la práctica y el desarrollo de habilidades propias de la programación por medio de la ejercitación autónoma.

2.5. Trabajos relacionados

La importancia de las herramientas de evaluación automática en el proceso de aprendizaje radica en que permiten: corrección, ejecución, compilación de código fuente, verificación de funcionalidad, validación de test antes o después del envío, análisis de comparación entre fuentes, análisis de calificación, soporte de diversos lenguajes de programación. Además, cumpliendo con características tales como velocidad, disponibilidad, consistencia y objetividad en la evaluación (Ala-Mutka., 2005; Christian y Trivedi., 2016; Ihantola *et al.*, 2010).

Cardoso, Marques *et al* (2020) realizan, a través de un amplio análisis, una revisión de algunas herramientas de evaluación automática en cursos de programación de computadores. La Tabla 2-5 muestra dicho análisis.

Tabla 2-5.: Herramientas de evaluación automática en cursos de programación

Herramientas	Lenguajes soportados	Tipo	Enfoque
BOCA	Varios	Web	Competición/Enseñanza
BOSS/BOOS2	C, Pascal y Java	Web	Enseñanza
Codeboard	Varios	Plugin Moodle	Enseñanza/Aprendizaje
CodeRunner	Python, C, C++, Java, PHP, JavaScript Octave y MATLAB	Plugin Moodle	Enseñanza/Aprendizaje
Marmoset	Varios	Único	Enseñanza
Mooshak	Varios	Web	Competición/Enseñanza
PC2	Java, C++, Python y otros	Web	Competición
PETCHA	Los soportados por Eclipse y Visual Studio	Web	Enseñanza
Pythia	Python y otros	Web	Enseñanza
SPOJ	Varios (about 45)	Web	Competición
UVa Edujudge	Pascal, C, C++ y Java	Moodle	Enseñanza
VPL	Java, C, C++, C#, Pascal, Ruby, Python y otros	Plugin Moodle	Enseñanza/Aprendizaje
Web-CAT	Java, C++ y Pascal	Web	Enseñanza

Fuente: Adaptado de Cardoso, Marques *et al* (2020)

Las herramientas mencionadas en la Tabla 2-5 manejan diferentes propósitos y características. La mayoría están enfocadas principalmente en la enseñanza. Sin embargo, difieren entre sí por: el lenguaje de programación que soportan, su forma de evaluar y por su enfoque de aplicación. En esta última característica se encuentran: competencias de programación, procesos educativos de enseñanza/aprendizaje, o selección y reclutamiento de programadores. Existen algunas otras herramientas como lo son: Assyst, AutoLEP, Automata, Bottlenose, CAP, CourseMarker, Code Submission Evaluation System, eGrader, GAME, Qualified, RoboLIFT, RoboProf, SAC o YAP3+APAC, que difieren de igual manera en las características previamente mencionadas, pero por su antigüedad o propósitos no enfocados propiamente en la educación en ciencias de la computación no se abordaron con profundidad en la revisión realizada. De dicha revisión se puede inferir que existe una amplia gama de herramientas y estudios que abordan el uso de éstas como apoyo a la educación en programación de computadores.

En particular, las herramientas de evaluación automática han representado en los últimos años una notable mejoría en los procesos de enseñanza/aprendizaje, y aún más, cuando éstas permiten su integración en diferentes ambientes de aprendizaje (De Oliveira Brandão *et al.*, 2016). La integración de estas herramientas en cursos de primeros semestres en la universidad genera una experiencia positiva en los estudiantes y fomenta su motivación dentro del campo de la programación. Tal como lo afirman Croft y England (2019), basándose en los resultados obtenidos a través de la integración de la herramienta CodeRunner en un curso de programación. En este proceso se evidenció una mejora en cuanto al manejo de estructuras de control y condicionales, además de generar un cierto grado de satisfacción en los estudiantes motivo de la sana competencia en los cuestionarios virtuales realizados a través de la herramienta. Para Becerra-Alonso *et al* (2020), al integrar una herramienta de este tipo se pueden identificar las ventajas de la retroalimentación oportuna y la importancia que tiene para el proceso de aprendizaje de los estudiantes. A través de esta integración con la herramienta EduZinc se complementa de manera óptima la enseñanza desde la vista del instructor y ese trabajo autónomo realizado por el estudiante en sus tiempos de dedicación a la práctica.

Gallego-Romero *et al* (2020) evidencian una mejora en el aprendizaje de los estudiantes a partir de la práctica y la ejercitación de habilidades de programación en herramientas de evaluación automática. A través de CodeBoard se evidencia una respuesta positiva en los estudiantes a las herramientas que permiten compilar y ejecutar código fuente antes de enviarlo a calificación, generando así una serie de pruebas antes de ser evaluado por el sistema. Brito y Goncalves (2019), a través de la herramienta CodeFlex describen lo importante que es la práctica en estas asignaturas y adicionan, como aspecto positivo dentro de los resultados, que se mejoran los procesos de aprendizaje cuando se emula un entorno más cercano a la industria y se estimula la sana competencia mediante estas herramientas.

Shao *et al* (2019) integran su herramienta Programming Assigment Automatic Assessing (PAAA) en un curso de primer semestre. Esta herramienta permite generar evaluaciones a modo de competencia y verificar a través de un juez las diferentes comprobaciones del código fuente. Si bien, el resultado evidencia una mejora sobre el manejo de conceptos, condiciones, ciclos y demás competencias en el

desarrollo de un programa, también se puede verificar un cierto grado de dificultad de alguna de estas herramientas en estudiantes con bajo o ningún conocimiento previo de programación. Esto puede conllevar a una desmotivación en el estudiante, al experimentar un cierto grado de frustración comprendiendo que serán los primeros pasos en este campo para muchos de ellos.

Para Benotti *et al* (2018), estas herramientas facilitan considerablemente los procesos de aprendizaje, teniendo como factor importante la retroalimentación propia de estas tecnologías. Se evidencian mejoras a través de indicadores de evaluación mediante el modelo de aceptación tecnológica TAM (Technological Acceptance Model) en cuanto a habilidades adquiridas, gestión de tiempo, velocidad de realización de tareas, soluciones óptimas a tareas más complejas, y comprensión de conceptos de programas. Sin embargo, no se evalúa el efecto que estas herramientas tienen en la motivación en los estudiantes, teniendo en cuenta que gran parte de los últimos estudios buscan mejorar el proceso de enseñanza.

Gordillo (2019), de igual forma, describe una integración en la cual, a partir de herramientas de evaluación automática, centra su resultado en analizar los procesos de enseñanza y la ventajas que producen estos elementos tecnológicos en el desarrollo de la docencia y la innovación. Por otra parte, Restrepo-Calle *et al* (2018) desarrollaron la herramienta denominada UNCode, resultado de investigación de la Universidad Nacional de Colombia, la cual aporta a los procesos del aprendizaje en programación de computadores y la evaluación automática de tareas de programación, con soporte en varios lenguajes. Aporta al proceso de aprendizaje, teniendo en cuenta una evaluación y retroalimentación sumativa y formativa dentro del entorno de aprendizaje propio. Dentro de sus recomendaciones se encuentra validar el impacto que tenga cada uno de los módulos en un ambiente académico como sistema de apoyo para superar las dificultades asociadas al aprendizaje de programación de computadores.

Sun *et al* (2019) desarrollaron OOCourse, un sistema para la evaluación automática formativa en línea basado en un modelo de revisión de código de pares, el cual no solo permite la óptima calificación de ejercicios de programación, sino que apoya el problema de plagio a través de una detección por comparación. Su enfoque está orientado a la mejora de la experiencia del docente en gran medida, teniendo en un menor porcentaje la motivación de esta herramienta en los estudiantes.

Queirós y Leal (2012), a través de la herramienta PETCHA, describen el proceso de enseñanza en el cual, a través de ejercicios realizados por el docente, se ayuda a generar el proceso de aprendizaje de los estudiantes, teniendo en cuenta la ejecución y análisis de código fuente realizado. De igual forma, el estudio está orientado a la práctica y mejora de pedagogía docente, más que en la validación óptima del impacto motivacional en los estudiantes que la usen.

En el estudio realizado por Cardoso, Marques *et al* (2020) se evidencia esa mejora a través de la práctica realizada con la herramienta Virtual Programming Lab (VPL) Rodríguez-del-Pino (2012), a través de la cual se puede ejecutar, probar, compilar y evaluar programas completos, simulando en gran

manera las ventajas de un entorno de desarrollo integrado (en inglés, IDE - Integrated Development Environment) antes de realizar las entregas. Éstas brindarán oportunamente la retroalimentación del ejercicio, mediante casos de prueba. Los resultados de la integración de este entorno basado en Moodle permiten comprender el apoyo que puede ser en el proceso de aprendizaje efectivo y autónomo en los estudiantes, mejorando así mismo sus habilidades de programación, la capacidad de resolución de problemas y el pensamiento lógico mediante el uso constante de esta herramienta. Por otro lado, esta herramienta mejora considerablemente el tiempo y esfuerzo de los docentes en la evaluación de actividades y ejercicios, permitiendo analizar no solo el resultado, sino también el proceso. Evita errores humanos de juicio y subjetividad; además, como complemento, permite verificar la existencia de plagio entre las entregas.

De acuerdo a lo expuesto anteriormente, surge la necesidad de realizar una caracterización de la motivación en el aprendizaje de los estudiantes, de los programas académicos de ingeniería que inician su proceso de formación con la asignatura de programación de computadores como materia transversal. Esto implica la integración de una herramienta de evaluación automática en un curso de programación, y así generar un espacio único de concentración y foco de los estudiantes durante el proceso de ejercitación y codificación. Merece la pena aclarar que no sólo se trata de poner a disposición dicha herramienta, sino evaluar su impacto en el aprendizaje, determinando si estas estrategias son bien recibidas por los estudiantes de acuerdo a su motivación. Esto amerita una medición con técnicas estadísticas para validar su impacto dentro de un curso de programación. Por lo tanto, es importante presentar no solo una estrategia, sino la evaluación de su relevancia para los estudiantes.

Resulta indispensable la buena preparación de los futuros ingenieros para abordar los retos del sector empresarial, lo cual parte de la motivación que tengan para aprender al interior de sus asignaturas, dado que, si se cuenta con bases sólidas en programación y algoritmos, al estudiar nuevos lenguajes de programación, estructuras de datos y arquitecturas de software, se les podría facilitar su comprensión y uso.

3. Selección de la herramienta de evaluación automática

Inicialmente se recopiló información mediante una revisión de literatura, con el propósito de identificar algunas de las herramientas existentes para la evaluación de código fuente, que permiten apoyar a través de metodologías activas los procesos de enseñanza y aprendizaje de las asignaturas relacionadas con la programación de computadores. Posteriormente, se describen cada una de las herramientas identificadas, teniendo en cuenta: principales características, lenguajes de programación soportados, tipo de herramienta, tipo de evaluación, ejecución y retroalimentación, enfoque, especialidad y población objetivo. Con base en esta información recopilada se construyó un cuadro comparativo que permite analizar la información y determinar a partir de diferentes criterios cuál herramienta se adapta a los requerimientos y objetivos puntuales de este estudio. Los criterios considerados incluyeron: enfoque en la enseñanza, elementos de herramientas de evaluación de código fuente, lenguajes soportados, entre otros. A partir de esto, se realizó la selección de la herramienta de evaluación automática de código fuente.

3.1. Antecedentes de herramientas de evaluación automática

A través de distintas herramientas computacionales se han integrado y desarrollado los componentes que requiere un juez en línea para su uso dentro del contexto de la programación de computadores con el fin de mejorar la práctica del aprendizaje en este campo de acción. A continuación, se presentan algunas de las herramientas de evaluación automática de código fuente que apoyan el proceso de la enseñanza y el aprendizaje en la programación.

Codeboard: es un ambiente de programación basado en web creado el 2014 por Christian Estler y Martin Nordio, miembros de ETH Zurich's Chair of Software Engineering (Codeboard., 2020). Este permite administrar tareas de programación, ejecutar código fuente, agrupar cursos para la calificación por parte del docente y así apoyar las clases de programación. Esta se puede utilizar en tres escenarios específicos: Mediante la integración dentro de plataformas MOOC, lo que permite integrar a diversos LMS como contenido, mediante la creación de un usuario con el fin de poder practicar en la herramienta y mediante el desarrollo de ejercicios de programación asignados por parte de un

docente que haya creado un salón en la plataforma. Cuenta con soporte para ejecutar y validar automáticamente 11 lenguajes de programación, tales como: Java, C++, PHP o Python.

La herramienta ha tenido gran acogida por su versatilidad a la hora de ejecutar e integrar este ambiente dentro de distintas plataformas de aprendizaje. En el estudio realizado por Chi *et al* (2016) se indica que la plataforma es un sistema de evaluación de código fuente enfocado más en el instructor con la posibilidad que le permite a la hora de evaluar y revisar los diversos programas de los estudiantes. De igual manera, indica que para los cursos de introducción a la programación es bastante recomendable puesto que permite integrar al estudiante a esta nueva temática sin que los conceptos puedan saturar o desfavorecer su proceso de aprendizaje.

CodeRunner: es un plugin o complemento gratuito para el LMS Moodle de tipo pregunta con el cual los docentes podrán realizar o establecer preguntas y la respuesta por parte del estudiante corresponderá a código fuente (Lobb y Harlow., 2016). Los estudiantes podrán desarrollar el código en el entorno de desarrollo brindado por este a modo de quiz y podrán enviarlo cuando lo consideran correcto. Este cuestionario o pregunta se comparará con casos de prueba que se determinan por el docente y evalúa cada uno de los envíos que se realizaron. Así mismo, dependiendo de lo configurado por el docente, este permitirá mostrar una retroalimentación automática siempre que la configuración de la actividad así lo permita.

La herramienta cuenta con la posibilidad de poder ejecutar el código en varios lenguajes de programación actuales y de alto nivel y se integra con el libro de calificaciones del ambiente de aprendizaje Moodle. En el estudio realizado por Lobb y Harlow (2016), la percepción de los estudiantes fue positiva, mencionando la importancia que tiene la retroalimentación instantánea en el proceso de aprendizaje, permitiendo así mantener la motivación mediante una comprensión constante de las temáticas que se abordan. Igualmente, es necesario tener presente que esta es solo una herramienta para los procesos evaluativos. Por otro lado, Sangwin (2019) indica que los estudiantes aprecian esta herramienta puesto que les permite practicar en entornos similares a los procesos evaluativos y aprenden a partir de la retroalimentación que les brindan los ejercicios incorrectos.

DOMJudge Online: La herramienta DOMJudge es una ambiente de evaluación automático utilizado principalmente en maratones de programación o con énfasis en competencias de desarrollo a nivel académico y profesional (DomJudge-Online., 2022). Permite crear ejercicios en los cuales los diferentes estudiantes ya sea de forma personal o por equipos los resuelven, teniendo presente la descripción del problema y ponderando el cumplimiento de los casos de prueba que se tengan a partir de una puntuación base. A través del uso de esta herramienta es posible evaluar tareas asignadas en las competencias de forma continua. A nivel académico permite guiar a los estudiantes en el proceso de aprendizaje de la programación mediante evaluación formativa, y a su vez apoyar a los docentes a obtener información sobre lo que han aprendido los alumnos mediante evaluación sumativa. En el

estudio realizado por Restrepo-Calle *et al* (2019) se hace uso de la herramienta de evaluación automática con el fin de realizar un seguimiento oportuno al progreso del aprendizaje del estudiante y no solo al resultado final; al igual que permite apoyar a los estudiantes con la práctica de ejercicios y obtener continuamente retroalimentación sobre la eficacia y el rendimiento de sus programas. Esta herramienta de evaluación automática dentro del contexto educativo y a partir de los aspectos de competitividad provee una excelente manera de motivar al estudiante, impulsando su interés por aprender (Restrepo-Calle *et al.*, 2019).

Para Pham y Nguyen (2019), DOMJudge cuenta con muchos de los elementos de las herramientas de evaluación automática de tipo juez online más importantes. La detección de plagio a nivel académico permite a los docentes tomar medidas respecto a la detección de copia en el código fuente de los estudiantes, por lo cual, este tipo de información y reportes son bien aceptados por los investigadores y docentes, pues apoya no solo la compleja tarea de revisión puntual de cada código, sino determinar cuáles cuentan con similitud a nivel, sintáctico, semántico y de forma.

JAssess: es un acrónimo para herramienta de evaluación de Java (Java Assessment Tool) y permite la compilación y ejecución de código fuente Java únicamente (Yusof *et al.*, 2012). Desarrollada por dos universidades de Malasia; por un lado, la Universidad de Tecnología, y por otro, la Universidad Tun Hussien Onn. La función principal de la herramienta es mostrar una interfaz para poder evaluar código fuente de java mediante la lectura de un **.zip**. JAssess se plantea como un plugin o complemento del LMS Moodle. Para el proceso es importante que el estudiante haya realizado el código antes de enviar a evaluación. Uno de los inconvenientes que presenta es que las métricas para la calificación solo dependerán de la compilación y el proceso de evaluación no es totalmente automático. En el estudio realizado por Yusof *et al* (2012) se puede evidenciar la mejora notable que conllevan las herramientas de evaluación automática sobre el flujo de trabajo de los docentes y el tedioso proceso de calificación en asignaturas de programación. También, se respalda la importancia que tiene la retroalimentación inmediata sobre los estudiantes, aunque en este caso la retroalimentación esta dada a partir del proceso de compilación.

JavaBrat: es una herramienta de evaluación automática integrada como módulo o complemento dentro del LMS Moodle que permite la enseñanza y el aprendizaje de los lenguajes de programación Java y Scala (Patil., 2010). Es una herramienta construida a partir de la tesis de maestría de Patil (2010) en la Universidad Estatal de San José. Permite a los estudiantes: editar el código fuente desde el navegador, se pueden agregar actividades, tareas o problemas a resolver y los estudiantes pueden ejecutar pruebas para revisar la funcionalidad de los programas y su calificación automática. En este estudio se identifican características o componentes elementales de las herramientas de evaluación automática, tales como: ejecución y pruebas de actividades de programación, compilación, formato de código (en este caso sin corrector de sintaxis) y evaluación inmediata con retroalimentación automática a partir de los casos de prueba previamente definidos a la hora de construir el ejercicio.

Marmoset: es una herramienta de evaluación de código fuente desarrollada en la Universidad de Maryland por Jaime W. Spacco a través de su tesis doctoral (J. W. Spacco., 2006). Esta herramienta permite la realización de pruebas de código fuente a partir de los proyectos de desarrollo en los siguientes lenguajes: Java, C, Ruby y Objective Caml en la investigación original (J. Spacco, Hovemeyer *et al.*, 2006). Sin embargo, según la página web oficial de la herramienta, actualmente soporta muchos más lenguajes de programación. Las características puntuales que tiene este entorno de evaluación son: permite realizar distintos casos de prueba para la comparación con el código fuente (pruebas del estudiante, pruebas públicas, pruebas de lanzamiento, y pruebas secretas), permite la integración con Eclipse IDE, permite la recopilación de datos con el fin de permitir analítica de los mismos y comprender aspectos del progreso de los estudiantes y a partir del rol instructor permite obtener datos de las entregas y reportes estadísticos. A través de múltiples investigaciones, la herramienta ha obtenido buenos resultados en la integración de diversos campos involucrados con la programación, detallando una mejora en el aprendizaje de los estudiantes a partir de la integración de retroalimentación y evaluación instantánea en sus actividades prácticas (J. Spacco *et al.*, 2005; J. Spacco, Winters *et al.*, 2006).

Mooshak: es un sistema web de tipo juez online que apoya la calificación de código fuente en asignaturas de ciencias de la computación. Desarrollado en el 2001 con el objetivo de ser un sistema de administración de concursos de programación para las competencias ICPC (International Collegiate Programming Contest), sin embargo, evolucionó al punto de brindar apoyo a otros tipos de competencias en la programación. Y aunque es utilizado con el fin competitivo, eventualmente se hace un uso pedagógico de este en niveles de programación de secundaria y universidad (Leal y Silva., 2003). Aunque la característica principal de Mooshak es la evaluación de programas computacionales, también se admiten lenguajes de diagramas estudiados en disciplinas de la informática y la computación. Diagramas tales como: autómatas finitos deterministas (DFA), modelo entidad-relación (E-R), lenguaje de modelado unificado (UML). Para estos diagramas la herramienta se basa en dos componentes: un editor de diagramas integrado y un comparador de gráficos (Correia *et al.*, 2017).

A través de varias investigaciones se han demostrado los buenos resultados al integrar herramientas de tipo juez online con aspectos de competición en las clases de programación. Los estudiantes consideran que es una experiencia enriquecedora participar en este tipo de competencias y que, adicional, permite mejorar su entusiasmo y motivación por la programación, a su vez de que aumenta el interés por mejorar en este campo (Cardoso, de Castro *et al.*, 2020; Rubio-Sánchez *et al.*, 2012). De igual manera, indican que es positivo el hecho de integrar retroalimentación automática a la hora de resolver ejercicios pues los impulsan a comprender en dónde está su error y así apoyar su propio proceso de aprendizaje.

PC2 (Programming Contest Control): es un sistema de control y administración de competencias de programación basado en la tipología de herramientas de evaluación de un juez online desarrollado por la Universidad del Estado de California en Sacramento. Permite que los estudiantes

o equipos puedan compartir programas o enviar código a través de una red local previamente configurada para tal fin (Clifton., 2010). En este se encontrarán los jueces o docentes que correspondan y los equipos estarán inaccesibles externamente mediante la web. Los jueces o docentes podrán revisar los códigos y compilarlos, ejecutarlos, revisar los resultados y así mismo enviar mensajes a los diferentes equipos. La herramienta también permite la evaluación automática desde la configuración con el fin de evitar tener jueces humanos o docentes en el proceso, y siendo esta una de las funciones a tener presente en el contexto educativo y de enseñanza-aprendizaje de la programación. Al estar enfocado principalmente en concursos y competencias, cuenta con estas características: clasificación en tiempo real, información de envíos y cantidades de envíos delimitadas, temporalización de entregas y evaluaciones, adicional se permitan las consultas y los boletines generados para todos los equipos en el marco de la competitividad a cierto tiempo.

Igualmente, la herramienta cuenta con una amplia variedad de opciones que se permiten configurar por parte de los administradores de la competencia y así poder determinar condiciones para los concursantes. De igual manera, aunque la herramienta evalúa principalmente código en lenguaje de programación de Java, se pueden integrar otros lenguajes a partir de interfaz que permite añadir librerías y compiladores desde la herramienta.

PETCHA: es una herramienta desarrollada en la Universidad de Oporto y tiene como objetivo ser un asistente automático para la práctica de la enseñanza de la programación (Queirós y Leal., 2012). Entre sus características importantes: permite la integración con LMS como Moodle, pero igualmente con IDE como Eclipse o Visual Studio; soporta múltiples lenguajes; y cuenta con una interfaz gráfica que facilita las actividades tanto al estudiante como al docente. Si bien su propósito está en permitir apoyar la enseñanza, también tiene un uso a nivel competitivo para maratones de programación. Cuenta con elementos de las herramientas de evaluación automática, tales como: la retroalimentación a partir de los casos de prueba configurados por los docentes, la creación de múltiples ejercicios con las descripciones puntuales para los estudiantes, la creación de repositorios de ejercicios para su uso en distintos ambientes, la ejecución de actividades desde IDEs externos. En el estudio realizado por Queirós y Leal (2018) se indica que el integrar la herramienta en las prácticas de clases permite mejorar la cantidad de entregas que requieren estos para solucionar los diferentes problemas planteados, por lo que su intervención podría mejorar aspectos de eficiencia y rendimiento académico si se amplía su uso.

Pythia: es un framework desarrollado por Sébastien Combéfis y Vianney le Clément de Saint-Marcq en la Universidad Católica de Lovaina en Bélgica (Combéfis y Saint-Marcq., 2012). La plataforma busca ser un ambiente de aprendizaje para personas que no tengan ninguna clase de conocimiento en programación o de igual manera para personas que quieran mejorar sus habilidades de programación. El estudiante podrá realizar la codificación y, al mismo tiempo, obtener retroalimentación por parte de esta. Está enfocado en resolver los problemas aislados que enfrentan los estudiantes al iniciar en el mundo de la programación: perspectiva de problemas enormes, formas complejas de encontrar

soluciones y falta de motivación en concreto. Igualmente, esta herramienta permite programar en distintos lenguajes siempre que se disponga de un intérprete o compilador.

En el estudio realizado por Combéfis y Saint-Marcq (2012) proponen un conjunto de cursos de programación para la resolución práctica. Los estudiantes podrán enviar sus respuestas y ejecutar de forma segura sus soluciones a la vez que reciben comentarios y retroalimentación de forma completamente automatizada. Combéfis y de Moffarts (2019) indican que la herramienta aporta considerablemente al estudio de la programación de computadores, apoyando las clases prácticas, y adicional obtiene una mejora en los estudiantes al contar con retroalimentación automática, apoyando un aprendizaje a partir de sus propios fallos.

SPOJ: la herramienta denominada SPHERE Online Judge (SPOJ) fue desarrollada el 2003 por la Universidad Tecnológica de Gdansk (SPHERE-Online-Judge., 2022). Es una herramienta que permite por un lado ser un juez en línea, y por otro, una plataforma de aprendizaje, reforzando principalmente el conocimiento en algoritmos. Ha sido utilizada en diversas competencias de programación y a partir de los resultados y la retroalimentación permite que los estudiantes puedan afianzar más sus conocimientos y habilidades en programación. Permite la evaluación automática en más de 45 lenguajes de programación y compiladores, tales como: Java, C++, Pascal, Perl o Python. Incluye en la plataforma web cerca de 13000 tareas para practicar en cualquier momento del día traducidos en múltiples idiomas. De igual manera, permite tener un sistema de pruebas de código flexible con salidas de texto personalizables y acordes a los resultados de la evaluación.

En estudios como el realizado por Kosowski *et al* (2008), se pueden evidenciar mejoras en lo que al aprendizaje de la programación se refiere, indicando la importancia de la retroalimentación a partir de procesos de evaluación de código fuente. De igual manera, se explora la necesidad que puede surgir de integrar a los estudiantes de programación en procesos de competencia realizados en herramientas de este estilo. Igualmente, en el estudio realizado por Janczewski *et al* (2006), se puede observar la importancia de incluir estas herramientas tecnológicas en las metodologías de clase de programación y el apoyo de la enseñanza de esta área.

UNCode: es un entorno educativo web desarrollado por la Universidad Nacional de Colombia para el aprendizaje de habilidades de programación a partir de evaluación automática de actividades de programación en un contexto educativo (Restrepo-Calle *et al.*, 2018). Es una herramienta basada en el núcleo de la plataforma INGINIOUS, desarrollada por Guillaume Derval, Anthony Gého y Pierre Reinbold de la Universidad Católica de Lovaina y financiada parcialmente por Wallonie (Derval *et al.*, 2014). Permite no solo la retroalimentación sumativa mediante la ponderación de una nota, sino también la retroalimentación formativa que permite a los estudiantes mejorar las soluciones que proponen aun antes de ser enviadas a evaluación. Igualmente, cuenta con diferentes componentes propios de una herramienta de evaluación automática tales como: la calificación y validación de un programa (código fuente), la retroalimentación automática e inmediata, la verificación de código a través de

análisis léxico, sintáctico y semántico, el autocompletado de código fuente y la posibilidad de evaluación y ejecución de código en distintos lenguajes de programación (Java, C/C++, Python).

En el estudio realizado por Restrepo-Calle *et al* (2020) se pueden observar los resultados al integrar este tipo de herramientas en el ámbito académico y en clases de programación. La herramienta apoya el proceso de aprendizaje de los estudiantes al permitirles la posibilidad de obtener retroalimentación automática, la posibilidad de probar su código, de practicar de manera frecuente al contar con una herramienta en línea disponible para su uso. En la investigación realizada por Ramírez-Echeverry *et al* (2022) se observa una percepción positiva de los estudiantes al integrar herramientas de evaluación en su proceso de formación, permitiéndoles mejorar la resolución de problemas de programación en cursos enfocados en programación de computadores. Además, en el estudio de González-Carrillo *et al* (2021), UNCode se integró en asignaturas relacionadas con inteligencia artificial, aprendizaje de máquina e introducción a sistemas inteligentes en el cual indicaban los estudiantes que la retroalimentación formativa de estas herramientas de evaluación es fundamental para su proceso de aprendizaje, señalando como ventajas: la posibilidad de detectar errores en el código, la posibilidad de corregir rápidamente y solucionar problemas correctamente, y la posibilidad de probar sus programas a partir de casos de prueba adicionales.

UVa Online Judge: la herramienta UVa Online Judge fue desarrollada por Ciriaco García de Celis de la Universidad de Valladolid (UVa) en 1995 (Revilla *et al.*, 2008). En sus inicios la herramienta fue desarrollada con el fin de potencializar las competencias de programación de la universidad. El objetivo de la herramienta es tener un juez online con un carácter más pedagógico y educativo en la educación superior, permitiendo no solo contemplar la posibilidad de la competencia, sino también como apoyo a la enseñanza y el aprendizaje de la programación. Es una herramienta que permite la evaluación automática a partir de casos de prueba, la realización de ejercicios de programación de un repositorio de problemas propio y la evaluación sumativa de los proyectos enviados. En el estudio realizado por Revilla *et al* (2008) se observa una mejora en cuanto a las habilidades de programación de un curso de algoritmos, en el cual los estudiantes a través de la práctica ya sea a nivel individual o cooperativo pueden fortalecer las habilidades de programación.

Virtual Programming Lab (VPL): es un ambiente de evaluación automática desarrollado por la Universidad de Las Palmas de Gran Canaria (ULPGC) (Rodríguez-del-Pino., 2012). Esta herramienta funciona como un complemento (plugin) del LMS Moodle y permite realizar actividades de programación con un ambiente de desarrollo propio. En las actividades planteadas el estudiante puede ingresar el código fuente de manera externa (mediante un archivo correspondiente al lenguaje de programación que desarrolla) o mediante el ingreso en el editor de código, en el cual podrá ejecutar y/o evaluar su solución acorde a unos casos de prueba determinados por el profesor. La aplicación en este contexto es académica principalmente y permite la realización de código fuente en más de 45 lenguajes de programación, dentro de los cuales se puede realizar desarrollo con interfaz gráfica de usuario e integración con bases de datos. De igual manera, permite al docente revisar en detalle cada uno de

los códigos enviados por el estudiante con la posibilidad de realizar una comparación de todos los códigos, identificando a nivel léxico, sintáctico y semántico si existe plagio o copia entre los códigos enviados.

En el estudio realizado por Cardoso, Marques *et al* (2020) se puede evidenciar una clara mejora en los aspectos de programación de computadores. En este estudio se aplicó durante dos años la integración de esta herramienta en las clases de programación con el fin de mejorar el aprendizaje de la programación en estudiantes de primeros semestres. Aproximadamente 630 estudiantes hicieron parte de la investigación y se obtuvieron buenos resultados de la herramienta valorando la ejecución de código antes de evaluar automáticamente y la integración de retroalimentación en el proceso. Por otra parte, en el estudio de Lovos y González (2014) resultó innovadora la manera de integrar tecnologías en el ámbito de las clases de programación, apoyados por el trabajo colaborativo, permitiendo de igual manera obtener datos relevantes por parte de los docentes con el fin de determinar mejoras a las metodologías de las clases de esta área. El uso de la herramienta, lo indica Rodríguez *et al* (2011) en su estudio, permite mejorar notablemente habilidades y destrezas propias del campo lógico de la programación, con lo cual las integraciones obtienen un buen resultado si se integran en etapas tempranas del aprendizaje. En el estudio Barrios y Marin (2014), se propone la integración de laboratorios virtuales con el fin de garantizar un aprendizaje exitoso en el campo de la programación, permitiendo obtener un ambiente en donde la práctica fortalezca estas habilidades. Por otra parte, Thiébaud (2015) indica que si bien es un complemento muy bueno para las etapas tempranas del aprendizaje de la programación, puesto que permite a los estudiantes probar sus soluciones en base a una problemática, también requiere un esfuerzo por parte de los docentes en lo que concierne a la preparación de las actividades, el entorno de pruebas y la validación de los casos de prueba a comparar, pero posterior a este esfuerzo inicial la herramienta apoya todo el proceso de aprendizaje y enseñanza con menor esfuerzo por parte del docente.

Web-CAT: es un sistema de evaluación automática basado en la web desarrollado por Stephen Edwards de Virginia Tech (Edwards y Pérez-Quñones., 2008). Esta herramienta proporciona comentarios sobre los trabajos dirigidos por los estudiantes y permite la calificación automática basada en casos de prueba. Una de las características más importantes es que permite calificar a los estudiantes a través de lo bien que prueban su propio código y, al mismo tiempo, los estudiantes pueden medir en pro de las mejoras que han tenido con cada prueba. De igual manera, uno de los objetivos de la herramienta es que los estudiantes puedan desarrollar código de una mejor calidad a su vez que permite que los estudiantes tengan una retroalimentación completa en todo el proceso de aprendizaje. En el trabajo de maestría de Manniam Rajagopal (2018) se hace uso de esta herramienta la cual no solo apoya el proceso de aprendizaje de la programación, sino que a su vez ayuda a aliviar emociones negativas que se pueden producir al enfrentarse a la frustración al momento de realizar ejercicios de programación, por lo cual es un aporte importante en el campo de la pedagogía en este campo de acción.

3.2. Selección de la herramienta de evaluación automática

A través de la revisión de literatura se identificaron diferentes herramientas relacionadas con la evaluación automática como apoyo a la enseñanza y aprendizaje de la programación de computadores. Con el propósito de identificar ventajas y desventajas, así como similitudes y diferencias de estas herramientas de evaluación automática, se construyó un cuadro comparativo, que permite determinar las características que brinda cada una de estas herramientas. El cuadro comparativo de las herramientas de evaluación automática se puede observar en la Tabla 3-1.

En el cuadro comparativo se determinan criterios para la caracterización de ambientes de evaluación automática mencionados por Gupta y Gupta (2018) y Souza *et al* (2016) (como se observa en la Tabla 3-2), tales como: el tipo de evaluación, el tipo de ejecución, el enfoque de la herramienta, la especialidad, el tipo de herramienta, los lenguajes soportados, el tipo de retroalimentación y la población de enfoque.

Teniendo en cuenta que cada una de las herramientas cuenta con diferentes características, en las cuales se destacan algunas por varios de los criterios anteriores, se identificaron diferentes criterios a partir de los estudios realizados por múltiples autores, que apoyaron a la selección de la herramienta de evaluación automática más idónea para la integración dentro de las sesiones de clase de programación de computadores. El proceso se realizó a través de dos filtros, el primero se determina a partir de los criterios de selección que más se adaptan al modelo metodológico de las clases de la universidad y los recursos con los que ésta cuenta para el apoyo de la enseñanza y aprendizaje. Por otro lado, a partir de este primer filtro se analizaron los componentes de una herramienta de evaluación automática, y así, se seleccionó la herramienta que permitía obtener mejores resultados en el proceso de intervención en las clases.

Tabla 3-1.: Cuadro comparativo entre las herramientas de evaluación automática

N	Herramienta	Características	Lenguajes	Tipo Herramienta	Evaluación	Ejecución	Retro-alimentación	Enfoque	Especialidad	Población
H1	CodeBoard IDE (Codeboard., 2020)	Permite ejecutar código fuente en plataforma independiente, se puede integrar con calificaciones de Moodle, no permite evaluación ni retroalimentación	Múltiples*	Plugin Moodle	Automática	Dinámica	Manual	Enfoque mixto	Enseñanza-Aprendizaje	Universidad
H2	CodeRunner (Lobb y Harlow., 2016)	Apoyo en procesos evaluativos mediante la integración de quices o actividades con ejecución de código en respuesta o selección.	Múltiples*	Plugin Moodle	Semi-automática	Estática	Manual	Centrado en el docente	Enseñanza-Aprendizaje	Universidad
H3	DOMJudge Online (DomJudge-Online., 2022)	Herramienta de evaluación de código fuente para procesos de maratonos o competencia, solamente evalúa lo enviado y que cumpla con casos de uso de prueba.	Java, Python	Web	Automática	Dinámica	Automática	Enfoque mixto	Competición	Desarrolladores
H4	JAssess (Yusof et al., 2012)	Evaluador y compilador de java en Moodle. Retroalimentación manual por parte del docente. Envío de código fuente en zip.	Java	Web	Manual	Estática	Manual	Centrado en el docente	Enseñanza-Aprendizaje	Universidad
H5	JavaBrat (Patil., 2010)	Integración con Moodle, evaluación de código en Java.	Java	Plugin Moodle	Semi-automática	Estática	Automática	Centrado en el estudiante	Enseñanza-Aprendizaje	Secundaria
H6	Marmoset (J. Spacco, Hovemeyer et al., 2006)	Información detallada. Independencia del idioma. Módulo de seguridad y escalabilidad para evaluación.	C, Ruby, Ocaml, Java	Escritorio	Automática	Dinámica	Automática	Centrado en el estudiante	Testing de software	Desarrolladores
H7	Mooshak (Leal y Silva., 2003)	Apoyo en programación competitiva o maratonos de programación. Perfiles de usuario independientes. Ambiente web.	C, Prolog, Tcl/Tk, SQL	Web	Semi-automática	Dinámica	Automática	Enfoque mixto	Competición	Universidad
H8	PC2 (Programming Contest Control) system (Clifton., 2010)	Apoyo a los procesos de competición. Cuenta con componentes para los jurados, los equipos, la tabla de puntuación, el administrador y el servidor de ejecución.	Java, C++, Python	Web	Automática	Dinámica	Automática	Enfoque mixto	Competición	Desarrolladores
H9	PETCHA (Queirós y Leal., 2012)	Uso de tecnología como interoperabilidad, apoyo a programación competitiva. Retroalimentación inmediata	Múltiples*	Plugin Moodle	Automática	Dinámica	Automática	Enfoque mixto	Enseñanza-Aprendizaje	Universidad
H10	Pythia (Combéfis y Saint-Marcq., 2012)	Permite la construcción de problemas completos y evaluación de código a partir de casos de prueba. Configuración y calificación automática en el proceso de enseñanza.	Python	Web	Automática	Dinámica	Automática	Enfoque mixto	Enseñanza-Aprendizaje	Universidad
H11	SPHère Online Judge (SPHère-Online-Judge., 2022)	Permite el registro de cualquier usuario y cuenta con repositorio de ejercicios. Planteado inicialmente para competición de programación. Existen rankings y foros de discusión.	Múltiples*	Web	Automática	Dinámica	Automática	Enfoque mixto	Competición	Desarrolladores
H12	UNCode (Restrepo-Calle et al., 2018)	Ambiente de evaluación automática de código fuente, permite el envío de código y la evaluación a partir de casos de prueba, permite revisar retroalimentación y configurarla.	Python, C++, Java	Web	Automática	Dinámica	Automática	Enfoque mixto	Enseñanza-Aprendizaje	Universidad
H13	Uva Edujudge (Revilla et al., 2008)	Se consolida inicialmente como una herramienta de apoyo a la competición en pruebas locales. Cuenta con apoyo estadístico y reportes para las competencias.	Pascal, C, C++ y Java	Plugin Moodle	Automática	Dinámica	Automática	Centrado en el docente	Enseñanza-Aprendizaje	Universidad
H14	Virtual Programming Lab (VPL) (Rodríguez-del-Pino., 2012)	Evaluación automática, calificación inmediata, retroalimentación automática, detección de plagio, ejecución de código fuente, calificación a través de casos de uso de prueba, integración con Moodle.	Múltiples*	Plugin Moodle	Automática	Dinámica	Automática	Enfoque mixto	Enseñanza-Aprendizaje	Universidad
H15	Web-CAT (Edwards y Pérez-Quiñones., 2008)	Extensibilidad y flexibilidad basada en complementos. Seguridad de acceso. Portabilidad. Evaluación automática y semiautomático.	Java, C++ y Pascal	Web	Semi-automática	Dinámica	Automática	Enfoque mixto	Testing de software	Desarrolladores

* Soporta los lenguajes Java, Python, C++, y además, muchos otros.

Tabla 3-2.: Criterios de selección de la herramienta

Categoría	Criterio
Tipo de evaluación	Evaluación manual Evaluación semi-automática Evaluación automática
Ejecución	Estática Dinámica
Enfoque	Centrado en el docente Centrado en el estudiante Enfoque mixto
Especialidad	Enseñanza - Aprendizaje Competición Quices Testing de Software
Tipo de Herramienta	Web Plugin Escritorio - Local
Lenguajes soportados	Java Python C++
Retroalimentación	Manual Automática
Población	Primaria Secundaria Universitaria Desarrolladores

Fuente: Adaptado de Gupta y Gupta (2018)

En la primera parte se filtraron herramientas de evaluación automática a partir de unas condiciones dadas por los recursos tecnológicos propios de la universidad y por las indicaciones en el proceso formativo que se lleva a través del currículo. Uno de los principales criterios de selección es que, durante el proceso de clases en la intervención, la pandemia COVID-19 llevó a que las clases fueran remotas, mediante herramientas de comunicación virtual. Por lo cual, fue fundamental apoyar las clases con sistemas de administración del aprendizaje (LMS), tales como Moodle, Google Classroom o Blackboard. Para este caso puntual, la Fundación Universitaria Compensar cuenta con el LMS Moodle para procesos de administración, seguimiento y monitoreo de cursos y estudiantes, por lo que era fundamental que la herramienta seleccionada fuera compatible con la integración en Moodle. Otro factor fundamental para la selección es que la especialidad de la herramienta fuera el proceso de enseñanza y aprendizaje en el contexto académico, puesto que muchas de las herramientas identificadas a través de la literatura tenían un gran reconocimiento y enfoque en las competencias de programación o las pruebas de software. La población objetivo de la herramienta se deseaba fuera la comunidad universitaria, teniendo en cuenta que hay varias herramientas que buscan fortalecer las habilidades lógicas y de programación en un entorno poblacional de educación primaria, y en otros casos, con un nivel avanzado, como la comunidad de desarrolladores. Para el caso del lenguaje de programación, la universidad sugiere que las prácticas de la asignatura se realicen en el lenguaje de programación C++, dada la continuidad que tiene con otras asignaturas posteriores, en las cuales se utiliza dicho lenguaje, por lo cual, para la selección se requería que la herramienta fuera compatible con este lenguaje de programación.

A partir de la identificación de estos criterios de selección, de las 15 herramientas de evaluación comparadas en la Tabla 3-1, las siguientes herramientas de evaluación automática cumplen con las condiciones planteadas:

- Codeboard (Codeboard., 2020)
- CodeRunner (Lobb y Harlow., 2016)
- PETCHA (Queirós y Leal., 2012)
- VPL (Rodríguez-del-Pino., 2012)

La segunda parte del proceso de filtrado se realizó con el fin de identificar la herramienta más adecuada para la integración en el curso de programación de computadores. Esto implica que no solo es suficiente encontrar criterios que se adecuen a los lineamientos metodológicos de la universidad, sino también que permitan facilitar el proceso de programación a los estudiantes y el proceso de evaluación y análisis del desempeño por parte del docente. Por un lado, para los estudiantes puede resultar importante, por ejemplo, ejecutar código fuente en la misma herramienta, realizar reenvíos de código, obtener una buena retroalimentación o encontrar fácilmente las causas de los errores en sus soluciones. Por otro lado, para el docente puede ser importante revisar cada código del estudiante, verificar la cantidad de entregas, tener reportes estadísticos o verificar si se realizó algún tipo de plagio o copia entre los estudiantes. Teniendo presente esto, las características de una herramienta de evaluación automática indicadas por Restrepo-Calle *et al* (2018) se tuvieron en cuenta para esta parte del proceso de selección, planteando la importancia de que, tanto para los estudiantes como para los docentes, estas herramientas de evaluación automática integradas en el contexto educativo cuenten con una buena retroalimentación formativa y sumativa.

A partir de las 31 características de las herramientas de evaluación automática determinadas para la selección, VPL cumple con 30 de las características como se puede observar en la Tabla 3-3, siguiendo Codeboard con 23 características, PETCHA con 22 característica y CodeRunner con 11 características. Teniendo en cuenta este análisis comparativo, se pudo identificar a VPL como la herramienta de evaluación automática propio para la integración en las sesiones de clase de programación de computadores.

Tabla 3-3.: Comparación entre herramientas de evaluación automática

Tipo	Característica	Codeboard	CodeRunner	PETCHA	VPL
Retroalimentación sumativa	Sintaxis	•			•
	Semántica (funcionalidad)	•		•	•
	Eficiencia	•		•	•
	Calificación numérica	•	•	•	•
	Reenvíos	•	•	•	•
	Calificación parcial	•			•
	Calificación automática (numérica)	•	•	•	•
	Calificación manual			•	•
Retroalimentación formativa	Resaltado de sintaxis	•			•
	Autocompletado de código	•			•
	Mantenibilidad de código (linters)	•			•
	Visualización de ejecución de código	•	•		•
	Pruebas de entrada personalizadas	•		•	•
	Envío de proyectos	•	•	•	•
	Asistente de configuración altamente personalizable	•	•	•	•
	Visualización del espacio de soluciones	•		•	•
	Puntuación (tablero de puntaje)			•	•
	Informes estadísticos básicos	•		•	•
	Informes estadísticos interactivos			•	
Características técnicas	Lenguajes de programación	Muchos	Java, C/C++, Python	Java, C/C++	Muchos
	Basado en Web	•	•	•	•
	Administración de usuario	•		•	•
	Administrador de cursos y tareas	•		•	•
	Soporte para LMS	•	•	•	•
	Distribución y disponibilidad	•	•	•	•
	Detección de plagio				•
	Historial y almacenamiento de entregas			•	•
	Escalabilidad				•
	Ejecución de código fuente	•	•		•
	Temporización de ejercicios			•	•
	Control de cantidad de envíos máximos			•	•

Fuente: Adaptado de Restrepo-Calle et al (2018)

3.3. VPL: Virtual Programming Lab

Virtual Programming Lab (VPL) es un sistema de evaluación automática para la enseñanza y el aprendizaje de la programación con compatibilidad de múltiples lenguajes de programación (Rodríguez-del-Pino., 2012). La herramienta se instala como un complemento (plugin) en el LMS Moodle lo que implica que será integrado como una actividad dentro de este ambiente de aprendizaje y permitirá integrarse al libro de calificación, reportes, control de eventos, copias de seguridad, restauración, acceso basado en roles y demás funciones dentro de Moodle. VPL requiere mínimo de la versión Moodle

1.9 y de PHP 5 o superior. De igual manera, al integrarse a este LMS podrán utilizarse las configuraciones básicas de los recursos de éste, tales como: las fechas de inicio y finalización de la actividad, los permisos para los usuarios que pueden hacer uso de esta, la agrupación de estudiantes y los tipos de archivos permitidos.

VPL cuenta con un editor de código basado en un applet de Java que permite editar, ejecutar, depurar y evaluar código fuente de los programas y soluciones realizadas por los estudiantes de forma simple y sencilla. Para que VPL funcione requiere contar con un servidor *jail* (el que ejecuta cada una de las entregas a partir de la configuración de compilación brindada) el cual es el encargado de compilar y ejecutar el código fuente enviado por los estudiantes, como se puede observar en la Figura 3-1.

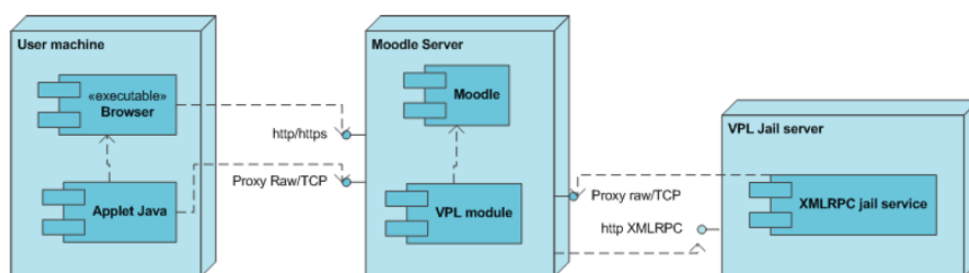


Figura 3-1.: Componentes de VPL

La herramienta VPL puede configurarse para distintos tipos de actividades de aprendizaje tales como ejemplos, ejercicios tipo rompecabezas, ejercicios de desarrollo, tareas extra clase o exámenes en clase. Las actividades de ejemplos, talleres o exámenes cuentan con una descripción del problema, unas entradas y salidas esperadas y unos casos de prueba de guía para que el estudiante pueda verificar su solución acorde a estos casos brindados. Para el caso de los ejemplos, éstos pueden ser mutables o inmutables; el inmutable es todo aquel que no puede modificarse y sirve como guía para una propia revisión; por otro lado, el mutable puede modificarse y cuenta con la opción de restablecerse a su estado original. Para el caso de los ejercicios rompecabezas, funcionan similar a los ejemplos, sin embargo, en éstos el estudiante deberá modificar las líneas que no dejan funcionar o están cortadas del código brindado. En todos los tipos de ejercicios se contará con un editor de código para la resolución en la herramienta (tal como se observa en la Figura 3-2) o una opción para cargar el programa realizado en caso de haberse utilizado alguna herramienta externa para su desarrollo.

Para cada uno de los tipos de actividades se requiere plantear unos casos de prueba con los cuales la herramienta va a comparar respecto al código ingresado por el estudiante. Estos casos de prueba están basados en una verificación entrada-salida del programa enviado, permitiendo evaluar la corrección del programa y de igual manera elaborar informes de evaluación que permitan proporcionar una retroalimentación a los estudiantes como apoyo a la calificación. La configuración de cada caso se realiza indicando el nombre del caso, las entradas y la salida esperada y el porcentaje de reducción de nota en caso de no cumplirse con alguno de estos como se observa en la Figura 3-3.

The screenshot shows the VPL interface with a C++ code editor on the left and a problem description on the right. The code editor displays the following code:

```

1 #include<iostream>
2 #include<math.h>
3
4 using namespace std;
5
6 int main () {
7     int meses, inversion, cantrime;
8     float ganancia;
9
10    cantrime =trunc(meses/3);
11
12    cout<<"Digite la cantidad de meses: ";
13    cin>>meses;
14    cout<<"Digite la cantidad de inversion: $";
15    cin>>inversion;
16
17    if(inversion>=700000 and inversion<2000000){
18        ganancia=(inversion*0.08)*cantrime;
19    } else if (inversion>=2000000 and inversion<4000000){
20        ganancia=(inversion*0.1)*cantrime;
21    } else {
22        ganancia=(inversion*0.12)*cantrime;
23    }
24    cout<<"Su inversion fue: $"<<inversion<<"\n";
25    cout<<"Sus intereses fueron: $"<<ganancia;
26 }

```

The problem description on the right is titled "EJERCICIO 3" and includes the following sections:

- Descripción del problema:** Un trabajador de Compensar desea conocer cuánto dinero generaría a través de los intereses de una inversión en su entidad bancaria. Dicha entidad fija un porcentaje de interés trimestral sobre el 8% mientras estos excedan los \$700.000, un 10% si exceden los \$2.000.000 y un 12% si excede los \$4.000.000. Dados los meses y la inversión determinar su futura ganancia
- Entrada:** La entrada debe incluir inicialmente los intereses de la inversión, seguido de la cantidad de meses en el cual dejará su dinero.
- Restricciones:**
 - 0 < Meses < 48
 - 700000 < Inversión < 100000000

At the bottom right, there is a table with the following data:

Input	Output
6	.
3000000	.
.	600000

Figura 3-2.: Actividad del curso “Algoritmos y Programación” en VPL

The screenshot shows the VPL interface with the configuration of test cases. The test cases are listed as follows:

```

1 case = Prueba1
2 grade reduction = 33%
3 input =5
4 3000000
5 output =400000
6
7 case = Prueba2
8 grade reduction = 33%
9 input =3
10 3000000
11 output =300000
12
13 case = Prueba3
14 grade reduction = 33%
15 input =6
16 3000000
17 output =100000

```

Figura 3-3.: Configuración de casos de prueba en VPL

La herramienta permite de igual manera que el docente puede verificar la cantidad de entregas de los estudiantes, pudiendo mirar cada uno de los códigos que ha enviado el estudiante, verificar cada evaluación y ejecución realizada por el estudiante, y permite que el docente pueda realizar evaluación manual y dejar comentarios a modo de retroalimentación personalizada para cada una de las entregas seleccionadas, tal como se puede observar en la Figura 3-4.

Nombre / Apellido(s)	Entregada el	Entregas	Calificación	Evaluada por	Evaluada el
1 CRISTIAN CAMILO ALFONSO CUERVO	lunes, 24 de mayo de 2021, 22:39	1	5.00 / 5.00	Calificación automática	lunes, 24 de mayo de 2021, 22:39
2 NEIDER EDUARDO AMAYA GOMEZ	sábado, 22 de mayo de 2021, 22:36	1	5.00 / 5.00	Calificación automática	sábado, 22 de mayo de 2021, 22:36
3 SERGIO	jueves, 27 de mayo de 2021,	2	5.00 / 5.00	Calificación	jueves, 27 de mayo de 2021,

Figura 3-4.: Lista de entregas de una actividad del curso “Algoritmos y Programación” en VPL

Ahora bien, uno de los problemas que se enfrentan en el ámbito de la educación en proceso de enseñanza y aprendizaje de la programación es el plagio o la distribución del código entre estudiantes. Para este caso VPL incluye una característica que permite observar la similaridad de código fuente entre los envíos realizados a través de un análisis léxico y sintáctico. La idea principal de esta función es permitir detectar el plagio realizado entre los diferentes envíos de una actividad o problemas en un curso, pero igualmente se pueden incluir otras fuentes tales como envíos de semestres anteriores o tareas similares en otros cursos. En la Figura 3-5 se puede observar una comparación entre códigos de dos estudiantes en los cuales el docente podrá identificar el porcentaje de plagio indicado por la herramienta y tomar decisiones a partir de la información suministrada.

```

taller3.cpp
1 #include<iostream>
2 #include<math.h>
3
4 using namespace std;
5
6 int main () {
7
8     int meses;
9     int valor;
10    int intereses;
11    int total;
12
13    cin>> meses;
14    cin>> valor;
15
16    if (valor >= 4000000){
17        intereses= valor*0.12;
18        total= intereses;
19    }else if (valor >= 2000000 and valor <= 3999999){
20        intereses= valor*0.10;
21        total= intereses;
22    }else if (valor>= 700000 and valor <= 1999999){
23        intereses= valor*0.08;
24        total= intereses;
25    }
26    cout<< total;
27 }

taller3.cpp
1 #include<iostream>
2 #include<math.h>
3
4 using namespace std;
5
6 int main () {
7
8     int meses;
9     int valor;
10    int intereses;
11    int total;
12
13    cin>> meses;
14    cin>> valor;
15
16    if (valor >= 4000000){
17        intereses= valor*0.12;
18        total= intereses;
19    }else if (valor >= 2000000 and valor <= 3999999){
20        intereses= valor*0.10;
21        total= intereses;
22    }else if (valor>= 700000 and valor <= 1999999){
23        intereses= valor*0.08;
24        total= intereses;
25    }
26    cout<< total;
27 }
  
```

Figura 3-5.: Similaridad de código fuente de una actividad del curso “Algoritmos y Programación” identificado por VPL

Esta herramienta permite apoyar los procesos de enseñanza y aprendizaje propios de la codificación y programación, por lo que pretende fortalecer las habilidades lógicas del estudiante en este proceso formativo. Se plantea dentro de la investigación apoyar las actividades de docencia en un 90 % con esta herramienta de evaluación en lo que corresponde a actividades de carácter práctico.

4. Diseño del estudio

El diseño llevado a cabo en esta investigación se realiza teniendo en cuenta el procedimiento de estudio de un diseño cuasi-experimental, el cual como indica Fernández: “es un plan de trabajo con el que se pretende estudiar el impacto de los tratamientos y/o los procesos de cambio en situaciones donde los sujetos o unidades de observación no han sido asignados de acuerdo con un criterio aleatorio” (Fernández *et al.*, 2014). Este diseño de estudio tiene características parecidas al diseño experimental en la medida en la que la intervención de la variable independiente impactará en la variable dependiente; sin embargo, en este diseño propuesto los grupos de estudio propiamente tendrán una asignación por conveniencia, y no aleatoria como se plantea en el diseño experimental, lo cual a su vez no permite asegurar o garantizar la igualdad inicial de los grupos. Por todo esto, se deberá tener un especial cuidado con las diferencias previas que puedan tener los diferentes grupos participantes y los efectos que estas mismas puedan presentar en el análisis de los resultados obtenidos.

4.1. Metodología planteada

En esta sección se describirá el diseño de la metodología propuesta para el presente estudio con el fin de determinar una ruta a seguir en el desarrollo de la actual investigación. Es importante determinar en el estudio las diferentes variables que se medirán posterior a los resultados obtenidos en cada una de las intervenciones según el instrumento de medición seleccionado. En este estudio el objetivo general es el de determinar el impacto que una herramienta de evaluación automática (variable independiente) produce en la motivación en el aprendizaje de los estudiantes de programación de computadores (variable dependiente).

Para cumplir con esto, en el diseño experimental planteado participaron dos grupos de estudiantes de programación de computadores. Un grupo experimental en el cual se integró la herramienta de evaluación automática en los contenidos programáticos de la asignatura, y otro grupo de control en los que se realizaron las sesiones de clase con la metodología tradicional. Primeramente, se utilizó un instrumento para recolección de información sobre la motivación en el aprendizaje de los estudiantes en los grupos participantes previo y posterior a la intervención. Al finalizar la intervención, se aplicó una encuesta tan solo en el grupo experimental, para recolectar las percepciones de los estudiantes respecto a su experiencia con la integración de la herramienta. De este modo, al concluir con la investigación se puede determinar si en el grupo experimental donde se integró la herramienta de

evaluación automática, existió o no un impacto en la motivación, así mismo determinar cuál fue la situación en relación con la motivación en el grupo de control.

En la Figura 4-1 se puede observar el diagrama que determina la metodología del diseño cuasi-experimental planteado para la investigación. Se presentan 6 fases con 11 etapas, en las cuales se desarrollaron las actividades puntuales para iniciar el proceso experimental propuesto.

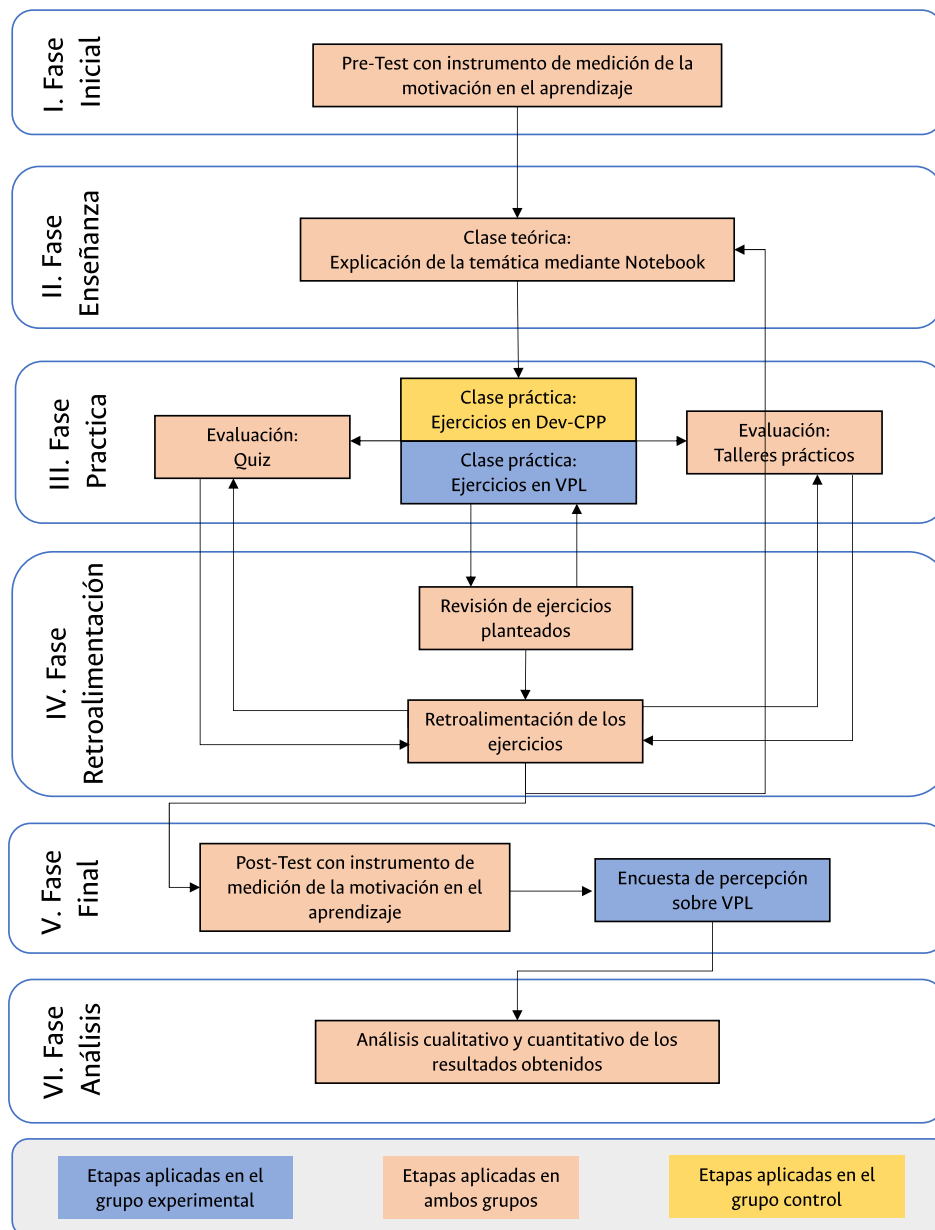


Figura 4-1.: Metodología propuesta de diseño del estudio

En la “Fase I: Inicial” se aborda a partir de una única etapa, la cual se denominó: “Pre-Test”. En dicha etapa se dieron indicaciones respecto al proceso de investigación que se estaba llevando a cabo y se

les solicitó a los estudiantes participantes en el experimento diligenciar el instrumento de caracterización de la motivación en el aprendizaje. La aplicación se llevó a cabo tanto en el grupo experimental como en el grupo control en la misma semana. Al inicio se dio una introducción al proyecto de investigación, presentando así el objeto de estudio, el propósito de la investigación y la participación voluntaria de ellos dentro del proceso experimental. Se hizo especial énfasis en la importancia de la honestidad frente a cada una de las respuestas y la relevancia dentro del proceso de estudio en la asignatura. Además, se dieron indicaciones respecto a: repercusión de la participación en las calificaciones de la asignatura, indicando su no afectación en el proceso académico, a la participación voluntaria y la confidencialidad y tratamiento de la información recolectada, indicando que se tratará de forma anónima. Se finalizó con la habilitación del cuestionario brindándoles a los estudiantes el enlace para su resolución, teniendo presente que al iniciar se encontraría el consentimiento informado.

Teniendo en cuenta lo anterior para esta primera sesión de clases se determinaron unos tiempos tal como se indica en la Tabla 4-1.

Tabla 4-1.: Diseño de la primera de sesión de clase del experimento

Introducción a la investigación	Motivación en el aprendizaje *	Introducción a herramienta de evaluación y práctica **	Características de la herramienta como estudiante y docente ***
20 minutos	40 minutos	1 hora	1 hora

* Explicación de la motivación en el aprendizaje y habilitación de instrumento de evaluación de la motivación

** Grupo control: DevCPP - Grupo experimental: VPL

*** Sobre los 10 minutos finales se brindan indicaciones del primer taller

Para esta primera sesión se tuvo en cuenta una explicación minuciosa a la investigación, teniendo presente los aspectos teóricos y lo que se desea medir. Posterior a esto se hizo una explicación sobre la motivación en el aprendizaje teniendo en cuenta que esta es la temática fundamental de la investigación, y al finalizar esta se habilita el acceso para el instrumento de medición utilizado. A partir de aquí se tendrá en cuenta la herramienta de evaluación automática previamente seleccionada y la práctica de la misma, adicional a brindar algunas características de esta desde el punto de vista del estudiante como del docente. Para el caso del grupo experimental se utilizará VPL y para el grupo control se realizará con una herramienta más tradicional como lo es DevCPP, un entorno de desarrollo integrado (IDE) para programar en los lenguajes C/C++.

En las etapas correspondientes a las “Fase II: Enseñanza”, “Fase III: Práctica” y “Fase IV: Retroalimentación” se llevaron a cabo las actividades correspondientes a la metodología de clase en las sesiones semanales, tanto en el grupo control como en el grupo experimental. En las etapas que corresponden a estas fases mencionadas se determina la estrategia propuesta para las sesiones de clase con los grupos indicados, teniendo presente la intervención de la herramienta en el grupo experimental y el trabajo de forma normal con el grupo control. La metodología para las clases se planteó teniendo en cuenta las clases tradicionales, teóricas o magistrales en las cuales el docente presenta una determinada temática a través de la revisión de distintas perspectivas y la presenta a los estudiantes con el fin

de motivarlos a explorar el contenido a mayor profundidad (Gatica-Saavedra y Rubí-González., 2020), a la vez que busca fomentar una participación activa por parte de los estudiantes. Sin embargo, las sesiones de clase bajo esta metodología conllevan a que el aprendizaje este basado principalmente en la memoria (Videla., 2010), por lo cual se hace muy necesario apoyar con metodologías activas basadas en la práctica de los diferentes conceptos adquiridos en las clases teóricas.

En la Fase II se contemplan las etapas que están directamente relacionadas con el descubrimiento de temáticas o conceptos nuevos a partir del trabajo en notebooks desde la web o el apoyo en la consolidación de fundamentos conceptuales por parte del docente. En la Fase III se agrupan todas las etapas que corresponden a la parte práctica en las cuales se integran las herramientas de evaluación de código fuente, para el grupo de control se desarrolla a partir de DevCPP y para el grupo experimental mediante VPL. En estas etapas se tienen presente los quizzes, talleres y las prácticas desarrollados y realizados a partir de ejercicios propuestos en las sesiones de clase correspondientemente. En la Fase IV se relacionan las etapas que corresponden a la retroalimentación o aclaración de inquietudes la cual está establecida dentro de la metodología de la clase, esto con el fin de brindar una explicación más acertada a cada uno de los estudiantes y permitir así despejar las dudas o vacíos propios de la nueva temática que se aborde.

Como se evidencia en la Tabla 4-2, este fue el diseño de las sesiones dentro de la metodología de la clase durante el experimento, en las cuales se iniciaba con un quiz de 30 minutos con la temática abordada en la sesión anterior, una retroalimentación por parte del docente del quiz y taller previo. Posterior a esto se brindaba toda la información de la nueva temática a partir de un componente de clase teórica mediante la explicación en una herramienta de notebook para C++, y después de esto se procedía con la parte práctica, dentro de la herramienta de evaluación correspondiente para cada grupo participante.

Tabla 4-2.: Diseño de sesiones de clase durante el experimento

Quiz *	Retroalimentación Talleres/Quiz *	Temática nueva (Clase Teórica) **	Ejercicios (Clase Práctica) * ***
30 minutos	30 minutos	1 hora	1 hora

* Herramientas: Grupo control: DevCPP - Grupo experimental: VPL

** Notebook C++

*** Sobre los 10 minutos finales se brindan indicaciones del siguiente taller

En la etapa denominada “Clase teórica: Explicación de la temática mediante Notebook” se brindan conceptos nuevos a partir de una fundamentación práctica en una herramienta de tipo notebook, teniendo presente el contenido temático propio de la asignatura a través del syllabus. Se brinda mediante una clase magistral exponiendo los componentes teóricos propios de la temática tratada. Según Gatica-Saavedra y Rubí-González (2020) la clase magistral permite presentar un tema mediante la revisión general de diferentes perspectivas, actualizar el conocimiento y describir los resultados incluyendo la experiencia para provocar en el estudiante la motivación a explorar el contenido en

mayor profundidad. En esta etapa el docente tiene la función de proporcionar los conceptos teóricos necesarios para las nuevas temáticas, relacionar cada uno de los temas unos con otros, transmitir el propósito de la enseñanza y su importancia, estructurar correctamente los temas a tratar, analizar y diagnosticar el conocimiento previo y plantear a partir de ello una mejora para que todos los estudiantes estén alineados al aprendizaje.

Dentro de la Fase III, en las etapas que corresponden a “Clase práctica: Ejercicios en DevCPP” para el grupo control y “Clase práctica: Ejercicios en VPL” para el grupo experimental, se determina el uso y apropiación de la herramienta de desarrollo y de práctica en la metodología de clase. Es aquí donde los estudiantes interactúan y hacen uso de la herramienta correspondiente para cada grupo poniendo en práctica los conceptos previamente adquiridos y llevando a cabo ejercicios propuestos para su realización individualmente. En esta Fase los estudiantes junto con el acompañamiento del docente realizan un conjunto de ejercicios en la clase teniendo en cuenta la temática planteada para cada sesión y se dejan a disposición algunos otros para su realización extraclase y de manera autónoma. En la Fase actual se propone la técnica dentro de las metodologías educativas, la cual es “aprender haciendo”, lo que permite al participante involucrarse dentro del aprendizaje mediante la experimentación, la manipulación, la práctica y los juegos en los ambientes de aprendizaje con lo que el estudiante puede descubrir y desarrollar las diferentes dimensiones de su enseñanza (Sanmartín y Pilco., 2020). Es de gran relevancia proponer este tipo de escenarios y ambientes para que los estudiantes fortalezcan y profundicen aún más el conocimiento adquirido.

En la Fase III se encuentra una integración con la Fase IV puesto que al mismo tiempo que los estudiantes realizan ejercicios y practican dentro de la herramienta de evaluación automática VPL tienen la facilidad de presentar sus inquietudes a través de la etapa denominada “Revisión de ejercicios planteados”. Es en esta etapa en donde existe un canal directo de comunicación entre el docente con cada uno de sus estudiantes, lo que permite que, bien sea en los conceptos de la clase teórica o en el ejercicio planteado en la clase práctica, puedan generar las dudas y obtener la respectiva retroalimentación, aclarando todo aquello que no quedó claro en alguna parte de la enseñanza, identificando falencias y vacíos en el aprendizaje. Es importante que el docente promueva el auto aprendizaje indicando que es el mismo estudiante el responsable de su propio aprendizaje y entendiendo este como una mejora en su proceso de formación. Las etapas correspondientes a las Fases II, III y IV son repetitivas puesto que entre más se fomenta la práctica y el aprendizaje en VPL, se van presentando dudas que requieren de una retroalimentación en el momento que se va generando.

Las otras etapas correspondientes a la Fase III, denominadas “Evaluación: Quiz” y “Evaluación: Talleres prácticos”, se incluyen como componente evaluativo y de modo práctico aumentando progresivamente la dificultad de los ejercicios dentro de VPL para el grupo experimental y mediante un envío del código fuente por la plataforma Teams en el grupo control. Por un lado, el quiz pretende ser un ejercicio rápido que permita medir el conocimiento adquirido en la sesión anterior de forma específica, mientras que los talleres son ejercicios más extensos con plazo de entrega de una semana. Los talle-

res se componían de una construcción de conceptos en los cuales se iban integrando los diferentes temas vistos a lo largo de las sesiones, por lo cual se podría decir que son un conjunto de diferentes conceptos que se agrupaban en dichas entregas.

Las Fases II, III y IV se llevaron a cabo semanalmente teniendo presente la temática abordada desde el contenido de la asignatura. Fueron 8 semanas en las que se repitieron tales fases. Adicionalmente, el 70 % de las sesiones se integró la herramienta de evaluación automática VPL dentro de las clases en el grupo experimental.

En la “Fase V: Final”, se relacionan dos etapas, la etapa denominada “Post-Test con instrumento de medición de la motivación en el aprendizaje” y la etapa “Encuesta de percepción sobre VPL”, las cuales se dan posterior a la intervención dentro de la clase con el fin de obtener los datos e información para su posterior análisis. En la etapa “Post-Test con instrumento de medición de la motivación en el aprendizaje” se aplicó el mismo instrumento utilizado en el Pre-Test en el grupo control y en el grupo experimental. Se llevó a cabo de la misma forma que en la primera sesión, teniendo presente una explicación de la investigación y la importancia de la honestidad respecto a cada uno de los datos brindados, indicando de igual manera su participación voluntaria en el proceso. Posterior a la aceptación del consentimiento informado, los estudiantes diligenciaron el cuestionario habilitado durante la sesión. La siguiente etapa referente a la “Encuesta de percepción sobre VPL” solamente se aplicó en el grupo experimental. Esta encuesta se aplicó la semana posterior a la que se realizó el Post-Test. En esta se preguntaba la opinión de cada estudiante respecto al uso y la integración de VPL en las sesiones de clase, y de manera global, al uso de estas herramientas de evaluación automática en las metodologías de clase de las asignaturas correspondientes a programación de computadores.

En la “Fase VI: Análisis” se llevan a cabo las tareas de análisis de la información extraída a partir de la aplicación de los instrumentos mencionados anteriormente. En la etapa “Análisis cualitativo y cuantitativo de los resultados obtenidos” se tienen en cuenta los datos extraídos del instrumento para caracterizar la motivación en el aprendizaje en el Pre-Test y Post-Test con el fin de determinar y analizar si hubo algún cambio en la motivación en el grupo control y el grupo experimental. Se plantea como hipótesis que el grupo experimental, al tener la herramienta de evaluación automática en las sesiones de clase, tendría una variación en la motivación después de la intervención, mientras que el grupo control se espera que no tenga incrementos significativos en su motivación. Por otro lado, los datos cualitativos recolectados con la encuesta de percepción sobre VPL aplicada en el grupo experimental, dan a conocer un panorama más amplio sobre los aspectos que influyen en la motivación en el aprendizaje de los estudiantes y cómo percibieron el uso de VPL en sus sesiones de clase. Esta información permite complementar los efectos en la motivación en el aprendizaje hallados a través del análisis de datos cuantitativos.

4.2. Participantes

Para llevar a cabo el objetivo principal del estudio, el cual es conocer el impacto que genera la integración de una herramienta de evaluación automática en la motivación en el aprendizaje de los estudiantes de programación, se contó como participantes con dos grupos, uno de control, en el cual no se integró la herramienta de evaluación y otro experimental, en el que se integró VPL dentro de las sesiones de clase. Para esta integración los participantes fueron estudiantes de los programas de Ingeniería de Sistemas e Ingeniería de Telecomunicaciones de la Facultad de Ingeniería de la Universidad Fundación Universitaria Compensar sede Bogotá en Colombia. Estos estudiantes eran de primer semestre y se encontraban matriculados en la primera asignatura de programación de computadores. La distribución y selección de los estudiantes participantes se realizó por conveniencia, ya que los grupos no fueron seleccionados aleatoriamente, sino de acuerdo a la matrícula. Los dos grupos estuvieron a cargo del mismo docente en todo el proceso de formación.

En el experimento participaron 40 estudiantes en total, 19 estudiantes matriculados en el programa Ingeniería de Telecomunicaciones correspondientes al 47,5 % de la muestra y 21 estudiantes matriculados en el programa de Ingeniería de Sistemas correspondiente al 52,5 % de la muestra. El grupo de control estuvo conformado por los estudiantes de Ingeniería de Telecomunicaciones y el grupo experimental fue conformado por los estudiantes de Ingeniería de Sistemas. Los estudiantes matriculados en ambos programas cursaron la misma asignatura denominada “Algoritmos y Programación”, la cual es una asignatura de 3 créditos, de componentes teórico-prácticos.

4.3. Integración de la herramienta de evaluación automática en la asignatura

En esta sección se explica detalladamente la integración de la herramienta de evaluación automática VPL en la asignatura de programación de computadores. En un inicio, se expone un análisis de la estructura curricular de la asignatura “Algoritmos y Programación”. En ese apartado se detalla la unidad de competencia, elementos de la competencia, resultados de aprendizaje, criterios de realización, etapa metodológica, contenidos temáticos y actividades propuestas para el programa académico de la asignatura seleccionada. Posterior a esto, se presentan las actividades en VPL propuestas para integrar en las sesiones de clase del grupo experimental, teniendo en cuenta la conexión de estas actividades con las temáticas planteadas por la estructura curricular del curso. Por último, se enuncia la metodología propuesta para cada sesión de clases con el grupo experimental, presentando detalladamente la manera en la que se desarrolló cada sesión semana a semana.

4.3.1. Estructura curricular de la asignatura

Para la integración de la herramienta de evaluación automática se tuvo en cuenta la estructura curricular establecida en la asignatura “Algoritmos y Programación” de la Universidad Fundación Universitaria Compensar sede Bogotá del grupo de Ingeniería de Sistemas, el cual será el grupo experimental en el proceso. La asignatura tiene como principal objetivo dar a conocer al estudiante las herramientas necesarias para construir algoritmos que permitan resolver problemas asociados con la construcción de programas, usando los elementos y estructuras de control básicas en un lenguaje de programación.

Respecto a la dinámica de las clases y al tener en cuenta que la asignatura es de carácter teórico-práctico, se deben contemplar la integración de aspectos y fundamentos teóricos en los procesos académicos y de enseñanza en la programación, así como fomentar un proceso de aprendizaje práctico a través de ejercicios en las sesiones de clase, talleres y quices que permitan plantear un escenario más real en el que el estudiante se enfrente a retos que permitan ampliar su proceso de aprendizaje y el proceso de construcción de programas.

Es importante tener presente que la asignatura está planteada desde la universidad con una metodología de aprendizaje basado en proyectos (ABP), en la que los estudiantes adquieren un rol activo y se favorece la motivación académica. Sin embargo, al momento de integrar la herramienta de evaluación se hará uso de un enfoque pedagógico denominada aprendizaje basado en retos (ABR), el cual permite ubicar al estudiante en un contexto real al involucrarlo en el proceso de desarrollo emulando experiencias de este campo. Esto a su vez, le permitirá enfrentarse a desafíos puntuales, obteniendo un gran potencial en el campo de la educación en ingeniería a través de: las simulaciones de maraton de programación, la resolución de problemas algorítmicos o la preparación para procesos de reclutamiento de personal en la industria del desarrollo de software.

Las actividades integradas en el ambiente virtual de aprendizaje promueven la práctica a través de los diferentes ejercicios brindados para tal fin. Estos podrán ser resueltos en distintos momentos de su proceso de formación sin afectar su proceso académico. Dicha disponibilidad les permitirá de una u otra manera mejorar su proceso de aprendizaje. Esta mejora se puede dar por medio de un aprendizaje autónomo que les permita también generar dudas e inquietudes de las temáticas planteadas en cualquier momento de su formación.

En la Tabla 4-3 se pueden evidenciar la unidad de competencia, elementos de competencia en el cual aparecen dos categorías: Razonamiento cuantitativo (RC) y Lectura crítica (LC), resultados de aprendizaje, criterios de realización (CR), etapa metodológica, contenidos temáticos y actividades, según el currículo de la asignatura “Algoritmos y Programación” del programa de Ingeniería de Sistemas.

Tabla 4-3.: Syllabus o contenido curricular de la asignatura de “Algoritmos y programación”

Unidad de competencia	Resultado de aprendizaje	Elemento de la competencia	Criterios de realización	Etapas metodológicas	Contenido temático	Actividades
Codificar soluciones de software empleando lenguajes y paradigmas de programación que cumpla con los requisitos de la solución informática.	Resuelve problemas abstractos para describir el funcionamiento de un sistema o parte de él, mediante el uso de ciclos, sentencias de decisión, variables y funciones con base en pseudocódigos y diagramas de flujo.	RC 1 - Establece los tipos de datos, sus características y el tipo de variable definida por la situación. Se propone el desarrollo de algoritmos secuenciales, mediante pseudocódigo o diagramas de flujo, utilizando los diferentes tipos de variables y expresiones.	CR1 - Resolver problemas de programación secuencial, utilizando elementos básicos de programación y representados mediante diagramas de flujo	Etapa 0 - Reconocimiento	Concepto de algoritmo y característica de los algoritmos	
					Diagramas de flujo para construcción de algoritmos	
					Diseño y resolución de expresiones aritméticas, relacionales y lógicas	
		RC 2 - Selecciona, organiza y estructura los datos e información suministrada para dar solución a una situación problemática planteada.		Etapa 1 - Contextualización	Jerarquía de los operadores	
					Operadores Aritméticos, relacionales y lógicos	
					Operadores Aritméticos, relacionales y lógicos	
	Introducción al lenguaje de programación	Presentación del Entorno de Desarrollo Integrado (IDE)	1. Taller: Básico de programación			
	Tipos de datos, declaración de variables y constantes	1. Quiz: Básico de programación 2. Taller: Variables entrada/salida				
	RC 3 - Se propone codificar algoritmos utilizando sentencias de decisión, simples y anidadas y los diferentes tipos de ciclos repetitivos.	Etapa 2 - Profundización	Construcción de programas a partir de algoritmos secuenciales	2. Quiz: Variables entrada y salida 3. Taller: Condicional simple		
			Sentencia de decisión simple y anidadas	3. Quiz: Condicionales simples 4. Taller: Condicional anidada		
			Sentencias de repetición condicionada al final Sentencias de repetición condicionadas al comienzo	5. Quiz: Switch Case 6. Taller: Ciclo FOR		
	Aplicar buenas prácticas de calidad en el proceso de programación de software, de acuerdo con el referente adoptado en la organización.	LC 1 - Comprende cómo se articulan las partes de un texto para darle un sentido global.	CR2 - Construir algoritmos que involucren sentencias de decisión y de repetición	CR3 - Construir métodos documentados y utilizarlos en la construcción de programas	Etapa 3 - Transferencia	
LC 2 - Jerarquiza las ideas secundarias en función de la idea principal que compone un texto.			Introducción al uso de métodos Construcción de métodos con y sin retorno			7. Quiz: Ciclos simples y anidados WHILE
LC 3 - Comprende las relaciones entre diferentes partes o enunciados de un texto.			Construcción de métodos con y sin parámetros Construcción de programas que involucran el uso de métodos			

Fuente: Elaboración propia

Respecto a los elementos de la competencia, según el contenido curricular de la asignatura, se contemplan dos categorías, las cuales son:

- **Razonamiento cuantitativo (RC):** Conjunto de componentes a nivel técnico, lógico y/o matemático que se requieren para adquirir un conocimiento puntual, teniendo presente la competencia.
- **Lectura Crítica (LC):** Son habilidades o elementos transversales que facilitan la abstracción de información y la agrupación de ideas propias para adquirir un conocimiento.

Por otra parte, en el caso de las etapas metodológicas dentro de la estructura curricular de la asignatura, éstas se pueden abordar de la siguiente manera:

- **Etapas 0 – Reconocimiento:** En esta etapa el estudiante empieza a identificar los aspectos fundamentales de la asignatura, el contenido, su importancia en el contexto de la ingeniería y los conceptos fundamentales.
- **Etapas 1 – Contextualización:** En la etapa de contextualización se involucra al estudiante en los elementos principales de la asignatura, aportando a construir conceptos a partir de abordar los aspectos esenciales del programa.
- **Etapas 2 – Profundización:** En la etapa de profundización se llevan a cabo todos aquellos conceptos en un proceso de inmersión sobre temáticas adicionales que lleven al estudiante a explorar a través de distintas fuentes un conocimiento de forma autónoma.
- **Etapas 3 – Transferencia:** En la etapa de transferencia el estudiante comprende los aspectos relevantes de la asignatura y puede complementar el conocimiento a partir de los planteamientos básicos finales de la asignatura con el fin de que amplíe mucho más el conocimiento y lo conecte con las asignaturas posteriores, permitiendo que llegue preparado para el siguiente escalón en su proceso de formación.

4.3.2. Configuración de las actividades a desarrollar dentro de la herramienta de evaluación automática

Inicialmente para la configuración se solicita a la universidad acceso al ambiente virtual de aprendizaje basado en Moodle puesto que VPL funciona como una extensión de este LMS, tal como se indicó en la Sección 3.3. A partir de aquí se deben crear y configurar todas las actividades dentro del curso utilizando VPL, teniendo en cuenta la descripción, los casos de prueba, el lenguaje de programación esperado, los archivos esperados, la cantidad de intentos y ejecuciones, y los tiempos de apertura y finalización de cada actividad. Para la descripción se estructura a partir de la especificación del problema, entradas, salidas, restricciones y casos de prueba de ejemplo para que el estudiante pueda corroborar a partir de estos si su código cumple en principio con estos. Cada una de las actividades se

creó en base a la temática planteada por el contenido curricular de la asignatura.

Cada actividad tanto las de práctica como las evaluativas estaban agrupadas por módulos que permitían categorizar el componente al cual hacía referencia dicha actividad, teniendo como componentes: Elementos básicos de la programación, estructuras de control condicional y estructuras de control iterativo, facilitando de igual manera su búsqueda y consulta.

Las actividades evaluativas tales como los talleres y quices tenían el mismo objetivo que era evaluar el conocimiento a partir de un ejercicio con enfoque a un caso real. Los quices tenían un planteamiento más sencillo de alguna temática con el fin de realizarlos en un periodo corto de tiempo como se indicó en la metodología. Por otro lado, el taller buscaba llevar al estudiante a explorar de manera más amplia los conceptos adquiridos a través de los ejercicios prácticos suministrados y las clases teóricas; estaban compuestos por algunos ejercicios que integraban varias temáticas previamente abordadas en las sesiones de clase. En total fueron 14 actividades evaluativas distribuidos de la siguiente forma: 7 quices y 7 talleres.

A continuación, se exponen de manera más detallada los contenidos temáticos propuestos durante el experimento tanto en las sesiones teóricas como en las actividades evaluativas (quices, talleres y ejercicios de práctica) dentro del curso:

- **Básico de programación:** Las actividades correspondientes a este ítem relacionan el conocimiento de un entorno de desarrollo IDE, las funciones elementales de estos, la introducción al uso de la herramienta de evaluación automática en el grupo experimental o herramienta de desarrollo en el grupo de control, la explicación de los tipos de datos en el lenguaje de programación a tratar y demás aspectos importantes del lenguaje seleccionado, a su vez de un planteamiento práctico para familiarizarse con la herramienta.
- **Variables entrada/salida:** Las actividades correspondientes a “Variables, entrada/salida” buscan introducir al estudiante a conocer los primeros pasos de la programación. Se describen las partes esenciales de un programa y se brinda información relevante de C++. También se brinda una explicación respecto a las variables, su proceso en memoria y los tipos de datos a tener en cuenta. De igual manera se integran comandos de entrada y salida para el lenguaje de programación seleccionado.
- **Condicionales simples:** En estas actividades se presentan los diferentes conceptos orientados a las condiciones simples, integrando directamente un refuerzo de las entradas y salidas, y fortaleciendo los aspectos de variables y tipos de datos dentro del contexto de la programación.
- **Condicionales anidadas:** Se abordar los conceptos previos, sin embargo, se amplía la conceptualización de las condiciones a partir de una anidación condicional IF-ELSE, o una anidación comprimida con IF-ELSE IF. De igual manera, éstos integran entrada y salida de datos por consola, y uso de variables definidas por el estudiante.

- **Switch case:** En esta actividad se brinda la introducción a las estructuras de control condicional por casos, resaltando las diferencias con la estructura condicional IF. Se indica su uso en casos reales y se resalta la utilidad de éste en la construcción inicial de menús que permitan al usuario identificar la función a realizar. Se tiene en cuenta también el componente lógico que a su vez se integran en estas estructuras de control condicional.
- **Ciclos simples y anidados FOR:** Se brinda una explicación detallada sobre las estructuras de control iterativas. Para estas sesiones se abordan inicialmente los ciclos FOR, indicando su importancia, los casos de uso y relevancia dentro de la programación, los componentes que se incluyen en estas iteraciones, conceptos de incrementos, pruebas lógicas, y ciclos anidados.
- **Ciclos simples y anidados WHILE:** Se indica una introducción sobre la estructura de control iterativa WHILE. En las sesiones correspondientes se detallan las diferencias con la iteración FOR, los casos de aplicación en el contexto de ingeniería, su complemento con DO WHILE, la forma de anidar los ciclos basados en WHILE y la manera en la que se integran los conceptos previos en estas últimas temáticas.

A partir de las temáticas abordadas anteriormente se determinaron las actividades evaluativas y así mismo fueron integradas dentro de la metodología del curso. Estas temáticas están relacionadas con los contenidos temáticos planteados por la asignatura y permiten abordar los componentes esenciales del curso de programación desde lo más básico.

4.3.3. Relación entre los contenidos temáticos de la asignatura y las actividades de VPL

En esta sección se evidenciará la relación entre los contenidos temáticos planteados por la estructura curricular de la asignatura y las actividades desarrolladas en VPL para el grupo experimental. Para esto, la Tabla 4-4 relaciona las temáticas propuestas y los contenidos temáticos del currículo de la asignatura con las actividades la herramienta de evaluación VPL. Como se puede observar en la tabla, se aborda toda la temática propuesta por la universidad, a su vez que se integra una herramienta con el ánimo de buscar generar motivación en el proceso de aprendizaje de los estudiantes.

Las actividades planteadas durante las 8 semanas que conllevó la experimentación están compuestas por las actividades básicas con las cuales se busca llevar a iniciar al estudiante en el ámbito de la programación, al descubrimiento de las estructuras de decisión, y el conocimiento de las estructuras iterativas.

Tabla 4-4.: Relación actividades en VPL con el contenido temático de la asignatura

Temática propuesta	Contenido temático	Actividad en VPL
Explicación de Herramienta VPL	Presentación del Entorno de Desarrollo Integrado (IDE)	1. Taller: Básico de programación
Variables Tipos de datos Entradas y Salidas	Tipos de datos, declaración de variables y constantes	1. Quiz: Básico de programación 2. Taller: Variables entrada/salida
Sentencias de decisión simple IF IF ELSE	Construcción de programas a partir de algoritmos secuenciales	2. Quiz: Variables entrada y salida 3. Taller: Condicional simple
Sentencias de decisión anidadas IF ELSEIF	Sentencia de decisión simple y anidadas	3. Quiz: Condicionales simples 4. Taller: Condicional anidada
Sentencias de decisión múltiple SWITCH CASE	Sentencia de decisión múltiple (casos)	4. Quiz: Condicionales anidadas 5. Taller: Switch case
Sentencias iterativas o por ciclos simple y anidados FOR	Sentencias de repetición Sentencia de repetición condicionada al final Sentencias de repetición condicionadas al comienzo	5. Quiz: Switch Case 6. Taller: Ciclo FOR
Sentencias iterativas o por ciclos simple y anidados WHILE, DO-WHILE	Iteraciones ascendentes, descendentes Construcción de programas a partir de algoritmos que involucran control de flujo de ejecución con sentencias de repetición	6. Quiz: Ciclos simples y anidados FOR 7. Taller: Ciclo WHILE 7. Quiz: Ciclos simples y anidados WHILE

Fuente: Elaboración propia

De igual manera se puede observar en la tabla que la integración de la herramienta dentro del proceso de enseñanza y aprendizaje en la asignatura permite continuar con la estructura general del curso, a su vez que permite generar un proceso práctico dentro de cada una de las temáticas abordadas en las sesiones, en la disponibilidad y espacio deseado por el estudiante. Esto permite centrar el aprendizaje al propio ritmo de los estudiantes y alejar el proceso de obligación en el aula por un compromiso propio de fortalecer sus habilidades de programación.

4.4. Cronograma de actividades

Teniendo en cuenta las sesiones correspondientes a todo el periodo académico se establecieron en el cronograma las actividades en VPL a partir de la semana 8, en la cual se inicia con la aplicación del instrumento de evaluación de la motivación y posteriormente se empiezan a trabajar tanto en la práctica como en los procesos evaluativos dentro del grupo experimental. En la semana 14 se realiza nuevamente la aplicación del instrumento de evaluación de la motivación y en la semana 15 se realiza la aplicación de la encuesta de percepción.

Se puede observar en la Tabla 4-5 el cronograma de actividades, el cual presenta el desarrollo de la integración y experimentación de la investigación.

Tabla 4-5.: Cronograma de actividades

Sem	Temática	Contenido	Actividad práctica	M1				M2				
				S1	S2	S3	S4	S1	S2	S3	S4	
1	Introducción al curso y a la programación de computadores	Presentación de la asignaturas, contenidos y evaluación de las actividades		X								
		Introducción al pensamiento lógico y algorítmico		X								
2	Principios de algoritmia, resolución de problemas	Introducción a los algoritmos, diseño y resolución de problemas de lógica computacional y algorítmica			X							
	Representación de algoritmos y herramientas para la programación	Representación de los algoritmos mediante diagramas de flujo de datos (DFD)			X							
3	Expresiones, jerarquía de los operadores, IDEs	Diseño y resolución de expresiones aritméticas, relacionales y lógicas, introducción al lenguaje de programación				X						
4	Diagramas de flujo y su importancia	Diagramas de Flujo a través de DFD					X					
5	Diagramas de flujo y su importancia	Diagramas de Flujo a través de DFD						X				
6	Pseudocódigo	Pseudocódigo a través de PSeInt							X			
7	Pseudocódigo	Pseudocódigo a través de PSeInt								X		
8	Introducción a la investigación (motivación en el aprendizaje)	Explicación de la investigación propuesta (Motivación en el aprendizaje y su caracterización)										X
	Aplicación del instrumento MSLQ (Pre-Test)											X
	Introducción a la herramienta de evaluación de código fuente	Explicación de Herramienta VPL/DevCPP (Evaluación desde el estudiantes y desde el docente)	1. Taller: Básico de programación									X

Tabla 4-5.: Continuación de cronograma de actividades

Sem	Temática	Contenido	Actividad práctica	M3				M4				
				S1	S2	S3	S4	S1	S2	S3	S4	
9	Constructores básicos de programación	Variables, Tipos de datos, Entradas y Salidas	1. Quiz: Básico de programación	X								
			2. Taller: Variables entrada/salida	X								
10	Estructuras de control condicionales 1	Sentencias de decisión simple: IF, IF ELSE	2. Quiz: Variables entrada y salida		X							
			3. Taller: Condicional simple		X							
11	Estructuras de control condicionales 2	Sentencias de decisión anidadas IF, ELSEIF	3. Quiz: Condicionales simples			X						
			4. Taller: Condicional anidada			X						
12	Estructuras de control condicionales por casos	Sentencias de decisión múltiple: SWITCH CASE	4. Quiz: Condicionales anidadas				X					
			5. Taller: Switch case				X					
13	Estructuras de control iterativas 1	Sentencias iterativas o por ciclos simple y anidados: FOR	5. Quiz: Switch Case					X				
			6. Taller: Ciclo FOR					X				
14	Estructuras de control iterativas 2	Sentencias iterativas o por ciclos simple y anidados: WHILE, DO WHILE	6. Quiz: Ciclos simples y anidados FOR						X			
			7. Taller: Ciclo WHILE						X			
			Aplicación del instrumento MSLQ (Post-Test)						X			
15	Funciones	Definición e implementación de las funciones dentro de la programación	7. Quiz: Ciclos simples y anidados WHILE							X		
									X			
			Aplicación de la encuesta de percepción sobre VPL						X			
16	Funciones con/sin retorno	Variables globales y locales Construcción de métodos/funciones con y sin retorno									X	

Fuente: Elaboración propia

4.5. Variables a medir

A través de esta investigación se buscó evaluar el impacto en la motivación en el aprendizaje de los estudiantes de programación de computadores a partir de la integración de una herramienta de evaluación automática. Es por esto por lo que una de las variables importantes a analizar es la motivación en el aprendizaje de los estudiantes participantes del proceso. De igual manera, se pretende conocer la percepción de los estudiantes pertenecientes al grupo experimental sobre la integración de herramientas de evaluación automática dentro de la metodología de clase de asignaturas relacionadas a la programación de computadores a su vez de conocer su opinión respecto a la utilización de VPL dentro de las actividades académicas y su proceso de formación en la asignatura de Algoritmos y Programación.

4.5.1. Motivación en el aprendizaje

La motivación es aquel factor que induce o conlleva a una persona a desarrollar alguna acción en concreto. En el contexto académico esto se traduciría en aquellos factores que conducen al estudiante a aprender y a perseverar en el esfuerzo que este requiera para conseguir la meta de aprendizaje propuesta sin importar el tiempo o dedicación que esto conlleve (Ruiz-de-Clavijo., 2009). Cuando se habla de motivación en el aprendizaje se define como aquel interés que adquiere el estudiante en su propio proceso de aprendizaje o por aquellas actividades o temáticas que lo conducen hacia este. Este interés puede mantenerse, aumentar o disminuir en favor de algunos elementos intrínsecos y extrínsecos que intervengan en el proceso. Según Ramírez-Echeverry (2017) la motivación en el aprendizaje se da como un proceso de construcción mental y psicológica que un estudiante realizada de forma personal o individual y puede repercutir en interpretaciones o valoraciones situacionales. De esto se desprende que la autorregulación en el aprendizaje tenga como centro la motivación en el propio proceso del estudiante, es decir, en el proceso de aprender el estudiante es consciente de la importancia que tiene ese aprendizaje y por ende determinará las competencias necesarias para llevar a cabo las metas que se haya propuesto. La intervención de estrategias pedagógicas innovadoras puede llevar a aumentar la motivación y la creencia en el control del aprendizaje que el estudiante percibe de su proceso, puede aumentar el interés en pro de metodologías activas más adecuadas para el aprendizaje que se está intentando transmitir y el interés personal en el desafío que puede provocar unas temáticas planteadas desde metodologías pedagógicas más actuales.

4.5.2. Percepciones sobre la herramienta

Las percepciones recopiladas a través de la encuesta aplicada al grupo experimental expresan las opiniones y sensaciones desde el contexto académico y respecto al proceso de enseñanza-aprendizaje al enfrentarse a VPL y a la integración de metodologías activas con herramientas tecnológicas de evaluación automática de código fuente en las asignaturas relacionadas con la programación de computadores. A partir de estas percepciones por parte de los estudiantes de Ingeniería de Sistemas permitió

conocer y entender la experiencia que obtuvieron al enfrentarse a los distintos retos en la herramienta y sus perspectivas frente al aprendizaje autónomo generado en base a la práctica, la recreación de los ejercicios de práctica, la comprobación de sus soluciones en base a unas respuestas a modo de casos de prueba, la posibilidad de ejecutar el código en el ambiente, todo ello con el fin de poder comprender el impacto que puede generar una herramienta de este tipo en la motivación de su propio aprendizaje y en la construcción de un proceso formativo basado en la integración de herramientas tecnológicas en el campo de la programación.

En múltiples estudios se recomienda aplicar encuestas para determinar la percepción u opinión de intervenciones pedagógicas en el proceso enseñanza-aprendizaje en las clases académicas. Hamidah *et al* (2019) recopilan opiniones mediante un cuestionario de tipo encuesta con el objetivo de determinar el impacto que genera la integración de una metodología de clase de tipo “blended” en la autoeficacia de la escritura en el idioma inglés. Por otro lado, Gonzalez-Escribano *et al* (2019) obtienen comentarios de los alumnos a través de la aplicación de una encuesta con el fin de evaluar el efecto de la motivación a partir de la actividad colaborativa y el aprendizaje subjetivo. Se sugería en el mismo estudio que la aplicación se realizara posterior a una semana de realizada la intervención con el fin de alejar factores emocionales producidas por la finalización de esta. En esta se utilizaron preguntas de tipo escala de Likert con 5 valores (1-5), una de tipo selección múltiple y algunas preguntas con respuesta abierta para determinar observaciones o percepciones generales. Estas indicaciones y sugerencias producidas en dichos estudios se tuvieron presentes al momento de realizar la encuesta de percepción aplicada a los estudiantes de programación de computadores.

4.6. Instrumentos de recolección de datos

En esta sección se presentan los instrumentos de recolección de datos seleccionados y utilizados para la captura y posterior análisis cuantitativo y cualitativo. En un inicio se da conocer el instrumento de medición encargado de recolectar los datos que permitirán caracterizar los diferentes factores motivaciones con el objetivo de determinar si existen cambios significativos en la motivación de los grupos participantes tanto el grupo control como experimental con la intervención. Por otro lado, se presenta la encuesta de percepción que permitirá recopilar las opiniones de los participantes del grupo experimental con el fin de evaluar el impacto en la motivación a partir de la integración realizada.

4.6.1. Motivated Strategies for Learning Questionnaire - Colombia (MSLQ - Colombia)

El Motivated Strategies for Learning Questionnaire (MSLQ) es un instrumento de autorreporte que permite medir la motivación en el aprendizaje (dividida en tres componentes: valor, expectativas, afectivo) y las estrategias de aprendizaje (dividida en tres componentes: cognitivas meta-cognitivas y gestión de recursos) de los participantes de la muestra en el proceso formativo y/o académico a

partir de intervenciones de prácticas educativas ayudando a evaluar los impactos generados en la población seleccionada. Dicho instrumento está originalmente escrito en idioma inglés y cuenta con permisos para uso de dominio público a partir de las políticas expresadas por los autores en sus artículos científicos (Garcia-Duncan y McKeachie., 2005; Pintrich *et al.*, 1991). Sin embargo, se ha traducido a múltiples lenguajes para su uso en el contexto educativo propio de cada país tales como México (Ramírez-Dorantes *et al.*, 2013), Colombia (Ramírez-Echeverry., 2017) o Chile (Burgos-Castillo y Sánchez-Abarca., 2012).

El instrumento de autorreporte MSLQ se ha utilizado ampliamente en diversos estudios por sus propiedades psicométricas de gran confiabilidad y validez analizado a través de diversos autores en contextos distintos, siendo el campo educativo uno de los más apoyados por este instrumento (Ramírez-Echeverry., 2017). Ramírez-Echeverry (2017) en la tesis doctoral realizada en la Universidad Politécnica de Cataluña lleva a cabo una adaptación al español del cuestionario de Motivación en el Aprendizaje y de Estrategias de aprendizaje del MSLQ en el contexto educativo de Colombia. De igual manera, llevó a cabo la validación del instrumento; la validez interna se evaluó a través del método estadístico de análisis factorial exploratorio y para la validez externa esta se dio mediante la validez externa referida al criterio del rendimiento académico, teniendo presente que el proceso de autorregulación en el aprendizaje es predictor de éste, todo esto a partir de los datos obtenidos en las pruebas que se realizaron con los estudiantes de la Universidad Nacional de Colombia. Es por esto que el MSLQ-Colombia, tal como se denominó por el autor presenta un alto grado de confiabilidad, con propiedades psicométricas similares al cuestionario original del MSLQ (Pintrich *et al.*, 1991), brindando una validez representativa en los componentes que permite analizar. Por esto, el instrumento construido por Ramírez-Echeverry (2017) permite realizar una caracterización de la motivación en el aprendizaje y las estrategias de aprendizaje de manera confiable en el contexto educativo colombiano en los estudiantes de ingeniería en universidades de Colombia.

El cuestionario MSLQ-Colombia se compone de 30 ítems que se asocian a cada uno de los aspectos o factores motivacionales en el proceso de aprendizaje y 45 ítems que se relacionan con el uso de estrategias de aprendizaje en el proceso formativo. Para el actual estudio tan solo se hará uso del aspecto motivacional teniendo presente el enfoque, el objetivo y la pregunta problema de la de investigación. El enfoque de la motivación en el aprendizaje del MSLQ-Colombia se compone de 7 sub-escalas. Cada una de estas se establece a partir de una cantidad de ítems planteados en el cuestionario de autorreporte. En el Tabla 4-6 se pueden definir cada una de las sub-escalas planteadas por Ramírez-Echeverry (2017) que componen la escala de motivación en el aprendizaje. El detalle de los ítems del cuestionario relacionados con la escala de motivación del aprendizaje puede consultarse en el Anexo A.

Tabla 4-6.: Sub-escalas de motivación en el aprendizaje del instrumento MSLQ-Colombia

Sub-escala	Nombre	Descripción	Ítems *
S-E1	Valoración de la tarea	Este aspecto motivacional explora el valor que le atribuyen los estudiantes a las tareas o actividades propuestas en su proceso formativo. Esta valoración dependerá de la importancia, interés o utilidad que el estudiante considere que esta tarea le aporta a su formación ya sea personal o profesional. De igual manera el valor de la tarea también se relaciona con que el estudiante encuentre reto, desafío o gusto por las temáticas que se plantean en la asignatura.	Tabla A-1
S-E2	Orientación hacia metas intrínsecas	Esta sub-escala buscan analizar o explorar si las razones que tiene el estudiante para involucrarse con la materia son implícitas al proceso de aprender. Es decir, que el estudiante se motiva por estudiar la asignatura puesto que le genera curiosidad, interés, emoción, gusto o un deseo natural y genuino por aprender más sobre esta, teniendo presente que fin es el aprender las temáticas planteadas.	Tabla A-2
S-E3	Orientación hacia metas extrínsecas	En este aspecto motivacional se busca explorar si las razones por las que el estudiante se motiva a estudiar tienen que ver con factores ajenos y externos al de aprender. Estos factores tales como: buenas notas, becas, monitorias, premios, reconocimientos, recompensas, estatus o reconocimiento social, evaluaciones positivas, participación en equipos universitarios, competencia con los demás compañeros, apoyos económicos pueden ser algunos de las metas que lleven al estudiante a encontrar su motivación principal, puesto que su preocupación al estudiar es alcanzar otro objetivo distinto al del aprender.	Tabla A-3
S-E4	Creencias sobre control del aprendizaje	Esta sub-escala busca explorar si el estudiante entiende, piensa, cree o siente que el aprendizaje en la asignatura va a depender exclusivamente de su propio esfuerzo, dedicación y tiempo y no de otros factores tales como: el docente, las temáticas complicadas, el aula de clase, la metodología de clase o el material de apoyo entre otros.	Tabla A-4

Tabla 4-6.: Continuación de Sub-escalas de motivación en el aprendizaje del instrumento MSLQ-Colombia

Sub-escala	Nombre	Descripción	Ítems *
S-E5	Expectativas de auto-eficacia en el rendimiento	Esta sub-escala explora si el estudiante se cuestiona o si auto-evalúa aspectos tales como su capacidad y su confianza para lograr el éxito esperado en el proceso de evaluación en la materia y sus temáticas.	Tabla A-5
S-E6	Expectativas de auto-eficacia en el aprendizaje	Esta sub-escala pretende conocer la percepción que tiene el estudiante sobre la probabilidad de éxito que tendrá en su proceso de aprendizaje o la capacidad para comprender las diferentes temáticas abordadas en la asignatura. Dicha percepción de éxito depende de las creencias que tiene el estudiante sobre sus propias capacidades y la confianza que se tiene para lograr el objetivo de aprender.	Tabla A-6
S-E7	Ansiedad en los procesos de evaluación	En esta sub-escala se busca conocer si el estudiante experimenta nervios, preocupación, angustia o inquietud al momento de estar frente a un proceso de evaluación tal como: quices, tareas, talleres o parciales. De igual manera se busca explorar y analizar si el estudiante presenta pensamientos negativos que puedan llegar a afectar desfavorablemente su desempeño o rendimiento académico.	Tabla A-7

* En el Anexo A se encuentran descritos todos los ítems del cuestionario relacionados con la escala de motivación del aprendizaje para cada sub-escala

Fuente: Elaboración propia

El instrumento MSLQ-Colombia estuvo disponible desde dos formularios de Google Forms, uno para el Pre-Test (<https://forms.gle/6uHU6BW32t4EdGCUA>) y otro para el Post-Test (<https://forms.gle/DKDHPXh2hntzzYEu8>), esto con el fin de tener un mayor orden, control sobre los tiempos de aplicación y organización sobre la agrupación de los datos para ambos grupos participantes. Al inicio del formulario se presenta una sección donde se trasmite el siguiente consentimiento informado y acuerdo de confidencialidad:

“Yo hago constar que he sido informado(a) con claridad respecto al ejercicio académico en el que Hernán Darío Lozano Rojas, investigador en proyecto de investigación de la maestría en Ingeniería de Sistemas y Computación de la Universidad Nacional de Colombia en la línea de Educación en Ingeniería, me ha invitado a participar; que actúo consecuente, libre y voluntariamente como colaborador, contribuyendo a esta investigación de forma activa. Soy conocedor(a) de la autonomía que poseo para retirarme de esta investigación cuando lo estime conveniente y sin necesidad de justificación alguna y que no se trata de una intervención con fines de tratamiento psicológico. Asimismo, se me informó que en todo momento será respetada la buena fe y la confidencialidad de los datos por mí suministrados, lo mismo que mi seguridad física y psicológica.”

(Consentimiento informado y acuerdo de confidencialidad)

Posteriormente, se obtienen algunos datos adicionales a partir de preguntas de tipo demográficas con el fin de caracterizar a los estudiantes. Por último, se exponen los ítems del cuestionario MSLQ-Colombia (Anexo A) en donde el estudiante responderá a consideración propia y bajo su punto de vista cada una de las afirmaciones, a partir de una escala de Likert con las siguientes 7 opciones:

1	2	3	4	5	6	7
Totalmente no me describe	No me describe	Algo no me describe	Ni no me describe ni me describe	Algo me describe	Me describe	Totalmente me describe

Figura 4-2.: Opciones de la escala de Likert del cuestionario MSLQ-Colombia

Con el fin de proceder al análisis de los datos que se recopilaron a través del cuestionario de MSLQ-Colombia se utilizó una prueba estadística con la que se pudo determinar si existió un cambio significativo en los factores de la motivación en el aprendizaje mencionados previamente. Para este tipo de estudios de tipo investigación cuasiexperimental es fundamental que las pruebas estadísticas ayuden a verificar si la hipótesis establecida está soportada por los datos. A partir del procedimiento que se halla determinado para validar los resultados obtenidos a lo largo del experimento, la prueba estadística será importante para determinar si se acepta o rechaza la hipótesis planteada. Por un lado se le determina hipótesis nula (H_0) a aquella hipótesis que “no tiene efecto o impacto” y, por lo general, se establece con el propósito puntual de ser rechazada, y por ende, si ésta se rechaza se está aceptando la hipótesis alterna (H_1) (Siegel y Castellan., 1998).

A partir de las variables a medir en la investigación se realizó el planteamiento de las hipótesis propuestas. Por un lado, como variable independiente se determinó la integración de la herramienta de evaluación automática de código fuente en las sesiones de clase de programación de computadores, y, por otro lado, como variable dependiente se identifica el impacto en la motivación del aprendizaje de los estudiantes a partir de aquella integración. Por ende, como hipótesis nula se determina la siguiente:

H_0 : “La herramienta de evaluación automática de código fuente no genera ningún impacto en la motivación en el aprendizaje de los estudiantes de la asignatura de programación de computadores”

Por lo cual, como hipótesis alterna se plantea la siguiente:

H_1 : “La herramienta de evaluación automática de código fuente genera un impacto en la motivación en el aprendizaje de los estudiantes de la asignatura de programación de computadores”

Al tener presente que el tamaño de la muestra de la investigación no es muy grande, se determina el uso de una prueba estadística paramétrica para una muestra pequeña. Las pruebas estadística no paramétricas son muy utilizadas puesto que en relación con los datos la cantidad de suposiciones realizadas suele ser menor (Siegel y Castellan., 1998). De igual manera, estas pruebas resultan ser convenientes cuando se desea tratar datos categóricos, los cuales son aquellos que se miden bajo una escala nominal.

Con este tipo de pruebas estadísticas se brindan herramientas para que el investigador o analista pueda determinar si algunos de los participantes o estudiantes cuentan con mayor o menor cantidad de las características que se estén analizando. Teniendo en cuenta que se desea comprender cuál es el impacto de una herramienta de evaluación automática de código fuente en la motivación en el aprendizaje de los estudiantes de programación de computadores es fundamental realizar dos mediciones (Pre-Test y Post-Test) del instrumento de medición de la motivación en el aprendizaje seleccionado, antes y después de la integración de la herramienta, tanto en el grupo experimental como en el grupo control.

La prueba de los rangos con signo de Wilcoxon es una prueba no paramétrica para comparar el rango medio de dos muestras relacionadas y establecer si existen diferencias significativas entre estas, determinando si la diferencia se debe a efectos de azar o no (en este caso, que la diferencia sea estadísticamente significativa). Se utiliza como opción o alternativa a la prueba *t-student* cuando no se logra determinar o suponer la normalidad de las muestras (Wilcoxon., 1945). En la prueba de Wilcoxon se asocia mayor peso a los pares que evidencian mayores diferencias entre las dos condiciones existentes, más que aquellos pares en las cuales la diferencia sea más pequeña (Siegel y Castellan., 1998).

Ahora bien, es necesario determinar el nivel de significancia (α) para brindar validez al resultado obtenido mediante la prueba estadística. Se determina como nivel de significancia (α) aquel valor que permitirá rechazar la hipótesis nula o aceptar la hipótesis alterna (Coolican., 1997). En la práctica más convencional, el nivel de significancia es de 0,05. Se puede identificar el resultado obtenido a partir del nivel de significancia de la siguiente forma:

- Si un resultado obtenido es significativo ($p \leq 0,05$), la hipótesis nula es rechazada.
- Si el resultado obtenido no es significativo ($p > 0,05$), la hipótesis nula se mantiene.

Para este estudio se determinó utilizar un nivel de significancia (α) del 0,05 en el cual es como se miden y validan las diferencias significativas en los estudios de este tipo de muestras.

4.6.2. Encuesta de percepción sobre la herramienta VPL

En cuanto a la recopilación de los datos para el análisis cualitativo se llevó a cabo mediante la aplicación de una encuesta de percepción realizada para conocer de manera detallada las opiniones y experiencias que los estudiantes participantes del grupo experimental presentaron al enfrentarse al uso de la herramienta de evaluación automática VPL dentro de la metodología de la asignatura de programación de computadores. De igual forma, se plantearon preguntas asociadas a las sub-escalas de la motivación en el aprendizaje brindadas por el MSLQ-Colombia, con el fin de poder complementar por medio de un análisis cualitativo, la información de tipo cuantitativo recopilada por el MSLQ-Colombia. La Tabla 4-7 muestra las preguntas de la encuesta de percepción.

Tabla 4-7.: Encuesta de percepción sobre la herramienta VPL

Pre-gunta	Enunciado	Tipo de respuesta	Sub-escala relacionada
1	Los ejercicios desarrollados en la herramienta VPL dentro de la asignatura me permitieron descubrir el gusto por los componentes temáticos de la asignatura de programación de computadores.	Likert	Orientación hacia metas intrínsecas
2	¿Por qué seleccionaste la respuesta anterior?	Abierta	
3	Considero que el uso de la herramienta VPL en la asignatura de programación de computadores me generó desafíos que despertaron en mí el deseo de aprender a programar.	Likert	Valoración de la tarea
4	¿Por qué seleccionaste la respuesta anterior?	Abierta	
5	El uso de la herramienta VPL en la asignatura de programación de computadores resultó útil para alcanzar mi meta más importante: obtener buenas calificaciones en la asignatura.	Likert	Orientación hacia metas extrínsecas
6	¿Por qué seleccionaste la respuesta anterior?	Abierta	
7	Usar VPL en la asignatura de programación de computadores aumentó mi confianza y seguridad para entender la temática y conceptos más complejos que se enseñan en esta asignatura.	Likert	Expectativas de autoeficacia para el aprendizaje
8	¿Por qué seleccionaste la respuesta anterior?	Abierta	
9	Usar VPL en la asignatura de programación de computadores aumentó mi confianza para la obtención de buenas calificaciones.	Likert	Expectativas de autoeficacia para el rendimiento
10	¿Por qué seleccionaste la respuesta anterior?	Abierta	
11	Considero que la utilización de VPL me ayudó a entender que el nivel de comprensión de las temáticas planteadas en la asignatura depende de mi esfuerzo y trabajo.	Likert	Creencias de control del aprendizaje
12	¿Por qué seleccionaste la respuesta anterior?	Abierta	

Tabla 4-7.: Continuación de Encuesta de Percepción sobre la herramienta VPL

Pregunta	Enunciado	Tipo de respuesta	Sub-escala relacionada
13	Considero que el uso de VPL en la asignatura me ayudó a mejorar mi confianza y seguridad para manejar la ansiedad y preocupación que siento cuando presento las evaluaciones de la asignatura.	Likert	Ansiedad frente a los procesos de evaluación
14	¿Por qué seleccionaste la respuesta anterior?	Abierta	
15	De los siguientes elementos que apoya VPL, ¿Cuál/es te gustó/aron más?	Opción múltiple con múltiple respuesta	Pregunta de percepción personal (no es sub-escala)
16	¿Cuál fue la percepción general acerca del uso de la herramienta VPL en la metodología de la clase?	Abierta	

Fuente: Elaboración propia

La encuesta de percepción y las preguntas que se encuentran relacionadas allí se elaboraron en consenso con el director y codirector de la tesis con el fin de obtener mayor cantidad de datos e insumos suficientes para contestar la pregunta de investigación planteada. La encuesta de percepción se realizó a partir de un formulario de Google Forms el cual se puede encontrar en el siguiente enlace: <https://forms.gle/FcRmn4JvzoNJWpFD8>.

Al empezar, se presenta el objetivo de la encuesta de percepción y el acuerdo de confidencialidad indicando que no tendrá repercusiones ni afectaciones al proceso académico, por lo que pueden contestar con total libertad al respecto, adicional a alguna información con datos personales para relacionar las respuestas con las anteriormente recolectadas en los datos cuantitativos. La introducción a la encuesta de percepción se puede ver a continuación:

“Apreciado estudiante: La información que te vamos a consultar se refiere a tus percepciones con respecto al uso de VPL en la clase de algoritmos y la metodología aplicada a la asignatura. Dado lo anterior, los docentes de esta asignatura estudiaremos detenidamente las respuestas que nos compartas con el fin de diseñar estrategias que favorezcan tus procesos de aprendizaje y los de futuros estudiantes. Debido a que no hay respuestas correctas ni incorrectas, te pedimos responder esta encuesta sin ningún tipo de prevención y tan preciso como te sea posible. La información que nos brindarás será tratada con total confidencialidad, no es calificable, ni influirá en ningún proceso académico que adelantes en la Universidad. La información personal será borrada al finalizar esta investigación. ¡Gracias por tu tiempo y por tus respuestas!”

(Consentimiento informado y acuerdo de confidencialidad)

Las 14 preguntas iniciales están relacionadas con las sub-escalas de la motivación en el aprendizaje evaluadas a través del MSLQ-Colombia. Estas están compuestas por 7 preguntas de tipo Likert con

las opciones que se pueden apreciar en la Figura 4-3:

1	2	3	4	5	6
Totalmente en desacuerdo	En desacuerdo	Algo en desacuerdo	Algo de acuerdo	De acuerdo	Totalmente de acuerdo

Figura 4-3.: Opciones de la escala de Likert de la encuesta de percepción sobre VPL

Las otras 7 preguntas seguidas a estas son de respuesta abierta, donde se les solicita a los estudiantes justificar la selección de cada respuesta dada en la pregunta inmediatamente anterior. Posteriormente, se presenta una pregunta donde se cuestiona al estudiante sobre las características de la herramienta de evaluación automática VPL que les resultó más útil o le encontraron mayor gusto en su proceso de formación, la pregunta es de múltiples respuestas y con 12 opciones posibles las cuales son:

- El entorno de desarrollo integrado.
- La posibilidad de ejecutar el código en la misma herramienta.
- El autocompletado de código fuente, tanto en variables como en estructuras de control.
- La ayuda en la sintaxis que brinda la herramienta a la hora de evaluar.
- La cantidad ilimitada de intentos para evaluar en cada uno de los ejercicios.
- La posibilidad de verificar errores al momento de evaluar el código.
- La retroalimentación inmediata que brinda la herramienta cuando el código no cumple con los casos esperados de prueba.
- La evaluación automática del código fuente.
- La posibilidad de ver los errores cometidos en el programa.
- La ayuda en la visualización del ejercicio (condiciones de entrada y salida) dentro del entorno de desarrollo.
- La posibilidad de guardar el código fuente y continuarlo posteriormente.
- La posibilidad de practicar con los diferentes ejercicios de práctica que estaban en la plataforma.

Por último, se expone una pregunta abierta en la que el estudiante brinde su percepción general sobre la herramienta VPL y su integración en los contenidos de la asignatura de programación de computadores.

Para llevar a cabo el análisis de los datos obtenidos con la encuesta de percepción se procede a realizar el análisis cualitativo a través de la técnica de codificación, fundamental para cualquier análisis de este tipo. La codificación es aquel proceso en el cual se segmentan y se reorganizan los datos recolectados por medio de códigos o categorías que sustentan el trabajo interpretativo con los datos. Estos códigos permiten organizar los datos y la información con el fin de examinar y analizar de forma estructurada la relación que existe entre los códigos y/o categorías, Estos códigos se abordan como indicadores de conceptos que permitirá comparar por el investigador y encontrar relaciones entre cada uno de estos (Hernández Sampieri *et al.*, 2016).

Teniendo en cuenta la definición de codificación abierta dada por Bryman (2012), en la cual se descomponen, examinan, comparan, conceptualizan y categorizan los datos, en la investigación se decide utilizar dicha técnica. Al hacer uso de esta codificación se obtienen conceptos, que al agruparlos eventualmente se transformaran en categorías. En esta etapa de la investigación se identifica este proceso como un proceso iterativo y en el cual se requiere la participación de dos o más investigadores con el objetivo de tener una mejor categorización de los datos analizados. Para este proceso se realizó el acompañamiento en múltiples sesiones con el director y codirector los cuales a través de varias revisiones minuciosas fueron apoyando a refinar cada una de las categorías y conceptos correspondientes.

Esta codificación pretende brindar conceptos que tengan alguna relación con los fenómenos o eventos evidenciados a lo largo del experimento. De igual manera, busca categorizar cada uno de los conceptos que se han hallado y, en algunos casos, generar categorías centrales que permitan agrupar de manera más general algunas que compartan relaciones específicas. Este último proceso es conocido como codificación axial.

5. Resultados experimentales

Antes de iniciar con los resultados, es importante tener presente el uso considerable que se le dio a la herramienta VPL por parte de los estudiantes a lo largo de toda la intervención. En la Tabla 5-1 se pueden apreciar las 14 actividades realizadas por el grupo experimental, en la herramienta VPL, las cuales reunieron 1256 intentos de solución por parte de los estudiantes (entregas), lo que indica que en promedio se realizaron cerca de 88 entregas por actividad. De esa forma, se puede determinar que aproximadamente cada entrega contó con un promedio de 4,67 entregas por actividad. Las actividades que más tuvieron estudiantes que acertaron la actividad puntual fueron: el quiz 5, el taller final con una participación y aciertos del 100%. Las actividades con menos aprobaciones fueron el taller 2 con un total de 12 personas aprobadas y el taller 3 con un total de 15, sin embargo, se debe mencionar que en éstas la participación no fue completa puesto que la cantidad de estudiantes que realizaron envíos fueron 14 y 19, respectivamente.

Tabla 5-1.: Número de estudiantes que participaron, aprobaron las actividades, el número de entregas por actividad y el promedio de entregas por estudiante en cada actividad para el curso Programación de Computadores

Actividades	Estudiantes	Estudiantes aprobados	# Entregas	Entregas por estudiante
1. Taller: Básico de programación	21	19	151	7,19
1. Quiz: Básico de programación	18	15	99	5,50
2. Taller: Variables entrada/salida	14	12	101	7,21
2. Quiz: Variables entrada y salida	21	20	97	4,62
3. Taller: Condicional simple	19	15	122	6,42
3. Quiz: Condicionales simples	19	16	87	4,58
4. Taller: Condicional anidada	21	19	89	4,24
4. Quiz: Condicionales anidadas	20	19	98	4,90
5. Taller: Switch case	20	20	97	4,85
5. Quiz: Switch Case	21	21	67	3,19
6. Taller: Ciclo FOR	19	17	78	4,11
6. Quiz: Ciclos simples y anidados FOR	19	18	63	3,32
7. Taller: Ciclo WHILE	19	17	58	3,05
7. Quiz: Ciclos simples y anidados WHILE	21	20	49	2,33

Los valores están en unidades y el promedio se toma a partir de la cantidad de estudiantes que presentaron el ejercicio

Se puede evidenciar de igual manera que existe una mejora considerable en la cantidad de entregas por estudiantes respecto a la aprobación, indicando así que entre más realizaba un uso considerable de la herramienta VPL, menos intentos requerían para culminar satisfactoriamente cada una de las actividades propuestas, disminuyendo la cantidad necesaria de entregas y aumentando los estudiantes que aprobaban cada actividad.

En cuanto al análisis de los resultados, los resultados cuantitativos obtenidos en el cuestionario MSLQ-Colombia tanto del grupo control y experimental con sus respectivas medidas Pre-Test y Post-Test fueron de suma importancia para determinar si hubo un efecto significativo en el aprendizaje respecto a la motivación en estudiantes de ingeniería de telecomunicaciones e ingeniería de sistemas. Cabe mencionar que en esta investigación solo se aplicó del cuestionario MSLQ-Colombia la escala de motivación en el aprendizaje con sus correspondientes sub-escalas. De igual manera, los estudiantes realizaron una encuesta sobre la percepción mediante la cual se pudo analizar las opiniones de estos mismos frente al uso de VPL en las sesiones de clase de programación de computadores. En esta sección se presentan los resultados finales obtenidos de la intervención.

5.1. Motivación en el aprendizaje

La motivación de los estudiantes fue analizada por medio del Pre-Test y Post-Test aplicando el instrumento MSLQ-Colombia en cada grupo participante: grupo control y grupo experimental. El propósito fue analizar si hubo algún cambio significativo en alguna de las sub-escalas motivacionales estudiadas por el MSLQ-Colombia en cada uno de los grupos nombrados previamente. La hipótesis que se planteó fue el aumento en los niveles de motivación en el grupo seleccionado como experimental a través del uso de una herramienta de evaluación automática integrada en la metodología de clase de programación de computadores. De llegar a haber alguna diferencia significativa entre el Pre-Test con el Post-Test de dicho grupo, evidenciaría el impacto en la motivación de los estudiantes al integrarse una herramienta de este estilo.

5.1.1. Grupo control - Ingeniería de Telecomunicaciones

Los datos obtenidos del cuestionario MSLQ-Colombia en el grupo control se obtuvieron a través de la participación voluntaria de 19 estudiantes pertenecientes a la Facultad de Ingeniería de Telecomunicaciones. El 23.7 % eran mujeres y el 76.3 % hombres. El 52.6 % de los participantes egresaron de un colegio público y 47.4 % de un colegio privado.

En la Figura 5-1 se observa un diagrama con las respuestas obtenidas con el MSLQ-Colombia en el Pre-Test de la facultad de telecomunicaciones (grupo de control). En el eje Y se presentan cada una de las sub-escalas pertenecientes a la escala de motivación del cuestionario MSLQ-Colombia y en el X el número de respuestas obtenidas en cada opción de la escala tipo Likert, siendo 1 el valor donde

los estudiantes expresaban “Totalmente no me describe”, y 7 donde manifestaban “Totalmente me describe” respecto a cada enunciado. Las sub-escalas motivacionales del eje Y corresponden a las mencionadas en la Sección 4.6.1: valoración de la tarea (S-E1), orientación hacia metas intrínsecas (S-E2), orientación hacia metas extrínsecas (S-E3), creencias sobre el control del aprendizaje (S-E4), expectativas de autoeficacia en el rendimiento (S-E5), expectativas de autoeficacia en el aprendizaje (S-E6), y por último, ansiedad en los procesos de evaluación (S-E7).

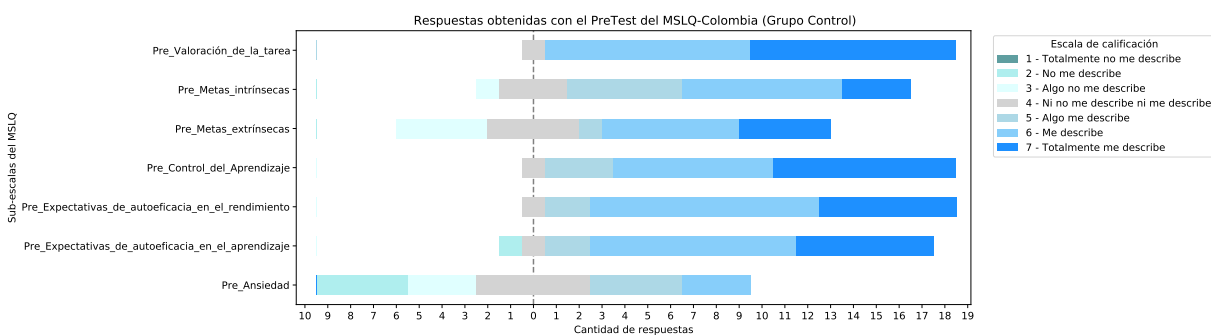


Figura 5-1.: Respuestas obtenidas con el MSLQ-Colombia en el Pre-Test del grupo control del Programa de Ingeniería de Telecomunicaciones

A través de la figura se visualiza que en las sub-escalas motivaciones 1, 4 y 5, los estudiantes generalmente se sentían identificados con las valoraciones más altas de la escala Likert, donde las afirmaciones establecidas los describe de ligera a completamente. En la sub-escalas 2, 3 y 6 relacionadas con la orientación hacia metas intrínsecas, la orientación hacia metas extrínsecas y las expectativas de autoeficacia en el aprendizaje se visualiza que las respuestas estuvieron divididas entre sentirse identificados y no sentirse descrito o ligeramente identificado con respecto a estas sub-escalas. Sin embargo, resaltamos el hecho de que la mayoría de los estudiantes en las tres escalas anteriormente mencionadas se sienten mayormente identificados con los enunciados presentados. Finalmente, se pudo evidenciar que la mitad de los estudiantes no se sintieron descritos o identificados con los enunciados presentados en la sub-escala 7 que está relacionada con la ansiedad frente a los procesos evaluativos.

En la Figura 5-2 se evidencian los datos recolectados por el MSLQ-Colombia en el Post-Test del grupo de control de la Facultad de Telecomunicaciones. En esta figura se puede visualizar un diagrama con las respuestas recolectadas en cada una de las sub-escalas motivacionales analizadas.

Analizando las respuestas obtenidas se puede evidenciar un efecto similar al recopilado en el Pre-Test en este mismo grupo. Las sub-escalas 1, 4 y 5 presentan un comportamiento semejante al observado en los datos del Pre-Test donde los estudiantes tienden a valoraciones donde se sienten de ligera a completamente identificados. De igual manera, las escalas 2, 3 y 6 presentan mayormente respuestas donde los estudiantes se encuentran de ligera a completamente identificados, similar a lo que se evidenció en el Pre-Test en estas mismas sub-escalas. Finalmente, podemos observar que en la sub-

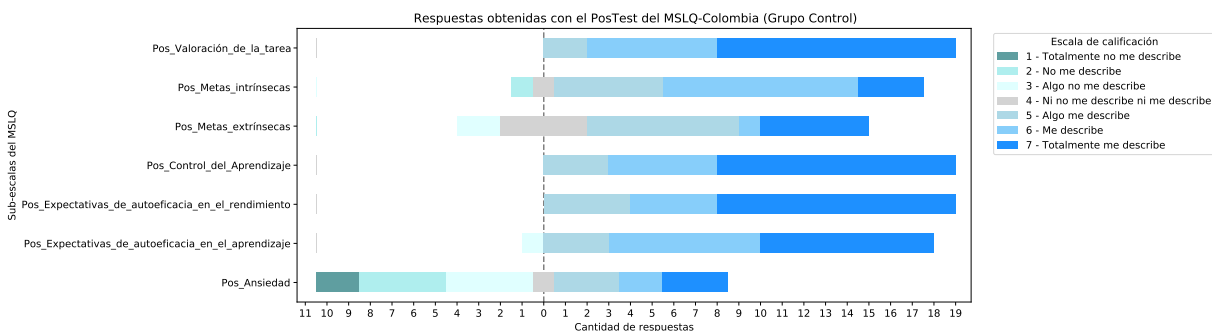


Figura 5-2.: Respuestas obtenidas con el MSLQ-Colombia en el Pos-Test del grupo control del Programa de Ingeniería de Telecomunicaciones

escala 7 relacionada con la ansiedad en procesos evaluativos hay un comportamiento similar que en el pre-test, aunque en este caso los estudiantes mostraron mayormente no sentirse identificados con esto.

Sin embargo, para entender si hubo o no una variación significativa entre los datos del Pre-Test y los del Post-Test del grupo control fue necesario dirigirse a los datos cuantitativos aplicando una prueba estadística que permitiera confirmar o rechazar que el cambio evidenciado fue dado a partir de factores reales y no a factores del azar. Las pruebas estadísticas no paramétricas son utilizadas por lo general cuando el tamaño de la muestra poblacional es pequeño o reducido, por lo que este caso específico se adecua teniendo presente que la muestra fue de 19 estudiantes.

Por lo anterior, se llevó a cabo la prueba estadística no paramétrica denominada Wilcoxon, la cual permite realizar diseños con dos mediciones, comúnmente denominada Pre-Test y Post-test, los cuales participan los mismos grupos seleccionados en las dos mediciones, denominados grupos correlacionados (Pagano., 2006). Wilcoxon es similar a la prueba t para muestras relacionadas solo que esta es una prueba paramétrica que cuenta con características que difieren de la utilizada en este estudio tales como: la muestra de los datos estará normalmente distribuida, los participantes serán seleccionados aleatoriamente (García Méndez y Rivera Aragón., 2012). Teniendo en cuenta lo anterior, tanto por el tamaño reducido de la muestra como por la selección aleatoria, ya que dichos estudiantes fueron seleccionados a conveniencia, se decidió hacer uso de la prueba de Wilcoxon con el propósito de determinar y analizar si existieron diferencias significativas en lo que corresponde a la motivación en el aprendizaje mediante la integración del instrumento MSLQ.

A partir de la Tabla 5-2 es posible determinar si hubo o no una diferencia significativa en el grupo control de ingeniería de telecomunicaciones, teniendo en cuenta los datos tanto en Pre-Test como Post-test. Para ello se realizó una serie de análisis estadísticos donde se evidencian los resultados en el grupo de control. En cada sub-escala se presentan los resultados promedios tanto de Pre-Test como Post-test, la diferencia entre estos, la desviación estándar y el *p-valor* de la prueba no paramétrica

Tabla 5-2.: Resultados estadísticos del Pre-test y Post-test del grupo control - Ingeniería de Telecomunicaciones

Sub-escala de motivación	Prueba	Promedio	Diferencia	Desviación Estándar	Wilcoxon p-valor
1 - Valoración de la tarea	Pre-Test	6,3421	0,1228	0,83236	0.478
	Post-Test	6,4649		0,68848	
2 - Orientación hacia metas intrínsecas	Pre-Test	5,5088	0,0701	1,09639	0.347
	Post-Test	5,5789		1,09906	
3 - Orientación hacia metas extrínsecas	Pre-Test	4,9342	0,0658	1,46923	0.742
	Post-Test	5,0000		1,38444	
4 - Expectativas de autoeficacia en el aprendizaje	Pre-Test	5,7368	0,2369	1,22041	0.138
	Post-Test	5,9737		1,06032	
5 - Expectativas de autoeficacia en el rendimiento	Pre-Test	5,9474	0,3026	0,81848	0.114
	Post-Test	6,2500		0,76376	
6 - Creencias sobre control del aprendizaje	Pre-Test	6,0395	0,2368	0,79172	0.171
	Post-Test	6,2763		0,79011	
7 - Ansiedad en los procesos de evaluación	Pre-Test	3,9158	-0,0421	1,31709	0.695
	Post-Test	3,8737		1,93587	

*p-valor < 0.05: Resultado estadísticamente significativo. No hubo diferencias significativas.

Wilcoxon, la cual determinó si hubo o no un cambio significativo; es importante resaltar que esta última prueba se realizó, ya que el tamaño muestral era menor a 50 y no hubo una distribución normal de los datos de cada sub-escala. Teniendo en cuenta lo anterior, se pudo observar que no hubo un efecto significativo en ninguna de las sub-escalas, después de la aplicación de las clases respectivas.

En el Pre-Test se pudo observar que el 24,8 % de las valoraciones se ubicaron en la opción más alta de la escala Likert, donde los estudiantes se sentían totalmente identificados con las afirmaciones propuestas en el cuestionario. El 39,1 % de las valoraciones se ubicaron en la segunda opción más alta de la escala Likert, donde los estudiantes se sentían identificados con los enunciados propuestos por el cuestionario en esta sub-escala de motivación. Adicionalmente, el 14,28 % de las respuestas obtenidas estuvieron orientadas a sentirse ligeramente identificados con las afirmaciones propuestas. Los estudiantes se sentían altamente identificados con las valoraciones más positivas de las opciones de la escala de Likert propuestas. El 78,1 % de las respuestas fueron positivas, demostrando la importancia que otorgaron los estudiantes a los factores motivacionales relacionados con el aprendizaje.

En el Post-Test se evidenció que el 15 % de las respuestas se relacionaron con las escalas neutrales o las que ligeramente no los describen. En comparación con el Pre-Test, las valoraciones de las escalas neutrales o negativas disminuyeron en 6 %, indicando que los estudiantes con el pasar del tiempo podrían haber aumentado su interés en la asignatura, y por ende, la motivación que está representa en su proceso académico.

Con relación a estos resultados, no se pudo rechazar la hipótesis nula, es decir, durante el tiempo del experimento los estudiantes del grupo de control de Ingeniería de Telecomunicaciones no presentaron cambios significativos en las sub-escalas relacionadas con los factores motivacionales estudiados.

Esto era lo esperado.

5.1.2. Grupo experimental - Ingeniería de Sistemas

Los datos obtenidos del cuestionario MSLQ-Colombia en el grupo experimental se obtuvieron a través de la participación voluntaria de 21 estudiantes pertenecientes a la Facultad de Ingeniería y del programa de Ingeniería de Sistemas. El 14.3 % de los participantes fueron mujeres y el 85.7 % hombres. El 73.8 % de los participantes egresaron de un colegio público y 26.2 % de un colegio privado.

En la Figura 5-3 se presentan las respuestas obtenidas en el MSLQ-Colombia en el Pre-Test del grupo experimental correspondiente al programa de Ingeniería de Sistemas. En el eje Y se relacionaron las sub-escalas analizadas por el MSLQ-Colombia y en el eje X el número de respuestas que obtuvo cada opción de la escala Likert planteada. Las sub-escalas establecidas en el eje Y siguen el orden definido en la Sección 4.6.1.

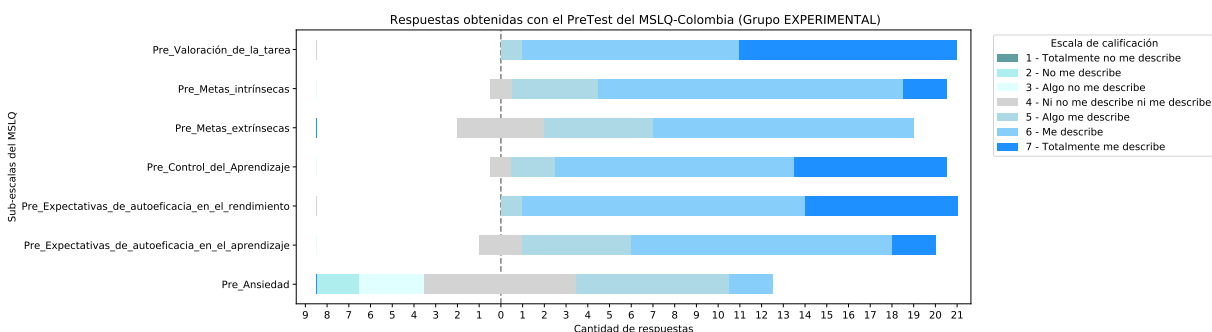


Figura 5-3.: Respuestas obtenidas con el MSLQ-Colombia en el Pre-Test del grupo experimental del Programa de Ingeniería de Sistemas

En la figura se observa un comportamiento parecido en las 6 primeras sub-escalas de la motivación, en donde los estudiantes generalmente se sentían de ligera a completamente identificados con las afirmaciones planteadas en estas sub-escalas. En la sub-escala 1, 4 y 5 se observa una mayor cantidad de estudiantes que se identificaban completamente con las afirmaciones planteadas por esta sub-escala. Sin embargo, en la escala 7 los estudiantes manifestaron no sentirse ligeramente identificados con los enunciados en esta escala, la cual se relaciona con la ansiedad, lo que nos indica que la mayoría de los estudiantes no presentan ningún tipo de temor o ansiedad en los procesos de evaluación al iniciar su formación por medio de la integración de VPL.

Con los datos obtenidos mediante el cuestionario del Post-Test del grupo experimental perteneciente al programa de Ingeniería de Sistemas se observó un efecto similar. En la Figura 5-4 se evidencia el número de respuestas para cada opción de la escala Likert en cada una de las sub-escalas estudiadas.

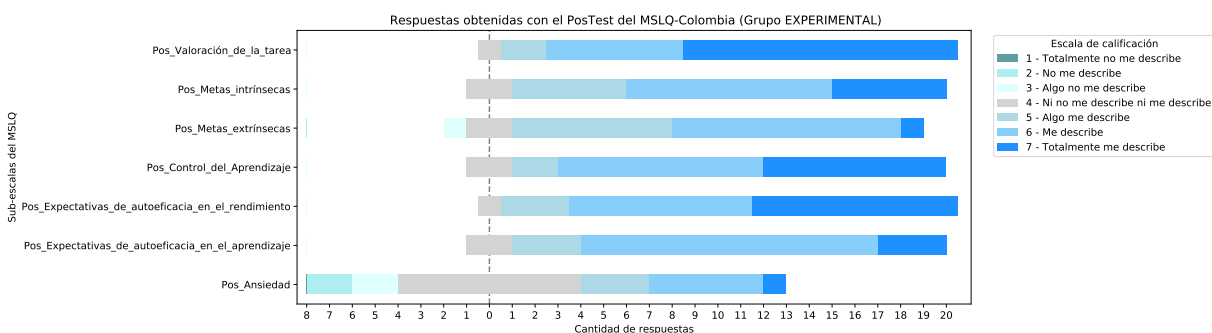


Figura 5-4.: Respuestas obtenidas con el MSLQ-Colombia en el Pos-Test del grupo experimental del Programa de Ingeniería de Sistemas

Visualmente se observa que los estudiantes se sentían generalmente de ligera a completamente identificados con las afirmaciones planteadas en las 6 primeras sub-escalas de la motivación. Sin embargo, se observa un cambio en la escala 3 relacionada con la orientación hacia metas extrínsecas puesto que algunos estudiantes manifestaron no sentirse ligeramente identificados con éstas, en comparación con el pre-test. De igual forma, se puede notar mediante la gráfica que los niveles de ansiedad se han mantenido estables tanto en el inicio como al finalizar la intervención de VPL. Esto teniendo presente, que el 57 % de los estudiantes indica no sentir preocupación o ansiedad respecto a los procesos evaluativos de la asignatura.

A partir de la Tabla 5-3 es posible determinar si hubo o no una diferencia significativa en el grupo experimental de ingeniería de sistemas, teniendo en cuenta los datos tanto en Pre-Test como Pos-test. Para esto se hizo una serie de análisis estadísticos de cada sub-escala, entre ellos, los promedios, diferencias, desviación estándar y el p-valor de la prueba no paramétrica Wilcoxon, la cual determinó si hubo o no un efecto significativo. Es importante aclarar que esta última prueba se realizó ya que el tamaño muestral era menor a 50 y no hubo una distribución normal de cada sub-escala.

Se pudo observar que no hubo un efecto significativo en ninguna de las sub-escalas, después de la intervención propuesta. Sin embargo, se observa una ligera mejora en las sub-escalas relacionadas con valoración de la tarea y creencias sobre control del aprendizaje, siendo éstas las que arrojaron un resultado en el test estadístico más cercano al resultado estadísticamente significativo ($p\text{-valor} < 0.05$).

En el Pre-Test se pudo observar que el 19 % de las valoraciones se ubicaron en la opción más alta de la escala Likert, donde los estudiantes se sentían totalmente identificados con las afirmaciones propuestas en el cuestionario. El 50 % de las valoraciones se ubicaron en la segunda opción más alta de la escala Likert, donde los estudiantes se sentían identificados con los enunciados propuestos por el cuestionario en esta sub-escala de motivación. Adicionalmente, el 17 % de las respuestas obtenidas estuvieron orientadas a sentirse ligeramente identificados con las afirmaciones propuestas. Los estudiantes se sentían altamente identificados con las valoraciones más positivas de las opciones Likert

Tabla 5-3.: Resultados estadísticos del Pre-test y Post-test del grupo experimental - Ingeniería de Sistemas

Sub-escala de motivación	Prueba	Promedio	Diferencia	Desviación Estándar	Wilcoxon p-valor
1 - Valoración de la tarea	Pre-Test	6,4683	-0,1985	0,49052	0,180
	Post-Test	6,2698		0,69845	
2 - Orientación hacia metas intrínsecas	Pre-Test	5,7460	0,1111	0,69845	0.465
	Post-Test	5,8571		0,82038	
3 - Orientación hacia metas extrínsecas	Pre-Test	5,2619	-0,0714	0,86051	0.641
	Post-Test	5,1905		0,89758	
4 - Expectativas de autoeficacia en el aprendizaje	Pre-Test	5,5833	0,1667	0,75139	0.334
	Post-Test	5,7500		0,66615	
5 - Expectativas de autoeficacia en el rendimiento	Pre-Test	6,1429	-0,0715	0,59985	0.979
	Post-Test	6,0714		0,86293	
6 - Creencias sobre control del aprendizaje	Pre-Test	6,0476	-0,0119	0,78110	0.175
	Post-Test	6,0357		0,91953	
7 - Ansiedad en los procesos de evaluación	Pre-Test	4,2381	0,1333	0,99723	0.542
	Post-Test	4,3714		1,39898	

*p-valor < 0.05: Resultado estadísticamente significativo. No hubo diferencias significativas.

propuestas. El 86 % de las respuestas fueron positivas, demostrando la importancia que otorgaron los estudiantes a los factores motivacionales relacionados con el aprendizaje.

Por otra parte, en el Post-Test se evidenció que el 15,7 % de las respuestas se relacionaron con las escalas neutrales o las que ligeramente no los describen. En comparación con el Pre-Test, las valoraciones de las escalas neutrales o negativas incrementaron en 2,2 %, indicando que los estudiantes con el pasar del tiempo podrían haber disminuido su interés por alcanzar propósitos distintos al aprendizaje. Con relación a estos resultados, no se pudo rechazar la hipótesis nula, es decir, durante el tiempo del experimento los estudiantes del grupo de Ingeniería de sistemas no presentaron variaciones significativas en las sub-escalas relacionadas con los aspectos motivacionales estudiados.

Finalmente, con relación a las sub-escalas motivacionales, no fue posible encontrar efectos significativos entre el Pre-Test y Post-Test. Los p-valores generados por la prueba de Wilcoxon indican que es probable que las variaciones relacionadas en las demás sub-escalas motivacionales se deban a efectos del azar.

5.2. Percepciones de los estudiantes

A través de la encuesta de percepción sobre la herramienta VPL aplicada a los 21 estudiantes que participaron en el grupo experimental, pertenecientes al programa de Ingeniería de Sistemas se logró obtener la información correspondiente a las opiniones respecto al uso de la herramienta y la integración de la misma en las sesiones de clase. La encuesta se aplicó en la sesión posterior a la aplicación del Pos-Test.

Inicialmente se les indicó a los estudiantes el motivo de la aplicación de la encuesta, explicándoles la importancia de contestar de manera sincera y honesta cada una de las preguntas correspondientes, puesto que, esta actividad no tendría ningún tipo de calificación o ponderación académica que afecte su proceso formativo. Teniendo en cuenta lo anterior, se les indicó que la encuesta contiene algunas preguntas de respuesta abierta en la que podrían expresar de manera personal todo lo que pudieron percibir de la herramienta en el contexto de la asignatura y la integración en las clases.

De igual manera, se les hizo saber la importancia de que las respuestas fueran lo más claras posibles, brindando una justificación apropiada desde su punto de vista sobre los diferentes aspectos que componían la encuesta y se procedió a indicarles el enlace de ésta para su correspondiente aplicación.

A partir de los datos recolectados mediante las preguntas de respuesta abierta, se llevó a cabo el análisis de los datos. Este análisis se realizó a través de la técnica de codificación abierta que se indica en la Sección 4.6.2. Durante el proceso de codificación se analizaban cada una de las respuestas recolectadas y se procedía al procedimiento de agrupación teniendo en cuenta su objetivo, propósito o intención por medio de indicadores, los cuales a su vez, se agrupaban en conceptos a partir de la relación que existía entre éstos.

Al finalizar, se diseñaron categorías con los conceptos generados que permitieron desarrollar una comprensión general partiendo de la intención de cada concepto planteado. Sin embargo, es importante aclarar que algunas de las respuestas obtenidas, en su contenido agrupaban información que hacía parte de dos o más indicadores, puesto que es común que el estudiante indicará diferentes consideraciones en una misma respuesta.

El proceso de codificación realizado a partir de: descomposición, comparación, agrupación por categorías, indicadores y conceptos, y conceptualización de los datos obtenidos se llevó a cabo con el acompañamiento, contribución y sugerencias del director y codirector de la presente tesis de maestría, a través de varias sesiones de trabajo remoto. En éstas se consolidó el proceso de definición y refinamiento de la codificación propuesta.

Por medio del análisis cualitativo de los datos recolectados se identificaron las categorías, conceptos e indicadores, los cuales facilitaron el proceso de comprensión y soporte a los resultados hallados y obtenidos mediante el análisis cuantitativo. En el proceso de codificación se identificaron dos categorías generales teniendo en cuenta la organización y clasificación de las respuestas abiertas obtenidas por medio de los estudiantes. Las categorías mencionadas son: motivación y evaluación automática.

A través de la Figura 5-5 se puede observar de manera más amplia y general el resultado de la cantidad de respuestas a partir del proceso de agrupación de los indicadores en cada una de las categorías y conceptos. Cada una de las dos categorías se abordarán en las siguientes sub-secciones.

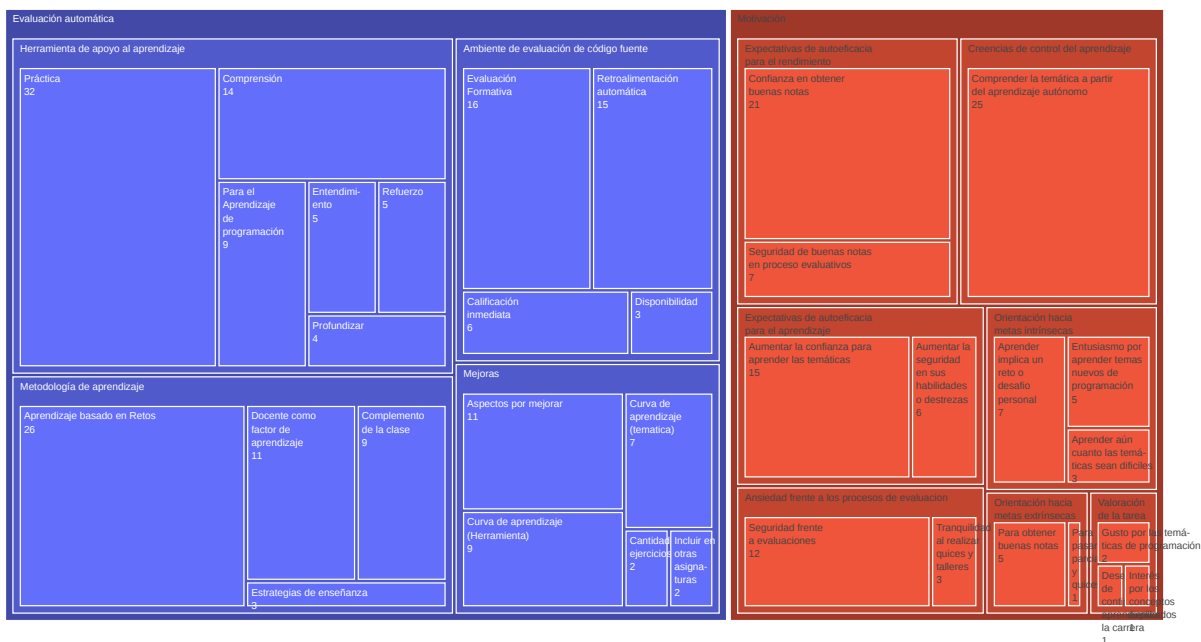


Figura 5-5.: Resultado del análisis cualitativo a partir de la encuesta de percepción - Categorías Motivación y Evaluación automática

5.2.1. Categoría: Motivación

A partir del proceso de análisis se identificó esta primera categoría, la cual está relacionada con la definición de **Motivación** en el aprendizaje. Esta categoría se crea a partir de las respuestas obtenidas de las preguntas tanto abiertas como por escala de Likert relacionadas con las sub-escalas de motivación en el aprendizaje las cuales están indicadas en la Tabla 4-7 de la Sección 4.5.2.

A través de la Figura 5-6 se observa la distribución de la cantidad de respuestas frente a cada una de las sub-escalas de la motivación en el aprendizaje a través de las preguntas en escala de Likert. Mediante esta gráfica de barras apiladas se pueden observar las respuestas de los estudiantes respecto a las respuestas por medio de la escala de Likert de la encuesta de percepción. A través de ésta se puede evidenciar que, de los 21 estudiante encuestados, la mayoría indicó algún nivel de acuerdo (4-Algo de acuerdo, 5-De acuerdo, 6-Totalmente de acuerdo) en todas las afirmaciones, donde el 95,2 % de los estudiantes están de acuerdo en que los ejercicios realizados en la herramienta VPL les permitió incrementar el deseo y gusto por la temáticas de programación de computadores (afirmación 1), el 100 % de los estudiantes está de acuerdo en que la herramienta VPL y los ejercicios generaron retos que incrementaron el deseo y las ganas de aprender programación (afirmación 3), igualmente el 100 % de los encuestados indicaron estar de acuerdo en que la herramienta de VPL les resultó importante, útil o necesaria para obtener mejores notas en la asignatura (afirmación 5), el 95,2 % están de acuerdo en

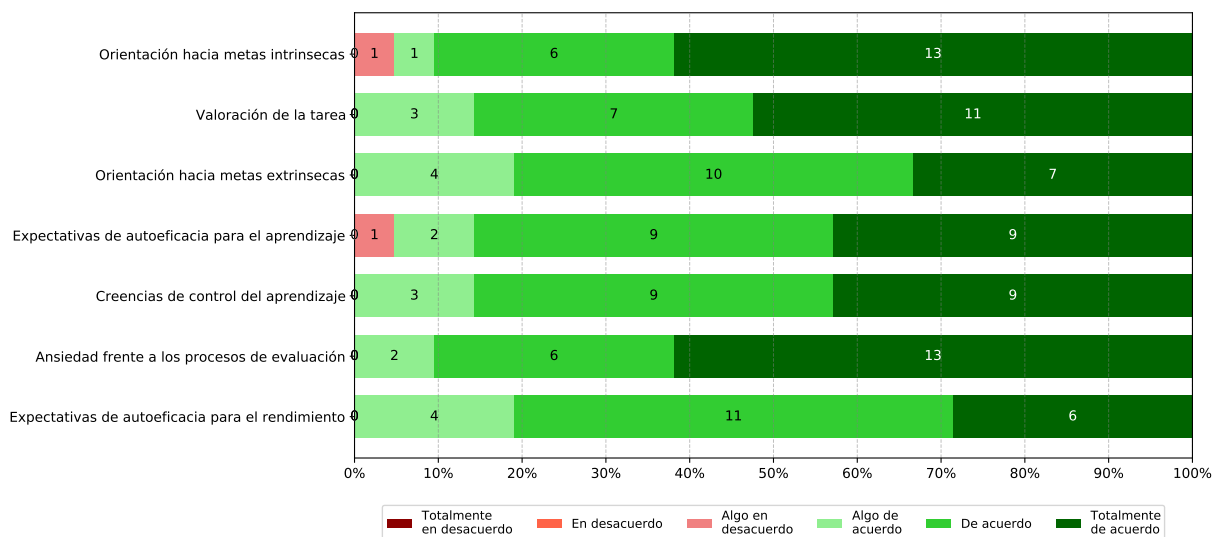


Figura 5-6.: Respuestas obtenidas con la Escala de Likert en la encuesta de percepción de VPL en el grupo experimental de Ingeniería de Sistemas

que la herramienta les aportó a incrementar los niveles de confianza y seguridad respecto a los temas más complejos o difíciles que se enseñan en programación (afirmación 7), el 100 % están de acuerdo en que usar la herramienta de evaluación automática les ayudó a incrementar la confianza para obtener buenas notas (afirmación 9), les ayudó a entender que la comprensión de las temáticas requerirá esfuerzo propio y dedicación (afirmación 11), y les ayudó a mejorar la seguridad frente al manejo y control de la ansiedad en procesos evaluativos (afirmación 13). Sin embargo, algunos encuestados expresaron algún nivel de desacuerdo, correspondiente a 4,8 % tanto en la afirmación 1 como en la afirmación 7.

Al observar la Figura 5-6 se puede evidenciar que las afirmaciones con más respuestas positivas por parte de los estudiantes están relacionadas con las sub-escalas de la motivación correspondientes a la de **orientación a metas intrínsecas** y la **ansiedad frente a procesos de evaluación**, lo cual indicaría una posible mejora respecto al análisis cuantitativo, el cual indicó que este último fue de los que menos mejoró respecto a la intervención. Igualmente, indica que los estudiantes perciben la herramienta como un complemento y apoyo útil para incrementar su deseo por aprender programación de manera genuina y que les aporta a mejorar su preocupación y seguridad al momento de enfrentarse a quices o exámenes parciales.

Por otro lado, a partir del proceso indicado para la categorización de las respuestas a las preguntas abiertas, se identificaron los siguientes 15 indicadores para los 7 conceptos correspondientes a las sub-escalas indicadas por el MSLQ-Colombia dentro de la categoría de motivación tal y como se observa en la Figura 5-7. Se puede evidenciar en el mapa de árbol (Treemap) que uno de los conceptos con más indicadores fue el asociado a las **expectativas de autoeficacia para el rendimiento** y las

creencias de control del aprendizaje. Es importante resaltar que estas dos sub-escalas de la motivación en el aprendizaje fueron dos de las que más altos niveles de acuerdo tuvieron por parte de los estudiantes en el análisis cuantitativo.

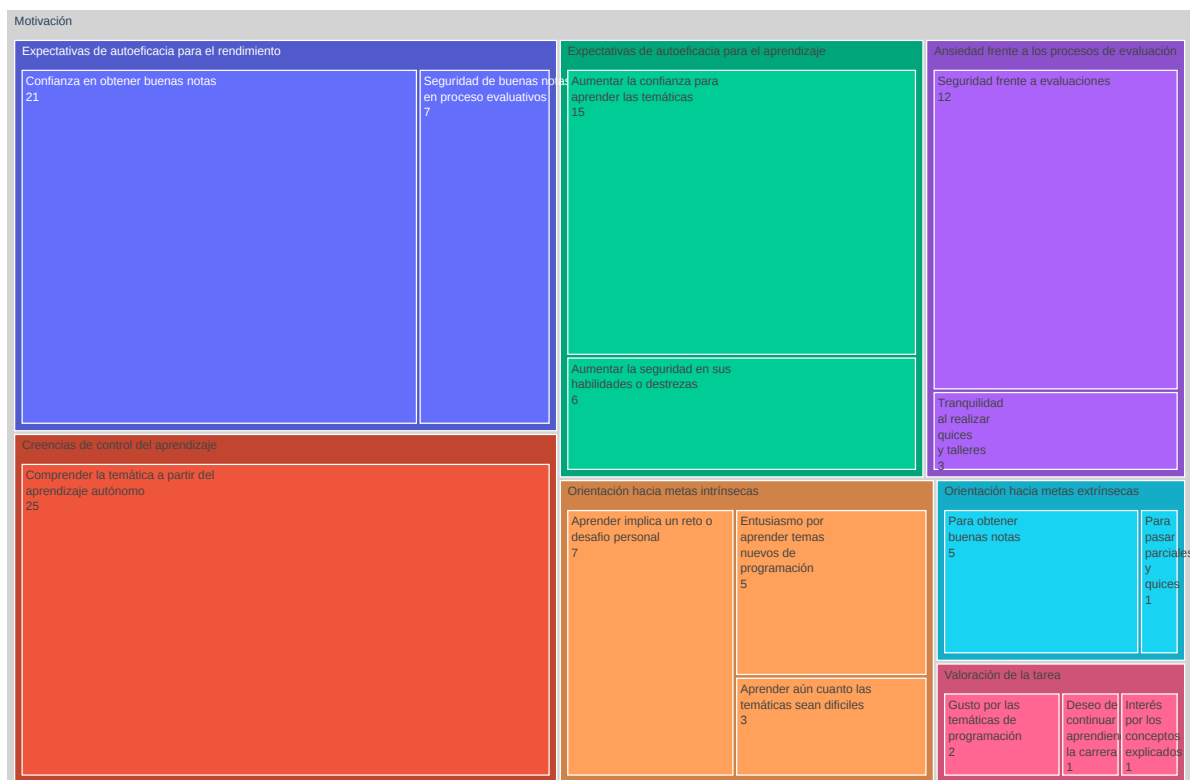


Figura 5-7.: Resultado del proceso de análisis cualitativo. Categoría “Motivación”

La Tabla 5-4 resume todos los conceptos, la definición, una cita de muestra respecto al concepto mencionado expresada por algún estudiante y la cantidad de opiniones expresadas obtenidas a partir de las respuestas dadas. Fueron 15 indicadores en total que expresan de manera positiva cada una de las afirmaciones mencionadas en la encuesta de percepción. Sin embargo, algunos de los estudiantes indicaron que la herramienta no es suficiente para apoyar el proceso formativo o para evaluar el desempeño de cada uno, por lo que igualmente se requiere una metodología que junto con el docente permita fomentar y mejorar el proceso de aprendizaje y enseñanza. Otros indicaron que la herramienta permite complementar la metodología de la clase o la explicación del docente permitiendo explorar de manera autónoma las temáticas comprendidas.

Tabla 5-4.: Conceptos en la categoría "Motivación" definidos a partir de las preguntas abiertas de los en la encuesta de percepción

Concepto	Definición	Respuesta de muestra	# Opiniones
Valoración de la tarea	El estudiante encuentra la tarea o actividad desafiante, útil e importante para su proceso de formación y muestra gusto por las temáticas planteadas.	<i>"Practicar con la herramienta vpl me permitio tomar un mayor gusto a los temas de programacion por los desafios de los ejercicios complejos" (sic)</i>	4
Orientación hacia metas intrínsecas	En este indicador el estudiante demuestra un interés, curiosidad por el conocimiento nuevo o deseo genuino en aprender las temáticas asociadas a la asignatura.	<i>"La herramienta me parece muy buena para el aprendizaje, ya que motiva a seguir aprendiendo puesto que cada vez hay mas retos con cada ejercicio" (sic)</i>	15
Orientación hacia metas extrínsecas	El proceso de aprendizaje en el estudiante se puede ver motivado por factores tales como: la nota, el reconocimiento, premios, becas o la competencia.	<i>"Me ayudo mucho a poder sacar buenas notas, porque pude entender algunos ejercicios de los quices con soluciones similares" (sic)</i>	6
Expectativas de autoeficacia para el aprendizaje	El indicador permite mostrar la percepción del estudiante sobre las probabilidades de éxito en el proceso de aprendizaje, teniendo en cuenta creencias sobre sus propias capacidades para alcanzar las metas propuestas.	<i>"Ayuda en generar confianza para cada ejercicio, al principio puede tornarse difícil pero a medida de que se practica se va adquiriendo mas seguridad" (sic)</i>	21
Expectativas de autoeficacia para el rendimiento	Permite al estudiante auto-evaluar su capacidad y la confianza en las probabilidades de éxito frente al proceso de evaluación de la asignatura.	<i>"Conforme utilice la herramienta me sentia mas seguro al enfrentarme a las evaluaciones" (sic)</i>	28
Creencias de control del aprendizaje	Este indicador explora la percepción del estudiante frente al proceso de aprendizaje, determinando que este se verá afectado por su propio esfuerzo y no por factores externos tales como la nota o el docente.	<i>"Aporto a tener mas seguridad en los exámenes pero todo depende de mi y de mi proceso de estudio y practica" (sic)</i>	25
Ansiedad frente a los procesos de evaluación	El estudiante se enfrenta a nervios o preocupación al momento de realizar los procesos de evaluación de la asignatura.	<i>"Aunque la ansiedad ante un parcial siempre estan se pueden disminuir con los ejercicios planteados en la herramienta" (sic)</i>	15

Fuente: Elaboración propia

El concepto inicial es el relacionado con la **valoración de la tarea** en la cual se tuvieron en cuenta 3,51 % de las respuestas. Para esta categoría se realizó la agrupación a partir de 3 indicadores relacionados a esta sub-escala de la motivación. Para el primer indicador asociado al **gusto por las temáticas de programación** se relacionaron 2 respuestas. Para este indicador los estudiantes mencionaban que la herramienta VPL les ayudaba a adquirir un mayor interés o agrado por los temas o explicaciones relacionadas con la programación de computadores. A continuación se citan algunos de los comentarios u opiniones textuales de los estudiantes respecto a este indicador: *“Practicar con la herramienta vpl me permitio tomar un mayor gusto a los temas de programacion por los desafios de los ejercicios complejos”* (sic), *“Tener una buena nota en la asignatura puede ser motivador para el aprendizaje personal, pero no es tan importante como el aprender todos los conceptos de la programación y entender que resulta ser más sencillo.”* (sic).

Para los otros dos indicadores del concepto de **valoración de la tarea** se dieron obtuvieron como resultado una opinión de cada uno, puesto que el comentario de los estudiantes indicaban algo puntual y distinto al anterior indicador mencionado. Por un lado, el indicador relacionado con el **deseo de continuar aprendiendo sobre la carrera** se evidencia un comentario en el cual el estudiante expresa que es útil la herramienta puesto que le incrementa el interés o ganas de continuar aprendiendo sobre su carrera y encuentra la herramienta de ayuda para su proceso de formación. La opinión indicada textualmente por el estudiante es la siguiente: *“La evaluacion automatica en la herramienta me permitieron aprender muchos temas de programación y motivarme por estudiar la carrera”* (sic). Por otro lado, el otro indicador hace relación al **interés por los conceptos explicados**. En éste el estudiante indica que la herramienta VPL y la práctica en los ejercicios dentro de ésta le permitieron comprender con agrado la programación de computadores pues lo considera útil para su crecimiento personal o profesional. La opinión textual del estudiante es la siguiente: *“los ejercicios que estan en la plataforma me permitieron entender lo emocionante de la programacion”* (sic).

Para el segundo concepto asociado a la sub-escala de **orientación hacia metas extrínsecas** las cuales obtuvieron un 5,36 % de las opiniones totales. Para este concepto se tuvieron en cuenta dos indicadores. En el primer indicador se obtuvieron 5 opiniones que hacían referencia a **obtener buenas notas**, en el cual los estudiantes indicaban que el practicar y utilizar la herramienta les permitía sacar buenas calificaciones y que esto motivaba su proceso de aprendizaje. Algunos de los comentarios obtenidos son los siguientes: *“Si aporta mucho para obtener mejores notas, seria bueno tener esta herramienta para las siguientes asignaturas de programacion y asi practicar”* (sic), *“Me ayudo mucho a poder sacar buenas notas, porque pude entender algunos ejercicios de los quices con soluciones similares”* (sic), *“Tener una buena nota en la asignatura puede ser motivador para el aprendizaje personal, pero no es tan importante como el aprender todos los conceptos de la programación y entender que resulta ser más sencillo.”* (sic).

Por otra parte, el otro indicador del concepto asociado a la **orientación hacia metas extrínsecas**, fue el que se relaciona con el apoyo para **pasar parciales y quices** brindado por la herramienta. En esta se tuvo en cuenta una opinión en la cual se indica que la practica en la herramienta VPL es buena para su

proceso de formación, pues permite pasar alguna evaluación con buenas notas a partir de los diversos ejercicios que se encuentran allí. La opinión del estudiantes es la siguiente: *“la herramienta es una gran ayuda para ejercitar los aprendido en clase y así pasar los parciales”* (sic).

Para el concepto relacionado con la sub-escala de **orientación hacia metas intrínsecas** que agrupó el 13,16 % de las percepciones de los estudiantes, se plantearon 3 indicadores. En 7 respuestas los estudiantes indicaron que **aprender implica un reto o desafío personal** puesto que los ejercicios planteados en la herramienta VPL contaban con un grado de dificultad que motivaba a los estudiantes a aprender para resolverlos pues los encontraban desafiantes o retadores. En este indicador se encuentran opiniones tales como: *“Porque me retaba a encontrar los componentes necesarios para realizar el ejercicio de manera adecuada”* (sic), *“La herramienta me parece muy buena para el aprendizaje, ya que motiva a seguir aprendiendo puesto que cada vez hay mas retos con cada ejercicio”* (sic), *“hubieron muchos ejercicios que generaron un reto en mi en lo personal eso motiva mucho a aprender algo porque no es tan facil a la primera”* (sic), *“Practicar con la herramienta vpl me permitio tomar un mayor gusto a los temas de programacion por los desafios de los ejercicios complejos”* (sic).

Otro indicador que se obtuvo a partir del análisis realizado, se asocia a **aprender aún cuando las temáticas sean difíciles** correspondiente al concepto de **orientación hacia metas intrínsecas**. Este indicador se dio como resultado de 3 opiniones que agrupaban las respuestas de los estudiantes que indicaban que el uso de la herramienta en su proceso formativo les apoyaba para aprender a partir de la práctica sin importar que los nuevos temas fueran complejos de entender. Entre las respuestas se pueden evidenciar las que se mencionan a continuación: *“Porque me motivaba aprender a programar ya que que cada ejercicio era un poco mas dificil”* (sic), *“Me motive a aprender mas sobre la programación al ver que algunos ejercicios se me hicieron complicados al inicio y al utilizar la herramienta empezaron a facilitarsen”* (sic), *“Porque no sabia nada acerca de la programacion y esta me motivo a estudiar mas y aprender al menos lo basico”* (sic).

Finalmente, las 5 percepciones recopiladas dentro del concepto de **orientación hacia metas intrínsecas** están relacionadas con el **entusiasmo por aprender temas nuevos de programación**. En este indicador los estudiantes expresan que la herramienta les permite aprender nuevos temas a partir de la practica de lo explicado en la clase y el desarrollo de ejercicios que impliquen comprender una nueva temática. Para este caso se mencionan algunas de las respuestas: *“La evaluacion automatica en la herramienta me permitieron aprender muchos temas de programación y motivarme por estudiar la carrera”* (sic), *“Porque me motivo para seguir aprendiendo temas nuevos”* (sic), *“Tener una buena nota en la asignatura puede ser motivador para el aprendizaje personal, pero no es tan importante como el aprender todos los conceptos de la programación y entender que resulta ser más sencillo.”* (sic), *“la herramienta me motiva a aprender mas sobre la programacion y los algoritmos”* (sic).

Otro de los conceptos corresponde al relacionado con la **expectativas de autoeficacia para el aprendizaje**. Este fue el tercer concepto con más opiniones, contando con un 18,42 % de respuestas respecto

al total de opiniones. En este concepto se agruparon las respuestas a partir de 2 indicadores. En el primer indicador los estudiantes expresaban que el uso de la herramienta de evaluación automática en las clases de programación de computadores les ayudaba a **aumentar la confianza para aprender las temáticas**, puesto que les permitía confiar en que el aprendizaje se daría de manera más adecuada entre más se practicara dentro de la herramienta. Algunas de las respuestas dadas por los estudiantes son las siguientes: *“Con los resultados durante la asignatura se genera más confianza para continuar con la temática de la programación y la seguridad de poder entender las asignaturas más avanzadas de programación.”* (sic), *“la herramienta de vpl me permitio comprender y tener mas seguridad al enfrentarme a temas mas dificiles”* (sic), *“la confianza se da apartir de una buena sesion de practica constante para desarrollar esos conceptos complicados”* (sic), *“cuando se usa la herramienta se puede practicar y genera confianza con los ejercicios”* (sic), *“el uso de eta herramienta en la asignatura ayuda a comprender mejor los temas y obtener mas confianza con nuestras soluciones”* (sic), *“Ayuda en generar confianza para cada ejercicio, al principio puede tornarse dificil pero a medida de que se practica se va adquiriendo mas seguridad”* (sic).

Por otra parte, el otro indicador relacionado con el concepto mencionado anteriormente, agrupaba las 6 respuestas en las que se indicaba que la herramienta ayuda a los estudiantes a **aumentar la seguridad en sus habilidades o destrezas**, puesto que el practicar y realizar los ejercicios dentro de la herramienta permite fortalecer estas habilidades de programación a la par que les permite motivarse por aprender entre más comprenden cada temáticas. Para este indicador se pueden observar las siguientes respuestas de los estudiantes: *“Por que me lleva a sentirme mas seguro de mis habilidades frente a los talles o evaluaciones”* (sic), *“la herramienta me permitio confiar en mis habilidades y en mis resultados al praticar”* (sic), *“Porque al ejercitar estas habilidades permitian tener mayor dominio del tema y obtener buenas notas”* (sic).

Con un 24,56 % de las opiniones totales se vincula el concepto con mayor cantidad de respuestas agrupadas, el cual es el relacionado con la sub-escala de **expectativas de autoeficacia para el rendimiento** con un total de 28 comentarios u opiniones agrupadas. Para este concepto se plantearon 2 indicadores. El primer indicador esta relacionado con la **confianza en obtener buenas notas** con un total de 21 opiniones. En este indicador los estudiantes indican que al hacer uso de la herramienta VPL podían aumentar o incrementar su confianza respecto a obtener buenas calificaciones en los talleres o actividades planteadas en la asignatura. Algunas de las opiniones recolectadas para este indicador son las siguientes: *“con la practica en la herramienta pude aumentar mi confianza para obtener buenas notas”* (sic), *“la confianza se va adquiriendo con la practica pero es complicado avanzar sin la ayuda del docente en algunos errores”* (sic), *“La herramienta permite practicar lo aprendido y esto permite que se obtengan mejores notas a medida de que se avanza el ccurso”* (sic), *“La herrmaienta de VPL me ayudo a obtener mejores calificaciones en la asignatura puesto que complementa la enseñanza del docente en la clase”* (sic), *“Si bien la herramienta me ayudo a mejorar mis habilidades la practica fue la que me llevo a obtener mejores notas”* (sic), *“la herramienta conto con muchos ejercicios que permitieron practicas y asi mismo poder mejorar mis calificaciones”* (sic), *“Porque obtuve muy buenas calificaciones ya que la herramienta me ayudo a practicar”* (sic).

Respecto al segundo indicador del concepto relacionado con **expectativas de autoeficacia para el rendimiento** se encuentra el que se asocia con la **seguridad de buenas notas en el proceso evaluativo**. En este indicador los estudiantes expresaban que la herramienta VPL les brindaba una ayuda importante en la preparación para las evaluaciones, a su vez que les aportaba mayor seguridad al presentar quices o parciales. Algunas de las opiniones expresadas por los estudiantes son: *“Pude adquirir mayor confianza en la asignatura al practicar, puesto que pude conocer las diferentes maneras de resolver los errores generados y así obtener buenas notas en los quices y parciales”* (sic), *“Conforme utilice la herramienta me sentía más seguro al enfrentarme a las evaluaciones”* (sic), *“conforme practicaba en los ejercicios dentro de la herramienta se podía tener más seguridad al momento de los quices y talleres”* (sic), *“la herramienta apoya muchísimo a mejorar las notas en los quices y parciales puesto que si no practicaba no podía saber la razón de alguna re-troalimentación en la plataforma y por ende no sabía cómo resolverlo”* (sic), *“VPL permite que al practicar se puedan empezar a facilitar los ejercicios y así se va uno preparando para los quices y para los parciales”* (sic), *“la herramienta permite aprender practicando por uno mismo lo que permite tomar más confianza al enfrentar los quices y parciales finales”* (sic).

En cuanto al concepto que se relaciona directamente con la sub-escala sobre las **creencias de control del aprendizaje** se tuvieron en cuenta 21,93 % de las respuestas obtenidas. Este resultó ser el segundo concepto con más opiniones agrupadas. Para este concepto solo se tuvo en cuenta un único indicador el cual se agrupó a partir de las 25 respuestas que corresponden a la sub-escala nombrada previamente. Para este único indicador los estudiantes indicaban que la herramienta VPL les apoyaba para **comprender la temática a partir del aprendizaje autónomo**, es decir, la herramienta VPL les permite practicar constantemente y realizar ejercicios con distintos grados de dificultad. Sin embargo, esto depende del estudio autónomo, la dedicación de tiempo y concentración entre otros factores que son propios y voluntarios de cada estudiante. Teniendo en cuenta la agrupación planteada se presentan algunas de las opiniones de los estudiantes: *“con la herramienta pude ver que habían muchas falencias en mis conocimientos y me motive a aprenderlos por mi cuenta practicando los ejercicios que estaban por realizar”* (sic), *“Aporto a tener más seguridad en los exámenes pero todo depende de mí y de mi proceso de estudio y práctica”* (sic), *“La herramienta permite ver la programación más sencilla pero para que sea así se debe tener en cuenta que depende de uno mismo el progreso del aprendizaje, con el interés, el esfuerzo, el entendimiento, la investigación.”* (sic), *“la herramienta me permitió profundizar en las temáticas puntuales sin embargo se necesita practicar por cuenta propia”* (sic), *“La herramienta ayuda a comprender y poner a prueba los conocimientos adquiridos pero también el tiempo invertido de manera personal a las temáticas”* (sic), *“Es muy importante esto, si no se practica no se ven cumplidos los retos de la asignatura, pero todo depende de nosotros”* (sic).

Finalmente, como último concepto a abordar, se encuentra el concepto que se relaciona con la sub-escala correspondiente a la **ansiedad frente a los procesos de evaluación**, el cual cuenta con el 13,16 % de las respuestas recopiladas. Para este concepto se proponen dos indicadores. El primero de ellos se relaciona con la **seguridad frente a evaluaciones**. En este los estudiantes indicaron que la herramienta VPL les ayuda a mejorar los niveles de confianza y seguridad al momento de enfrentarse a

cualquier tipo de proceso evaluativo (quiz, evaluación, taller en clase, parcial, etc). Para este primer indicador se pueden evidenciar respuestas tales como: *“La herramienta al ser más intuitiva permite tener los conceptos claros de lo que se debe realizar y cómo se debe hacer permitiendo no generar tanta ansiedad o dudas al momento de hacer un parcial”* (sic), *“Aunque a veces la herramienta te puede hacer sentir inseguro por los errores cuando practicas esa ansiedad ante las evaluaciones se van disminuyendo”* (sic), *“con ayuda de vpl puedo mejorar mi confianza propia al momento de enfrentarme a los parciales”* (sic), *“Aunque la ansiedad ante un parcial siempre están se pueden disminuir con los ejercicios planteados en la herramienta”* (sic), *“Entre más avanzaba la ansiedad por hacerlas bien aumentaba, la herramienta me ayudó a sentirme más seguro en los parciales”* (sic), *“Ayuda a mejorar la seguridad puesto que entre más practico ejercicios más puedo estar confiado en las evaluaciones”* (sic).

Por último, se encuentra el otro indicador del concepto de **ansiedad frente a los procesos de evaluación** el cual corresponde a la **tranquilidad al realizar quices y talleres**. Este indicador se agrupa a partir de 3 respuestas obtenidas de los estudiantes. Este indicador hace referencia al apoyo que brinda la herramienta de evaluación automática VPL en el proceso del control de la ansiedad a partir de una sensación de tranquilidad y seguridad mientras se practica para prepararse ante algún quiz o parcial. Las respuestas obtenidas para este indicador son las siguientes: *“Me sentí más tranquilo al realizar cualquier tipo de actividad en la herramienta, con la opción de poderse equivocar al menos 1 vez ya se tiene la tranquilidad de corregir algún error y poder la mejor nota posible”* (sic), *“Porque así no tengo tantos nervios en los parciales, ya que he practicado para tener mejores notas”* (sic), *“Porque me ayuda a controlar los nervios al realizar los ejercicios y sentir más seguridad en los quices”* (sic).

5.2.2. Categoría: Evaluación automática

Por otra parte, como resultado del análisis cualitativo a partir de las respuestas a las preguntas abiertas en la encuesta de percepción sobre VPL aplicada a los estudiantes del grupo experimental, se identificó otra categoría relacionada con el concepto de **Evaluación automática**. En esta se agrupan los conceptos e indicadores que se asocian a la percepción que tuvieron los estudiantes respecto a la integración de la metodología de evaluación automática a través de la herramienta VPL en las sesiones de la clase de programación de computadores. Se puede observar en la Figura 5-8 la organización de los conceptos e indicadores que se relacionan con esta categoría, el cual dio como resultado la agrupación de 189 opiniones.

La Tabla 5-5 agrupa cada uno de los conceptos, los indicadores, su definición, una cita textual de la muestra respecto al indicador mencionado expresada por algún estudiante y la cantidad de opiniones expresadas recolectadas mediante cada respuesta obtenida. En total se identificaron 19 indicadores que expresan de manera positiva cada una de las afirmaciones mencionadas en la encuesta de percepción sobre VPL. Sin embargo, algunos estudiantes indicaron varias sugerencias respecto al uso de la herramienta en la metodología de la clase de programación de computadores. Por un lado, unos indicaron que si bien la herramienta ayuda al proceso de aprendizaje de programación y puede ser-

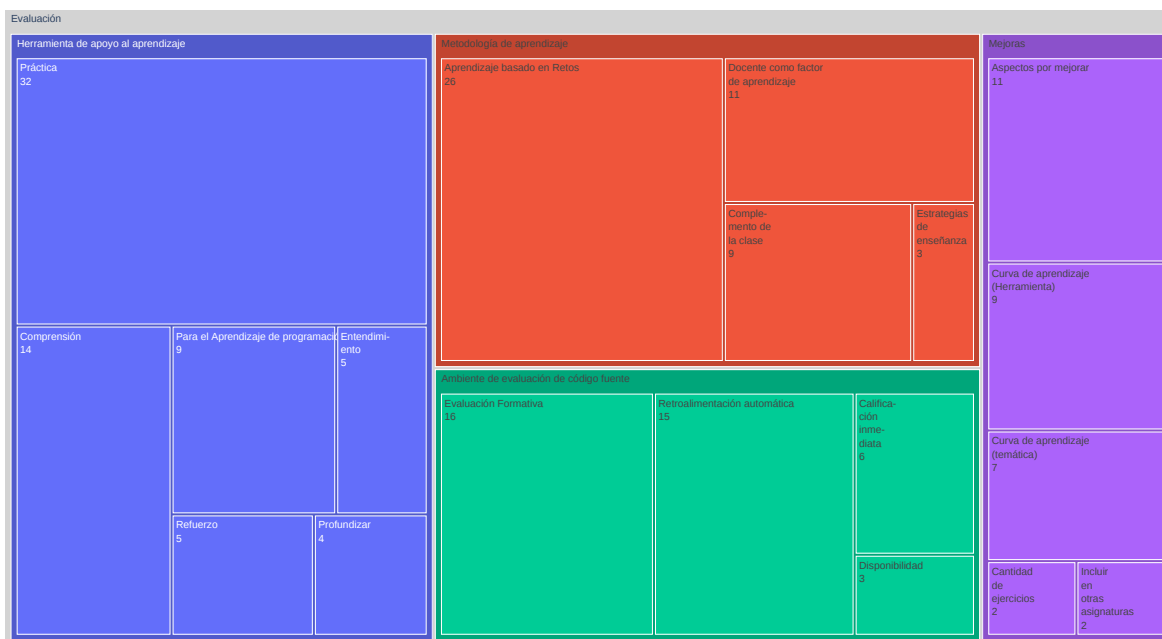


Figura 5-8.: Resultado del proceso de análisis cualitativo. Categoría “Evaluación automática”

vir como complemento a las temáticas explicadas en las sesiones de clase, el docente sigue jugando un papel importante en la comprensión y la enseñanza de este campo. Por otro lado, los estudiantes indicaban que se requerían muchos más ejercicios y con dificultades más amplias con el fin de desafiarse a aprender mediante el reto que les representa completar estos ejercicios. Así mismo, algunos estudiantes expresaban que el proceso de adaptación de la herramienta y su curva de aprendizaje implicaba un tiempo adicional para finalmente comprender las temáticas de la asignatura.

A través del mapa de árbol (Treemap) de la Figura 5-8 se puede evidenciar que uno de los conceptos al que más se refirieron los estudiantes fue el que se expresa que VPL sirve como **herramienta de apoyo al aprendizaje** agrupando el 36,51 % de las opiniones totales recolectadas, correspondientes a 69 opiniones. En este concepto, 32 de las opiniones agrupadas indicaban que la herramienta VPL fue útil e importante para el proceso de aprendizaje a través de la **práctica** en ejercicios de programación. Algunas opiniones obtenidas fueron las siguientes: “Porque me permiten aprender mientras practico y ejercito mis habilidades” (sic), “Porque fueron ejercicios difíciles que si no hubiera practicado con la herramienta no los habria comprendido” (sic), “la herramienta permite aprender practicando por uno mismo lo que permite tomar mas confianza al enfrenar los quices y parciales fnales” (sic), “con la herramienta pude ver que habian muchas falencias en mis conocimientos y me motive a aprenderlos por mi cuenta practicando los ejercicios que estaban por realizar” (sic), “con la herramienta pude ver que habian muchas falencias en mis conocimientos y me motive a aprenderlos por mi cuenta practicando los ejercicios que estaban por realizar” (sic), “El autoaprendizaje se puede lograr con la herramienta, se puede practicar y asi adquirir mejores habilidades en la programacion”

(sic), “Porque entre mas tiempo practicaba y mas tiempo dedicaba a estudiar los ejercicios en la herramienta mejores resultados tenia y mas claros quedaban los temas” (sic).

Tabla 5-5.: Conceptos e indicadores en la categoría "Evaluación automática" definidos a partir de las preguntas abiertas de los en la encuesta de percepción

Concepto	Indicador	Definición	Respuesta de muestra	# Opiniones
Herramienta de apoyo al aprendizaje	Práctica	Permite evidenciar la importancia de la herramienta en el proceso de aprendizaje facilitando la práctica para mejorar habilidades de programación.	<i>"Porque me permiten aprender mientras practico y ejercito mis habilidades" (sic)</i>	32
	Profundizar	Determina la necesidad de practicar en la herramienta con el fin de profundizar en las temáticas planteadas.	<i>"la herramienta me permitio profundizar en las tematicas puntuales sin embargo se necesita practicar por cuenta propia" (sic)</i>	4
	Refuerzo	Apoya el refuerzo de los temas a partir de la integración de ejercicios con dificultad similar a los presentados en procesos evaluativos.	<i>"Ya conocia muchas cosas de programacion y pude reforzar conocimientos a lo largo del curso con los ejercicios de la herramienta" (sic)</i>	5
	Comprensión	Se expresa como el conocimiento que se hace propio y se interioriza a partir de lo que se ha entendido.	<i>"al hacer uso de la herramienta como complemento de la clase permite comprender las tematicas a partir de un estudio autonomo" (sic)</i>	14
	Entendimiento	Se determina a partir de tener una percepción de claridad de algún conocimiento o situación, aún cuando no se logró comprender completamente.	<i>"Porque me permitio aclarar los conceptos vistos en clase y ponerlos en practica en la herramienta" (sic)</i>	5
	Para el Aprendizaje de programación	El estudiante indica que la herramienta permite mejorar o aportar al aprendizaje en el campo de la programación de computadores.	<i>"Excelente, una herramienta eficaz para el aprendizaje de los estudiantes, y una buena iniciación para la programación y que no se vea como algo imposible de aprender." (sic)</i>	9

Tabla 5-5.: Continuación de Conceptos e indicadores en la categoría "Evaluación automática" definidos a partir de las preguntas abiertas de los en la encuesta de percepción

Continuación de Tabla 5-4

Concepto	Indicador	Definición	Respuesta de muestra	# Opiniones
Ambiente de evaluación de código fuente	Evaluación Formativa	Permite que el estudiante comprenda sus errores y trabaje en pro de mejorarlos en situaciones similares sin la intervención de factores como la nota o el rendimiento académico.	<i>"La herramienta de vpl es muy buena para el aprendizaje, nos motiva a estar desarrollando ejercicios para aprender nuevas habilidades y nos ayuda a entender nuestros errores" (sic)</i>	16
	Calificación inmediata	Permite valorar la intervención de manera inmediata de la herramienta en la asignación de una nota cuantitativa para medir las habilidades adquiridas.	<i>"Si, al saber mi calificacion inmediatamente me motiva y de da seguridad de que hice bien las cosas" (sic)</i>	6
	Disponibilidad	La disponibilidad evidencia una de las características la cual es permitir así su uso en cualquier momento y cualquier lugar desde que se cuente con conexión a internet.	<i>"Ayuda mucho el hecho de que se puede generar ese autoaprendizaje en el momento que se quiera y que aun cuando el docente no esta presente se puede tener una retroalimentacion por parte del programa" (sic)</i>	3
	Retroalimentación automática	Expresa la posibilidad de la herramienta de mostrar de manera automática la retroalimentación acerca de la solución propuesta por los estudiantes a cada ejercicio.	<i>"VPL es muy buena herramienta permite la retroalimentacion automatica, la evaluacion automatica permite practicar y saber en que se esta equivocando y asi mismo ayuda a tener nuevos desafios para el aprendizaje" (sic)</i>	15
Metodología de aprendizaje	Aprendizaje basado en Retos	En este indicador se destacan los comentarios que valoran la integración de ejercicios que puedan representar un reto o desafío dentro del proceso de aprendizaje.	<i>"me motivo mucho aprender para lograr resolver los retos que estaban en la herramienta" (sic)</i>	26

Tabla 5-5.: Continuación de Conceptos e indicadores en la categoría "Evaluación automática" definidos a partir de las preguntas abiertas de los en la encuesta de percepción

Continuación de Tabla 5-4

Concepto	Indicador	Definición	Respuesta de muestra	# Opiniones
	Docente como factor de aprendizaje	En este indicador los estudiantes expresan que a pesar de la integración de alguna herramienta el docente aún tiene un rol fundamental en el proceso de aprendizaje.	<i>"si bien el esfuerzo depende de mi si no entendia la herramienta no podria tener buenas notas y no podria avanzar, el docente es fundamental en este proceso de aprendizaje" (sic)</i>	11
	Complemento de la clase	Expresa la importancia de que la herramienta este presente como un complemento, valorando su aporte al proceso de formación.	<i>"la herramienta ayuda a mejorar pero es complemento de las clases para que la tematica quede clara y despues poner en la herramienta" (sic)</i>	9
	Estrategias de enseñanza (Metodologías)	Indica que la integración de herramientas tecnológicas en el proceso de aprendizaje aporta a la formación del conocimiento y adquisición de habilidades de programación.	<i>"La herramienta es muy buena para que las clases no sean tan teoricas en algo como la programación, y motivan a que los estudiantes practiquen para aprender más, deberían implementarsen en otras clases" (sic)</i>	3
Aspectos por mejorar	Curva de aprendizaje (Herramienta)	Evidencian las opiniones de los estudiantes respecto a la dificultad presentada en la interacción inicial con la herramienta o con alguno de los mensajes presentados por esta.	<i>"me gusta la programacion pero al iniciar en la herramienta se torna dificil aunque a medida que se hacen ejercicios ya mejora" (sic)</i>	9
	Cantidad de ejercicios	Indica opiniones de sugerencia respecto a la cantidad de ejercicios, expresando contar con mayor cantidad de estos para su estudio autónomo.	<i>"aunque la herramienta lo prepara a uno para los parciales faltan mas ejercicios para poder seguir practicando" (sic)</i>	2

Tabla 5-5.: Continuación de Conceptos e indicadores en la categoría "Evaluación automática" definidos a partir de las preguntas abiertas de los en la encuesta de percepción

Continuación de Tabla 5-4

Concepto	Indicador	Definición	Respuesta de muestra	# Opiniones
	Curva de aprendizaje (temática)	Permite identificar la dificultad presentada para algunos estudiantes en las temáticas planteadas a lo largo del curso, presentando dificultades en el avance inicial de las mismas.	<i>"Ayuda en generar confianza para cada ejercicio, al principio puede tornarse difícil pero a medida de que se practica se va adquiriendo mas seguridad" (sic)</i>	7
	Por mejorar (Otros)	Algunos aspectos importantes que mejorarían la experiencia de integrar herramientas de evaluación automática en procesos de formación.	<i>"la verdad es que al tratar con la herramienta pueden salir errores que hacen desconfiar de las respuestas" (sic)</i>	11
	Incluir en otras asignaturas	Las opiniones indican la necesidad de tener ambientes de aprendizaje similares en asignaturas de programación de computadores.	<i>"es una herramienta muy completa, califica automáticamente, evalúa código, da retroalimentación y motiva a aprender con el reto de cada ejercicio, se debería incluir en otras asignaturas a futuro" (sic)</i>	2

Fuente: Elaboración propia

Por otra parte, para el concepto relacionado a que VPL es importante como **herramienta de apoyo al aprendizaje**, 14 de las opiniones estaban relacionadas a que la herramienta VPL apoya el proceso de enseñanza y aprendizaje pues permite la **comprensión** de las temáticas a partir del repaso o la realización de ejercicios con dificultad similar. Dentro de este indicador se encuentran las siguientes opiniones: *“Traer los temas de la clase a la herramienta es bueno para comprender completamente”* (sic), *“La herramienta ayuda a comprender y poner a prueba los conocimientos adquiridos pero tambien el tiempo invertido de manera personal a las tematicas”* (sic), *“Al poder evaluar en la misma herramienta y poder ejecutar cada codigo puedo repasar los temas vistos en clase y tambien comprenderlos con los ejercicios propuestos”* (sic), *“al hacer uso de la herramienta como complemento de la clase permite comprender las tematicas a partir de un estudio autonomo”* (sic), *“el uso de eta herramienta en la asignatura ayuda a comprender mejor los temas y obtener mas confianza con nuestras soluciones”* (sic), *“la herramienta de vpl me permitio comprender y tener mas seguridad al enfrentarme a temas mas dificiles”* (sic).

Así mismo, 9 de estas opiniones recolectadas en el concepto mencionado indicaban que VPL era útil **para el aprendizaje de la programación** puesto que contaba con ayudas como indicarle al estudiante sus errores, permitió la retroalimentación y evaluación de forma automática o permitía la posibilidad de realizar distintos ejercicios con el fin de mejorar en la temática a partir de poner a prueba lo aprendido. Para este indicador se relacionaron algunas de las siguientes opiniones: *“Me parece que es una buena herramienta para aprender a programar ya que si nos equivocamos nos muestra en que parte del código tenemos el error”* (sic), *“VPL me parece que es una excelente herramienta para ejecutar y evaluar de manera automatica y poder asi ver los errores de mi codigo, permite aprender mientras se desafia y motiva a continuar con los demas ejercicios”* (sic), *“La herramienta de vpl es muy buena para el aprendizaje, nos motiva a estar desarrollando ejercicios para aprender nuevas habilidades y nos ayuda a entender nuestros errores”* (sic), *“”* (sic), *“Es una excelente herramienta que permite desarrollar las habilidades de programacion a través de la ejecucion del codigo y la retroalimentacion, dependera del estudiante progresar o no y aprender si se practica”* (sic).

De la misma forma, 5 de las opiniones se referían a la utilidad que le encontraban a la herramienta VPL para el **refuerzo** de las temáticas y conceptos planteados en la asignatura, puesto que estos ejercicios que estaban dentro de la herramienta les apoyaba a repasar nuevamente lo visto en clase y realizar ejercicios con el fin de verificar el conocimiento adquirido. En este indicador se agruparon algunas respuestas tales como: *“Aunque siempre me ha gustado la programación la herramienta me permite reforzar muchos temas mientras practico con los ejercicios”* (sic), *“Porque la herramienta VPL ayuda a practicar y reforzar tematicas”* (sic), *“La herramienta considero que es bastante buena, al poder ejecutar automaticamente y ver mis calificaciones, errores de manera automatica se puede mejorar mucho en la pogramacon practicando y reforzando lo aprendido en la clase”* (sic). También, 5 de las opiniones totales brindadas en este concepto hacían referencia a la importancia que le hallaban a VPL para el **entendimiento** de las temáticas abordadas en la asignatura indicando que a partir de la práctica en la herramienta los conceptos vistos en clase podían quedar más claros y completar el aprendizaje mediante el uso de esta, al igual que entender temas más complejos. En este sentido, algunas de las opiniones relacionadas con este indicador son: *“Por que me permitio entender los explicado en la clase a traves de la practica”* (sic), *“al usar*

vpl se puede mejorar muchos ambitos y entender los temas que son mas dificiles” (sic), “Porque me permitio aclarar los conceptos vistos en clase y ponerlos en practica en la herramienta” (sic).

Finalmente, para este concepto mencionado se agruparon 4 opiniones que hacían relación al apoyo que brindaba la herramienta en el proceso de **profundizar** sobre los diversos temas de la asignatura. En este indicador se relacionaban todas las respuestas que expresaban opiniones respecto a la ayuda que ofrece la herramienta en ir más allá de lo que se explica en la clase a partir de ejercicios con dificultad similar o superior a partir de las mismas temáticas. Algunas de las opiniones que se relacionaron son: “aunque ya conocia algunos conceptos la herramienta me permitio profundizar en algunas tematicas puntuales” (sic), “Fue bueno para las notas sin embargo ayuda a profundizar y a dejar claros los conocimientos de cada sesion” (sic), “la herramienta me permitio profundizar en las tematicas puntuales sin embargo se necesita practicar por cuenta propia” (sic).

Por otra parte, se obtuvieron 49 opiniones correspondientes al 25,93 % de las percepciones obtenidas relacionadas con valoraciones y sugerencias en cuanto a la **metodología de aprendizaje** apoyada por herramientas de evaluación automática. En total, 26 de las opiniones recolectadas destacaban el hecho de que la herramienta de evaluación automática VPL posibilita obtener un **aprendizaje basado en retos**, indicando que el aprendizaje de la programación se aumenta a partir de los desafíos que encuentran al realizar los diversos ejercicios. Estas son algunas de las opiniones respecto a este indicador mencionado: “Practicar con la herramienta vpl me permitio tomar un mayor gusto a los temas de programacion por los desafios de los ejercicios complejos” (sic), “La herramienta me parece muy buena para el aprendizaje, ya que motiva a seguir aprendiendo puesto que cada vez hay mas retos con cada ejercicio” (sic), “en la herramienta pude motivarme cada vez mas al enfrentarme a retos mas complejos cada vez que avanzaba” (sic), “Porque me retaba a encontrar los componentes necesarios para realizar el ejercicio de manera adecuada” (sic), “hubieron muchos ejercicios que generaron un reto en mi en lo personal eso motiva mucho a aprender algo porque no es tan facil a la primera” (sic), “Si puesto que al realizar los ejercicios algunos casos nop se cumplian y me retaba a hacerlos hasta que quedaran perfectos” (sic), “Porque me motivaba aprender a programar ya que que cada ejercicio era un poco mas dificil” (sic).

Así mismo, se obtuvieron 11 opiniones que indicaban que si bien la herramienta apoyada muchos procesos de práctica de las temáticas seguía siendo importante el rol **docente como factor de aprendizaje**. En estas respuestas se indica que el profesor sigue teniendo un papel fundamental para la construcción del conocimiento. Algunas de las respuestas obtenidas fueron las siguientes: “la confianza se va adquiriendo con la practica pero es complicado avanzar sin la ayuda del docente en algunos errores” (sic), “si bien el esfuerzo depende de mi si no entendia la herramienta no podria tener buenas notas y no podria avanzar, el docente es fundamental en este proceso de aprendizaje” (sic), “ayuda en algunos aspectos pero confunde en otros, importante que el docente comparta informacion respecto a los errores” (sic), “Aunque la herramienta es muy buena para practicar los ejercicios y las tematicas, esta debe ser un complemento o ayuda más no una necesidad para el aprendizaje, el docente sigue siendo elemental en el proceso de enseñanza” (sic), “usar esta herramienta en la asignatura me genero muchos retos que con la ayuda del docente se pudieron resolver ” (sic).

De igual manera, en el concepto correspondiente a **metodología de aprendizaje** se obtiene como resultado otro indicador a partir de 9 opiniones que sugerían que aunque la herramienta era un gran apoyo para el proceso de enseñanza y aprendizaje, esta debía ser un **complemento de la clase** y no el centro de ésta. Algunas de las opiniones recolectadas son las siguientes: *“la herramienta ayuda a mejorar pero es complemento de las clases para que la tematica quede clara y despues poner en la herramienta”* (sic), *“La herrmaienta de VPL me ayudo a obtener mejores calificaciones en la asignatura puesto que complementa la enseñanza del docente en la clase”* (sic), *“Es un buen complemento para el aprendizaje pero siempre se requiere del docente no se podria depender unicamente de esta como conocimiento completo”* (sic), *“al hacer uso de la herramienta como complemento de la clase permite comprender las tematicas a partir de un estudio autonomo”* (sic). Finalmente, se encuentran 3 opiniones que se relacionan con los aportes que le brinda la **estrategia de aprendizaje** basada en la herramienta de evaluación automática VPL al estudiante. En este indicador los estudiantes expresan el apoyo que brinda la herramienta para que las clases de programación dejen a un lado los componentes teóricos y se enfoquen más en componentes prácticos. Las opiniones relacionadas con este indicador son las siguientes: *“La herramienta es muy buena para que las clases no sean tan teoricas en algo como la programación, y motivan a que los estudiantes practiquen para aprender más, deberían implementarsen en otras clases”* (sic), *“Es una muy buena herramienta para que la metodología no sea tan teorica, permite que los estudiantes podamos estar practicando lo visto en clase y retandonos con nuevos ejercicios, creo que falta integrar mas ejercicios con mayores niveles de dificultad”* (sic), *“la herramienta de vpl es muy buena, permite al profesor medir a los estudiantes y al mismo tiempo mejorar las formas de enseñanza, los ejercicios son buenos desafios que ayudan a querer seguir estudiando los temas de programacion”* (sic).

Otro concepto identificado a partir de las respuestas respecto a la percepción de VPL fue el relacionado con las características más destacadas por los estudiantes de un **ambiente de evaluación automática de código fuente**, siendo estas un 21,16 % de las opiniones totales, contando con 40 respuestas. En este concepto se recolectaron 16 opiniones que se referían a las ventajas y aportes al aprendizaje que brinda la **evaluación formativa** por parte de VPL. A partir de este indicador se pueden evidenciar respuestas como: *“Si uno como estudiante practica vera los resultados y esta herramienta permite eso, practicar sin penalizacion de nota”* (sic), *“Pude adquirir mayor confianza en la asignatura al practicar, puesto que pude conocer las diferentes maneras de resolver los errores generados y asi obtener buenas notas en los quices y parciales”* (sic), *“La herramienta de vpl es muy buena para el aprendizaje, nos motiva a estar desarrollando ejercicios para aprender nuevas habilidades y nos ayuda a entender nuestros errores”* (sic), *“Permite corregir y evaluar al detalle la forma de programar, obteniendo un aprendizaje más enriquecedor”* (sic), *“La herramienta considero que es bastante buena, al poder ejecutar automaticamente y ver mis calificaciones, errores de manera automatica se puede mejorar mucho en la pogramacon practicando y reforzando lo aprendido en la clase”* (sic), *“con la herramienta se pueden evidenciar los errores y asi poder comprender cuales fallas se tiene en el codigo para asi mejorar en los temas mas complejos ”* (sic), entre otras.

Otro indicador identificado dentro del concepto relacionado con las características de un **ambiente**

de evaluación de código fuente fue la **retroalimentación automática** con un total de 15 opiniones obtenidas. En este indicador se relacionaban todas las opiniones que expresaban el aporte de la información brindada por la herramienta VPL a cada estudiante al momento de evaluar sus soluciones, a través de la retroalimentación entregada automáticamente. Se pueden evidenciar algunas de las opiniones que se mencionan a continuación: *“Me ayudo a conocer mis errores con la retroalimentacion y la evaluacion rapida, por lo que pude estar preparada para los exámenes”* (sic), *“Porque me guia con cada ejercicio a aprender algo nuevo mientras me da una retroalimentacion de mis errores asi puedo estar preparado para las evaluaciones”* (sic), *“Ayuda mucho el hecho de que se puede generar ese autoaprendizaje en el momento que se quiera y que aun cuando el docente no esta presente se puede tener una retroalimentacion por parte del programa”* (sic), *“como la herramienta permite la retroalimentacion cada error genera un nuevo conocimiento para saberlo tratar y cuando enfrento las evaluaciones finales”* (sic), *“Gracias al poder programar en cualquier lugar con la herramienta y al tener una retroalimentacion instantanea”* (sic).

Adicionalmente, se identificó otro indicador de este concepto que hace referencia a la característica de **calificación inmediata** con 6 respuestas obtenidas. En este indicador los estudiantes expresaban que el conocer la nota de una entrega de forma inmediata permitía aprender de los errores cometidos y de igual manera poder corregir la solución enviada. Estas son algunas de las opiniones expresadas por los estudiantes: *“Era bastante clara y concisa con lo que se tenia que hacer, ademas podia saber mi nota inmediatamente”* (sic), *“Si, al saber mi calificacion inmediatamente me motiva y de da seguridad de que hice bien las cosas”* (sic), *“La herramienta considero que es bastante buena, al poder ejecutar automaticamente y ver mis calificaciones, errores de manera automatica se puede mejorar mucho en la pogramacon practicando y reforzando lo aprendido en la clase”* (sic), *“Me ayudo a conocer mis errores con la retroalimentacion y la evaluacion rapida, por lo que pude estar preparada para los exámenes”* (sic). Finalmente, se recolectaron 3 opiniones que hacían referencia a las ventajas de la **disponibilidad** de la herramienta para permitir realizar ejercicios prácticos en cualquier momento y lugar, siempre y cuando se cuenta con un equipo computacional y conexión a internet. Se tuvieron en cuenta las siguientes opiniones: *“Ayuda mucho el hecho de que se puede generar ese autoaprendizaje en el momento que se quiera y que aun cuando el docente no esta presente se puede tener una retroalimentacion por parte del programa”* (sic), *“al estudiar de manera autonoma y en cualquier momento permite comprendercon ejercicios las tematicas”* (sic), *“Gracias al poder programar en cualquier lugar con la herramienta y al tener una retroalimentacion instantanea”* (sic).

Por último, un 17,99 % de las opiniones recolectadas correspondientes a 34 opiniones relacionaban aspectos de **mejoras** respecto al proceso de integración de la herramienta VPL o a la metodología implementada a partir de esta. En el indicador de aspectos **por mejorar** se destacan 11 respuestas relacionadas con los errores presentados por la herramienta, la integración de mayor ejercicios de dificultad y la ausencia de casos más utilizados o reales en le campo del desarrollo. Se destacan algunas de las siguientes opiniones: *“ayuda en algunos aspectos pero confunde en otros, importante que el docente comparta informacion respecto a los errores”* (sic), *“la verdad es que al tratar con la herramienta pueden salir errores que hacen desconfiar de las respuestas”* (sic), *“Es una herramienta excelente para el aprendizaje y la motivacion para seguir explorando este campo, sin embargo se requieren mas ejercicios y con mayor dificul-*

tad para que pueda avanzar ese conocimiento” (sic), “Es una muy buena herramienta para que la metodología no sea tan teorica, permite que los estudiantes podamos estar practicando lo visto en clase y retandonos con nuevos ejercicios, creo que falta integrar mas ejercicios con mayores niveles de dificultad” (sic). Otro de los indicadores identificados fue el que se refiera a la **curva de aprendizaje que tiene la herramienta** al enfrentarse a ella las primeras veces. En este indicador se obtuvieron 9 respuestas que hacen referencia a la dificultad que presentan los estudiantes al hacer experimentar con la herramienta las primeras ocasiones de su aprendizaje. Algunas de las respuestas relacionadas a este indicador son: “aunque a veces esta herramienta puede tornar complicada para iniciar en la programacion con la practica se puede avanzar y obtener buenas notas” (sic), “Al principio se me dificulto entender algunos temas de la herramienta y los errores que generaba pero con el tiempo mejoro, la herramienta se dificulta en algunas cosas pero en cuanto a los temas ayudaba a ponerlos en practica” (sic), “fue una herramienta que llevo tiempo dominar y despues de esto con buena practica se pueden ver resultados” (sic), “me gusta la programacion pero al iniciar en la herramienta se torna dificil aunque a medida que se hacen ejercicios ya mejora” (sic).

Otro de los indicadores del concepto de **mejoras** fue el relacionado con la **curva de aprendizaje que tiene la temática** identificada a partir de 7 opiniones recolectadas. En este indicador los estudiantes expresaban la dificultad que presentaban al enfrentarse a las temáticas de la asignatura cuando han sido pocas o nulas las experiencias con el campo de la programación de computadores, sin embargo, opinan que la herramienta les brinda la seguridad que requieren para enfrentar estos retos presentes. Algunas opiniones relacionadas en este indicador fueron: “Hubieron algunos çasosün tanto complicados que exigian mas atencion y tiempo que otros” (sic), “Me motive a aprender mas sobre la programación al ver que algunos ejercicios se me hicieron complicados al inicio y al utilizar la herramienta empezaron a facilitarsen” (sic), “Ayuda en generar confianza para cada ejercicio, al principio puede tornarse dificil pero a medida de que se practica se va adquiriendo mas seguridad” (sic), “En la herramienta realizamos ejercicios faciles y dificiles que permitian comprender las tematicas en diferentes niveles de dificultad” (sic). El indicador correspondiente a la sugerencia o posible mejora a partir de incluir una mayor **cantidad de ejercicios** en la herramienta VPL tuvo 2 opiniones las cuales fueron: “la herramienta me parecio sensacional la evaluacion automatica resulta muy buena para implementar en clases de programacion recomendaria poner muchos mas ejercicios con retos complicados para motivar a resolverlos” (sic), “Es una herramienta excelente para el aprendizaje y la motivacion para seguir explorando este campo, sin embargo se requieren mas ejercicios y con mayor dificultad para que pueda avanzar ese conocimiento” (sic). Finalmente, se identificó a partir de 2 opiniones la necesidad que expresan los estudiantes de **incluir en otras asignaturas** este tipo de herramientas de evaluación automática con el fin de poder seguir practicando y desarrollando habilidades de programación. Las respuestas obtenidas en este indicador fueron: “Esta herramienta es bastante interesante, me gusta que permita ejecutar el codigo y ver si cumple con la solucion, seria chevere tenerla en otras asignaturas de programacion no como parte de la clase si no como ayuda para practicar y ejercitar las habilidades de codificacion” (sic), “la herramienta me parecio sensacional la evaluacion automatica resulta muy buena para implementar en clases de programacion recomendaria poner muchos mas ejercicios con retos complicados para motivar a resolverlos” (sic).

Para la encuesta de percepción se planteó una pregunta de opción múltiple con múltiple respuesta (como se observa en la pregunta 15 de la Tabla 4-7) con el fin de indagar sobre las características de la herramienta de evaluación automática VPL que más aceptación, gusto e interés tuvieron entre los estudiantes. Las características consultadas para evaluar y analizar fueron las siguientes:

- **Característica 1:** El entorno de desarrollo integrado.
- **Característica 2:** La posibilidad de ejecutar el código en la misma herramienta.
- **Característica 3:** El autocompletado de código fuente, tanto en variables como en estructuras de control.
- **Característica 4:** La ayuda en la sintaxis que brinda la herramienta a la hora de evaluar.
- **Característica 5:** La cantidad ilimitada de intentos para evaluar en cada uno de los ejercicios.
- **Característica 6:** La posibilidad de verificar errores al momento de evaluar el código.
- **Característica 7:** La retroalimentación inmediata que brinda la herramienta cuando el código no cumple con los casos esperados de prueba.
- **Característica 8:** La evaluación automática del código fuente.
- **Característica 9:** La posibilidad de ver los errores cometidos en el programa.
- **Característica 10:** La ayuda en la visualización del ejercicio (condiciones de entrada y salida) dentro del entorno de desarrollo.
- **Característica 11:** La posibilidad de guardar el código fuente y continuarlo posteriormente.
- **Característica 12:** La posibilidad de practicar con los diferentes ejercicios de práctica que estaban en la plataforma.

En la Figura 5-9 se pueden observar la cantidad de respuestas obtenidas a través de la encuesta de percepción sobre VPL respecto a las características consultadas. Se puede evidenciar que un 90,5 % de las respuestas, correspondiente a 19 estudiantes, indicaron tanto la opción de poder ejecutar sus soluciones dentro de la herramienta como poder evaluar múltiples veces sus soluciones con el fin de determinar si este estaba correcto o no fueron de mayor agrado por los estudiantes, relacionadas con las características 2 y 8, respectivamente.

Un 61,9 % de las respuestas permitieron evidenciar que fue interesante para los estudiantes, por un lado, poder verificar y observar los diferentes errores presentados en las soluciones entregadas y, por otra parte, poder guardar su solución o su código fuente con el fin de terminarlo cuando se cuente con una mayor disponibilidad de tiempo. Un 52,4 % de las respuestas indicaron que fue gratificante

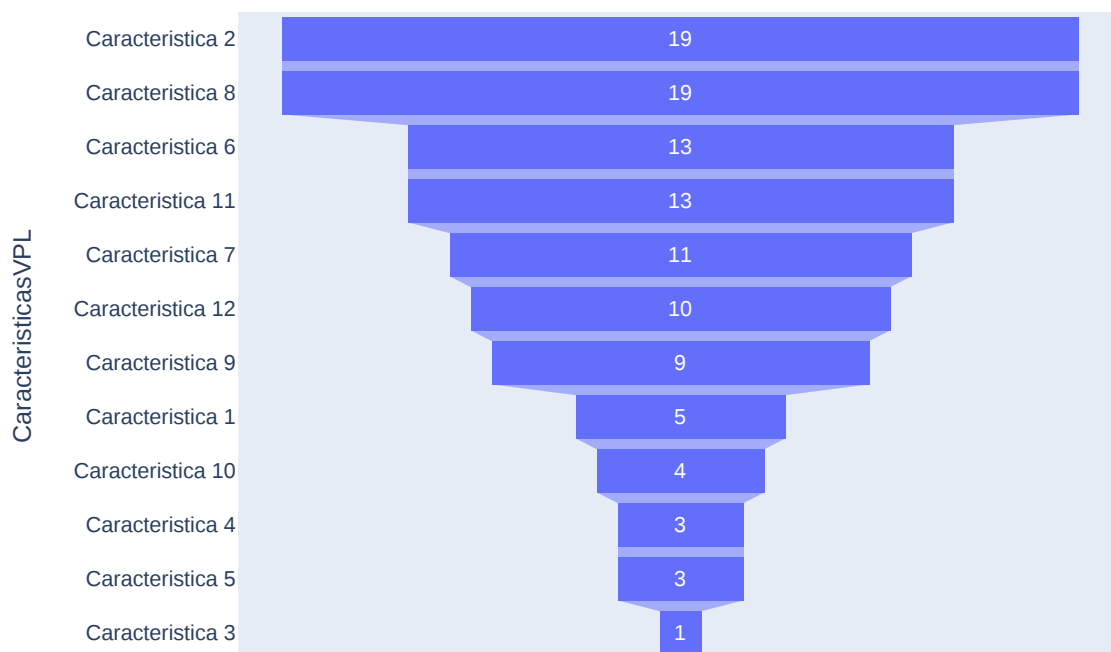


Figura 5-9.: Respuestas obtenidas sobre las características de VPL con mayor aceptación

contar con retroalimentación inmediata a la hora de comprobar sus soluciones con los casos de prueba planteados por el docente. Un 47,6 % de las respuestas evidenciaron que la posibilidad de tener múltiples ejercicios para practicar fue un valor agregado a la herramienta con la que los estudiantes encontraron un mayor agrado por la programación. Sin embargo, algunas de las características de VPL que no fueron de impacto para los estudiantes en su proceso de aprendizaje fueron las características 3, 4, 5 y 10, como se puede observar en el gráfico de la Figura 5-9. Estas cuentan apenas con un 4,8 %, 14,3 %, 14,3 % y 19 % de las respuestas obtenidas, respectivamente. Esto indica que de 1 a 4 personas les interesó esto en la herramienta, por lo que podría decirse que los estudiantes no le encontraron mayor utilidad a la función del autocompletado de código que tiene directamente la herramienta. De igual manera, tampoco le vieron una amplia funcionalidad a obtener una cantidad ilimitada de intentos a la hora de enviar sus soluciones. Así mismo, tan solo 3 estudiantes le hallaron relevancia a la ayuda que brinda la herramienta en la corrección de sintaxis y marcado de errores respecto a este mismo aspecto dentro de la herramienta.

6. Discusión de resultados

A través del diseño de investigación planteado y mediante los resultados que se obtuvieron a través de la encuesta de autoinforme MSLQ-Colombia y la encuesta de percepción sobre VPL, se pudo dar respuesta a la pregunta de investigación inicialmente propuesta en este estudio: ¿Cuál es el impacto en la motivación en el aprendizaje de los estudiantes a través de la integración de una herramienta de evaluación automática en un curso de programación de computadores?.

Los resultados que se obtuvieron permitieron evidenciar que la intervención de una herramienta de evaluación automática en el contexto académico de las clases de programación de computadores tuvo un impacto positivo en el proceso de enseñanza y aprendizaje de los estudiantes. Si bien, cuantitativamente no hubo una diferencia significativa, quizás debido a la muestra poblacional reducida, sí se puede percibir que existe una tendencia hacia una posible mejora en lo que respecta a las sub-escalas de la motivación relacionadas con las creencias sobre el control del aprendizaje y la valoración de la tarea. A partir de las opiniones brindadas por los estudiantes, se puede identificar que los estudiantes empezaron a apropiarse de su propio aprendizaje a través de una dedicación y estudio autónomo y un mayor agrado o interés en las temáticas de la asignatura gracias a la intervención y práctica brindada por VPL. Adicionalmente, estas dos sub-escalas fueron unas de las que mayor aceptación tuvieron dentro de la encuesta de percepción, por lo cual, puede atribuirse a un incremento en los niveles de atención y dedicación de los estudiantes en los conceptos propios de la materia y así mismo en su gusto de manera genuina en el aprendizaje y el desarrollo de habilidades de programación de computadores a partir de los ejercicios y actividades propuestas en la herramienta VPL.

En aspectos generales, al analizar los resultados cuantitativos iniciales, obtenidos a través del Pre-Test, se puede evidenciar que los participantes del estudio presentaban índices elevados de motivación, teniendo quizás como uno de los factores más bajos, la ansiedad presente en los procesos de evaluación. Sin embargo, la mayoría de los estudiantes en los dos grupos participantes, tanto el grupo experimental como el grupo control manifestaban sentirse identificados, de ligera a completamente, con las afirmaciones planteadas por el cuestionario del MSLQ-Colombia respecto a cada una de las sub-escalas de la motivación en el aprendizaje.

En cuanto a los resultados obtenidos por el Post-Test del grupo de control perteneciente al programa de Ingeniería de Telecomunicaciones, se pudo evidenciar que hubo un aumento leve en la cantidad de estudiantes que indicaron sentirse entre totalmente de acuerdo con las afirmaciones correspondientes a las sub-escalas de creencias sobre el control del aprendizaje y las expectativas de autoeficacia

en el rendimiento. A partir de esto se puede inferir que, si bien todas las valoraciones indicadas se mantuvieron estables respecto al Pre-Test como se esperaba, hubo una mejora en cuanto a la seguridad de aprender a través del esfuerzo personal y la dedicación autónoma, lo que a su vez permitía aumentar su confianza frente al éxito en sus procesos en las evaluaciones y/o parciales.

Respecto al grupo experimental, perteneciente al programa de Ingeniería de Sistemas, donde se integró la herramienta de evaluación automática VPL en las sesiones de clase por un tiempo de 8 semanas, no se halló una diferencia estadística significativa. Sin embargo, se evidencia un ligero aumento en las valoraciones correspondientes a las sub-escala motivacionales de las creencias sobre el control del aprendizaje y la valoración de la tarea. Esto se atribuye de igual manera a que los estudiantes, por un lado, perciben que el proceso de aprendizaje depende del estudio autónomo, del esfuerzo personal y la dedicación en las temáticas de la asignatura con el apoyo de la herramienta VPL y no de factores como el docente o la metodología de clase. Por otro lado, los estudiantes asocian niveles más altos en el gusto o el interés por la programación de computadores o las temáticas de la asignatura, gracias al uso de la herramienta de evaluación automática VPL.

A partir de los datos cualitativos obtenidos mediante las preguntas en escala de Likert de la encuesta de percepción aplicada únicamente en el grupo experimental, se pudo evidenciar que a partir de la integración de VPL los aspectos motivacionales más destacados por la mayoría de participantes en el estudio fueron las relacionadas con la orientación a metas intrínsecas y con la ansiedad frente a procesos de evaluación. Por un lado, respecto a la primera más destacada se puede relacionar con el hecho de que los estudiantes, por medio del uso de VPL, experimentarán un genuino deseo por aprender, encontrando los temas desafiantes o retadores, lo cual genera un gusto o motivación por comprender los diferentes temas de la asignatura. Por otra parte, respecto al segundo aspecto más destacado se puede relacionar con el hecho de que los estudiantes, al utilizar VPL para practicar o ejercitar sus habilidades, adquieren mayor seguridad y confianza al momento de presentar un quiz o parcial. Los estudiantes a través de las diferentes opiniones encontraron a VPL como una herramienta importante y útil en su proceso de motivación por aprender, indicando que a partir de los ejercicios y la práctica tanto en clase como extraclase empezaron a complementar el aprendizaje adquirido en las sesiones de clase. Igualmente, encontraron la herramienta de interés para su aprendizaje puesto que generaba en ellos un deseo de aprender por medio del estudio autónomo y de la dedicación personal que cada uno de los estudiantes le brindaban a las diferentes temáticas de la asignatura.

Sin embargo, a partir de las respuestas obtenidas mediante las preguntas abiertas de la encuesta de percepción, se evidenció que una mayor parte de las opiniones de los estudiantes estuvieron relacionadas con los constructos motivacionales de las expectativas de autoeficacia para el rendimiento y para el aprendizaje junto a las relacionadas con las creencias del control del aprendizaje y la orientación a metas intrínsecas. Este proceso permitió la agrupación de conceptos e indicadores en el análisis cualitativo.

A pesar de no haberse hallado una diferencia estadística significativa respecto a estas sub-escalas por medio de los datos cuantitativos, estos datos se pudieron contrarrestar con los datos cualitativos permitiendo afirmar que una sub-escala motivacional afectada positivamente fue la relacionada con las creencias sobre el control del aprendizaje. De igual manera, los estudiantes mediante las diversas opiniones destacaron que la herramienta VPL era muy interesante a la hora de la integración en la clase, puesto que les permitía practicar con múltiples ejercicios y como resultado les ayudaba a obtener mejores calificaciones, a su vez de que les brindaba más seguridad y confianza para obtener buenas notas en los procesos evaluativos. Así mismo, les brindaba más confianza y seguridad para aprender las diferentes temáticas sin importar la dificultad, puesto que complementaban las sesiones de clase con trabajo autónomo y esfuerzo personal que les permitía comprender de mejor manera las temáticas planteadas. También, la herramienta VPL les permitía motivarse por aprender a partir de los retos generados y los ejercicios que aumentaban el interés por investigar, o buscar información hasta resolver los desafíos propuestos. En la investigación realizada por Daradoumis *et al* (2022), en la cual se integró una herramienta de evaluación automática y se analizó el impacto que esta genera en los constructos relacionados con las expectativas de la autoeficacia y la motivación intrínseca mediante la aplicación del cuestionario MSLQ y la realización de un análisis cuantitativo con instrumentos estadísticos, coincide con el presente estudio puesto que no se obtuvo una diferencia significativa en las sub-escalas de la motivación mencionadas. Sin embargo, a pesar del hecho de no reportar efectos significativos, las investigaciones coinciden en obtener un alto número de percepciones de los estudiantes en donde se relacionan los aportes que brindan estas herramientas en estas sub-escalas motivacionales.

En cuanto a los resultados del análisis cualitativo relacionados con las metas intrínsecas y el verdadero interés de aprender por parte de los estudiantes, el estudio realizado tiene resultados similares a los de la investigación de Cheng *et al* (2021), los cuales integraron una herramienta de evaluación automática en una clase de programación en la cual se encontraban tanto principiantes como estudiantes experimentados en programación. Los estudiantes indicaron que los diferentes elementos de las herramientas de evaluación automática tales como: la retroalimentación automática, la calificación inmediata o los retos y ejercicios planteados les motivaban a seguir aprendiendo y les aumentaba el interés por comprender completamente cada temática. En ese estudio se encontraron diferencias significativas en el grupo experimental respecto a este constructo motivacional. Sin embargo, en ambas investigaciones se puede expresar, tal como indica el autor, que la motivación aumentó en los estudiantes con cierta experiencia en programación, mientras que en los estudiantes principiantes se mostró una disminución significativa en la actitud y la motivación en el aprendizaje, por lo cual sería interesante investigar este comportamiento. Así mismo, tal como indica Tarek *et al* (2022) en la revisión de literatura, se requieren enfocar las investigaciones en comprender los efectos que pueden tener estas herramientas en los deseos genuinos de aprender de los estudiantes y en mejorar los procesos educativos con vista a llevar a cabo un aprendizaje significativo en el estudiante.

Por otro lado, a partir de la encuesta de percepción se pudo detectar que los participantes del estudio

expresaban un compromiso con su propio aprendizaje a partir del estudio autónomo y el esfuerzo individual que le dedicaban a la asignatura, lo cual se evidenció en el gran número de opiniones obtenidas que se relacionaban con el deseo de aprender a partir del aprendizaje autónomo y la dedicación personal. Cabe recalcar que la sub-escala motivacional correspondiente a las creencias del control del aprendizaje no contó con ninguna diferencia estadística significativa, sin embargo, en las opiniones recolectadas fue la segunda sub-escala con más respuestas por parte de los estudiantes. Una gran parte de estas opiniones indicaban múltiples beneficios por parte de VPL en cuanto a esta sub-escala, indicando que la disponibilidad y la cantidad de ejercicios en la herramienta les propiciaba un entorno adecuado para el trabajo y aprendizaje autónomo.

Respecto a la sub-escala asociada con la ansiedad frente a los procesos de evaluación, muchos de los estudiantes plantearon diversas perceptivas en los datos cualitativos suministrados. Por una parte, mucho estudiantes expresaban que al practicar los diversos ejercicios con VPL podían preocuparse menos y contar con mayor seguridad ante cualquier quiz o parcial. Por otra parte, existían muchos estudiantes que aún seguían evidenciando una preocupación y desconfianza de su propio aprendizaje, por lo que en procesos de evaluación como los parciales eran los que obtenían generalmente resultados o calificaciones menores. *Qoiriah et al (2021)* menciona algo similar en su estudio indicando que la práctica y la constancia en esta área tan lógica como lo es la programación son fundamentales para disminuir la inseguridad o los miedos ante los quices y demás procesos formativos, sin embargo, no es clara la posición respecto a si es o no la herramienta el factor que genera ansiedad o realmente es la baja constancia en la práctica de cada estudiantes.

Igualmente uno de los conceptos más destacados a partir del análisis cualitativo es el relacionado con las expectativas de autoeficacia en el aprendizaje. Pese a que fue una sub-escala que no contó con diferencia significativa cuantitativamente, varios estudiantes a través de su opinión expresaron las ventajas que le brindaba VPL en el aumento de la confianza y seguridad para poder lograr entender, comprender y practicar las temáticas propuestas aún cuando fueren bastante difíciles. *Daradoumis et al (2022)* expresa un punto de vista parecido, en el cual, relaciona la integración de una herramienta de evaluación automática con propósito de facilitar la adquisición de conocimientos, influyendo de manera positiva en la autoeficacia que perciben los participantes al momento de enfrentarse a una nueva temática o concepto.

En algunos trabajos relacionados recientes como: *Cardoso, de Castro et al (2020)*, *Pringuet et al (2021)*, *Sousa Silva (2022)*, *Benotti et al (2018)*, *Jurado et al (2014)*, *Krusche et al (2017)*, *Barra et al (2020)*, la definición del concepto de motivación era abordada de manera amplia y en muchos casos se generalizaba o se mencionaba muy vagamente por los autores, además de que no se relacionaba con ninguno de los constructos motivacionales descritos en este estudio. Por lo cual, no se pudo realizar ningún proceso de comparación respecto a los efectos en la motivación del aprendizaje mediante la integración de herramientas de evaluación automática a partir de estas investigaciones.

Por otro lado, a través del análisis cualitativo realizado en este estudio, se pudo asociar las diferentes opiniones dentro de una categoría relacionada con el ambiente de evaluación de código fuente en el aprendizaje de la programación de computadores. Una gran parte de las opiniones recopiladas se asociaron a las características más destacadas de la herramienta VPL tal como: la evaluación formativa, la calificación inmediata, la retroalimentación automática y la disponibilidad. Los estudiantes indicaban que la posibilidad de poder ver sus errores sin necesidad de ser calificado numéricamente les permitía aprender a partir de los errores y les ayuda a apropiarse del conocimiento por medio de la práctica y la constancia. De igual manera, la ejecución y evaluación para probar las soluciones sin ninguna cantidad limitada de veces favorecía en los estudiantes la posibilidad de tener múltiples oportunidades con los cuales llegar a la solución correcta. En el estudio de Gordillo (2019) se pueden validar igualmente estas características mencionadas. En éste, se integra una herramienta de este tipo que permitió aumentar la motivación de los estudiantes y mejorar sus habilidades de programación, indicando que una de las características más importantes fue la retroalimentación que brindan este tipo de herramientas tecnológicas de evaluación automática.

De igual forma, también se tuvo en cuenta dentro de la agrupación en el análisis cualitativo la categoría correspondiente a metodologías de aprendizaje. En este se obtuvieron opiniones expresando la importancia de estas herramientas de evaluación automática en las clases de programación, destacando la ayuda y buen complemento que brinda a las clases. También se mencionaba la gran aceptación que se dio a partir de la integración de la metodología de aprendizaje basada en retos y problemas, sin dejar de lado la importancia del docente, como rol fundamental en el proceso de aprendizaje. Aunque estos sistemas de evaluación automático no pueden reemplazar el juicio humano ni validar muchas de los atributos de calidad no funcionales, si aportan un conjunto de ventajas, tal como lo indica Galan *et al* (2019). En el contexto universitario, estas metodologías que integran herramientas tecnológicas toman más territorio y permiten que a través de estrategias de aprendizaje innovadoras, la interacción entre docente y estudiante pueda darse de mejor manera, fomentando la participación de los estudiantes y llevan al estudiante a ser actor partícipe de su proceso de formación. Finalmente, es importante destacar que los demás conceptos identificados a partir de la encuesta de percepción en la fase de análisis cualitativo no fueron lo suficientemente relevantes por el bajo número de opiniones que agruparon.

La encuesta de percepción sobre VPL aplicada en el grupo experimental contaba con una pregunta de selección múltiple con múltiple respuesta en la que se buscaba conocer cuáles de los componentes de una herramienta de evaluación automática les interesó más a los estudiantes. El resultado indicó que al 90,5 % de los estudiantes les llamó la atención la posibilidad de ejecutar código fuente en la misma herramienta, de igual manera que la opción de poder evaluar sus soluciones las veces que sea necesario hasta que la calificación sea positiva. Esta es una de las características que se destacan en el estudio de Ullah *et al* (2018) indicando que, al igual que la retroalimentación automática, son factores que motivan e incrementan el interés de los estudiantes por aprender programación. Otra de las características destacadas por los estudiantes a partir de esa pregunta fue la posibilidad de conocer

sus propios errores y poder obtener retroalimentación aún antes de la calificación, por lo que se puede vincular con lo indicado por Marchisio *et al* (2018), quienes expresan la importancia de obtener comentarios y retroalimentación de forma automática y completa con el fin de apoyar el proceso de formación del estudiante.

Los resultados del presente estudio aportan evidencia empírica a las investigaciones relacionadas con el impacto de las herramienta de evaluación automática en la motivación de los estudiantes por aprender programación de computadores. Se pudo observar que VPL, si bien no tuvo cuantitativamente diferencias estadísticas significativas, sí evidenció un aumento en las valoraciones correspondientes al esfuerzo personal y dedicación individual por aprender las temáticas de la asignatura a su vez de la importancia, gusto e interés por estas. A partir de la integración de la herramienta se pudo evidenciar un aumento representativo en las valoraciones asociadas a las sub-escala motivacionales de las creencias sobre el control del aprendizaje y la valoración de la tarea recolectadas mediante el cuestionario MSLQ-Colombia. Así mismo, a través de la aplicación de la encuesta de percepción se recolectaron datos cualitativos que permitieron evidenciar el impacto que generó VPL en la seguridad y la confianza de los estudiantes al enfrentarse a los diferentes procesos evaluativos, a la confianza que les brindaba la herramienta para poder confiar en el aprendizaje personal de las diferentes temáticas y el desarrollo de habilidades y ejercicios a partir de un aprendizaje autónomo. Aún cuando no se pudieron evidenciar diferencias estadísticamente significativas en estos constructos motivacionales, a través del análisis cualitativo fue posible evidenciar que VPL influyó de manera positiva en distintos aspectos asociados a la motivación en el aprendizaje.

A partir de esta tesis de maestría se aporta y contribuye al campo de la educación en ingeniería en lo que respecta a la relación entre las herramientas de evaluación automática y el aumento en los niveles de la motivación dentro del proceso de enseñanza y aprendizaje de la programación de computadores soportado por ambientes virtuales educativos. Esta investigación buscaba validar el impacto de este tipo de herramientas, reportado por distintos autores y trabajos relacionados, en el proceso de enseñanza y aprendizaje de la programación. Si bien no se hallaron efectos cuantitativamente significativos en lo que respecta a las sub-escalas, es interesante analizar que sí se evidenciaron efectos cualitativamente importantes que indicaban una clara mejora en la motivación de los estudiantes respecto a la integración de VPL el del contexto académico de la programación de computadores.

7. Conclusiones y trabajos futuros

7.1. Conclusiones

En el actual estudio se llevó a cabo una investigación cuasi-experimental mediante la cual se integró la herramienta de evaluación automática VPL en los contenidos curriculares de la asignatura de “Algoritmos y Programación” de la Fundación Universitaria UCompensar, con el fin de analizar el impacto que puede tener una herramienta de evaluación automática de tareas de programación en la motivación por aprender de los estudiantes. En esta investigación hicieron parte 40 estudiantes distribuidos en dos grupos de la siguiente forma: 19 participantes en el grupo control pertenecientes al programa de Ingeniería de Telecomunicaciones y 21 estudiantes en el grupo experimental pertenecientes al programa de Ingeniería de Sistemas. En el grupo control se siguió la metodología tradicional de aprendizaje de las asignatura de programación de computadores, mientras que en el grupo experimental se integró la herramienta VPL en las sesiones de clase a lo largo de 8 semanas. Se aplicó el instrumento de autoinforme MSLQ-Colombia en dos lapsos de tiempo, antes de la intervención de la herramienta (Pre-Test) y después de la integración (Post-Test), con el propósito de caracterizar los diferentes aspectos motivacionales en el aprendizaje y determinar si estadísticamente existe alguna diferencia significativa tras la intervención. Asimismo, se realizó la aplicación de una encuesta de percepción sobre la herramienta VPL en el grupo experimental, con el fin de entender, contrarrestar y explicar los resultados obtenidos cuantitativamente. A partir de la intervención planteada en el estudio, se recolectaron y analizaron datos tanto cuantitativos como cualitativos relacionados con los niveles de motivación por aprender experimentados por los estudiantes participantes en el proceso investigativo.

Los datos de tipo cuantitativo que se obtuvieron mediante la aplicación de la encuesta MSLQ-Colombia, permitieron concluir que los participantes del grupo experimental, en donde se realizó la intervención de la herramienta VPL, no presentaron ninguna diferencia estadísticamente significativa en las sub-escalas de la motivación en el aprendizaje, sin embargo se pudo evidenciar un ligero aumento en la cantidad de estudiantes que indicaron estar totalmente de acuerdo con las afirmaciones correspondientes a las sub-escalas motivacionales relacionadas con las creencias sobre el control del aprendizaje y la valoración de la tarea. A través de los datos cualitativos recolectados mediante la encuesta de percepción se pudo evidenciar que la herramienta VPL tuvo un aporte importante en el aumento en los niveles de interés o gusto por las temáticas planteadas en la asignatura. De igual manera, se puede evidenciar que VPL tuvo un papel fundamental a la hora de brindar mayor seguridad y confianza en los estudiantes al momento de enfrentarse antes cualquier tipo de proceso evaluativo como quices o exá-

menes parciales. Adicionalmente, con los datos recolectados se pudo evidenciar que los estudiantes destacaron que VPL aporta favorablemente al aprendizaje a través del aprendizaje basado en retos o el aprendizaje basado en problemas, y en general, lo consideraron un complemento importante para las temáticas y un apoyo valioso para la práctica y ejercitación de las habilidades de programación.

A través de los resultados de este trabajo se logró contrastar y aportar a los hallazgos encontrados en investigaciones o estudios previos donde se observaban resultados que implicaban un aumento en los niveles de motivación mediante el uso de herramientas de evaluación automática en ámbitos educativos. El impacto hallado por medio de la intervención de la herramienta permitió aportar evidencia relacionada con el efecto que brinda la integración de técnicas de evaluación automática en la motivación del aprendizaje de los estudiantes de programación particularmente. Aunque no existió una diferencia significativa estadísticamente, se pudo evidenciar que, tanto en el análisis cualitativo como en el análisis cuantitativo, se asociaron una mayor cantidad de valoraciones con el constructo motivacional relacionado con las creencias sobre el control del aprendizaje, pues los estudiantes consideraban que el aprendizaje dependía de su esfuerzo, disciplina y dedicación por aprender las temáticas planteadas y no del docente o las estrategias propuestas. Por lo tanto, estos resultados permiten comprender de mejor manera el impacto que puede generar la integración de herramientas de evaluación automática en la motivación en los estudiantes por aprender aspectos de programación de computadores.

Finalmente, es importante considerar algunos factores que podrían ser considerados como amenazas a la validez del estudio. Por un lado, al contar con un número reducido de participantes, el tamaño de la muestra podría ser considerado como una limitación al trabajo actual, por lo que sería recomendable que en estudios futuros sobre este campo de la motivación en el aprendizaje se contará con una mayor cantidad de participantes. Por otro lado, el hecho de que los dos grupos participantes hicieran parte de distintos programas de formación podría ser una limitante teniendo en cuenta los enfoques de sus áreas de estudio y los conocimientos previos que podrían traer los estudiantes con respecto a los conceptos de programación, por lo que sería recomendable validar si esto podría llegar a influir en los resultados reportados en el presente estudio.

7.2. Publicación

Los resultados del presente estudio han sido aceptados para la divulgación a través de la Conferencia Internacional de Educación, Investigación e Innovación (ICERI 2022) realizada en la ciudad de Sevilla, España. El artículo se encuentra en construcción para su divulgación:

- Lozano-Rojas, H.D., Restrepo-Calle, F. y Ramírez-Echeverry J.J. (2022). Effects of an automatic assessment tool on the motivation to learn of computer programming students, in 15th annual International Conference of Education, Research and Innovation (ICERI), 7-9 November, Sevilla (España). (Borrador)

7.3. Trabajos futuros

A través del análisis de los datos obtenidos a lo largo de la investigación actual, se pudo determinar la necesidad de profundizar en investigaciones relacionadas con la motivación en el aprendizaje desde las diferentes sub-escalas que componen este constructo, a partir de la intervención de herramientas de evaluación automática, teniendo presentes instrumentos de validación estadísticas, las cuales permitan entender los ámbitos en los que se asemejan o no a los resultados brindados a partir de esta investigación. Sería recomendable para futuros trabajos, poder contar con una muestra con un mayor número de participantes con el fin de eliminar esta amenaza a la validez, y así poder contar con una mayor cantidad de datos con los cuales profundizar el análisis y comprender en mayor detalle el impacto que puede llegar a generar la integración de una herramienta de evaluación automática en el proceso de enseñanza y aprendizaje de la programación computadores. Por otra parte, sería recomendable generar una réplica de la integración generada en este estudio en investigaciones de tipo longitudinales que puedan ayudar a comprender de forma más profunda el impacto de las herramientas de evaluación automática en la motivación del aprendizaje en un periodo de intervención más extenso al que se logró en este estudio.

Finalmente, se hace necesario comprender, por medio de distintos estudios ya sean experimentales o cuasi-experimentales, el impacto que puede llegar a generar la integración de herramientas de evaluación automática en otros ámbitos dentro del contexto educativo tal como: las metodologías de aprendizaje, el rendimiento académico, las estrategias de trabajo colaborativo, la deserción de estudiantes, entre otros.

A. Anexo: Ítems de la escala de motivación en el aprendizaje del instrumento MSLQ-Colombia

Tabla A-1.: Sub-escala motivacional: Valoración de la Tarea

Número del ítem	ítem
4	Considero que lo que estoy aprendiendo en esta asignatura es de mucha utilidad porque lo podré aplicar en otras asignaturas.
10	Considero muy importante para mí, aprender los temas de esta asignatura.
17	Estoy muy interesado en el contenido de esta asignatura.
22	Considero que los temas de esta asignatura son útiles para mi formación.
25	Me gusta la temática de esta asignatura.
26	Considero muy importante para mí, entender la temática de esta asignatura.

Tabla A-2.: Sub-escala motivacional: Orientación hacia metas intrínsecas

Número del ítem	ítem
1	En esta asignatura, prefiero estudiar temas que sean desafiantes para mí con el fin de aprender cosas nuevas.
16	En esta asignatura, prefiero estudiar temas que despierten mi curiosidad aunque sean difíciles de aprender.
23	Cuando en esta asignatura tengo la oportunidad de escoger, elijo los trabajos o actividades con los que puedo aprender cosas nuevas aunque no me garanticen buenas calificaciones.

Tabla A-3.: Sub-escala motivacional: Orientación hacia metas extrínsecas

Número del ítem	ítem
7	Obtener una buena calificación en esta asignatura es la mayor satisfacción para mí en estos momentos.
11	Mi meta más importante ahora es mejorar mi promedio académico de la carrera, por eso mi preocupación principal en esta asignatura es obtener buenas calificaciones.
13	Mi meta es que mi nota definitiva en esta asignatura sea de las mejores del grupo.
29	Quiero obtener un buen desempeño académico en esta asignatura para demostrar a los demás mis capacidades.

Tabla A-4.: Sub-escala motivacional: Creencias sobre control del aprendizaje

Número del ítem	ítem
2	Pienso que depende de mí estudiar de manera apropiada, esto me llevará a aprender en esta asignatura.
9	Pienso que de mí depende aprender en esta asignatura.
18	De mi esfuerzo depende lo que aprenderé en esta asignatura.
24	Pienso que el nivel de comprensión del tema de esta asignatura depende de mí.

Tabla A-5.: Sub-escala motivacional: Expectativas de auto-eficacia en el rendimiento

Número del ítem	ítem
5	Confío en que obtendré una buena calificación definitiva en esta asignatura.
20	Espero presentar excelentes exámenes y trabajos en esta asignatura.
21	Espero tener un buen desempeño académico en esta asignatura.
30	Considerando la dificultad de la asignatura, los profesores y mis habilidades, pienso que tendré un buen desempeño académico en esta asignatura.

Tabla A-6.: Sub-escala motivacional: Expectativas de auto-eficacia en el aprendizaje

Número del ítem	ítem
6	Estoy convencido que soy capaz de entender incluso los temas más difíciles que presentan los libros recomendados para la asignatura.
12	Confío en que soy capaz de entender los conceptos principales que se enseñan en esta asignatura.
15	Confío en que soy capaz de entender hasta los temas más complicados que explique el profesor en esta asignatura.
28	Estoy convencido de ser capaz de dominar las habilidades que se enseñan en esta asignatura.

Tabla A-7.: Sub-escala motivacional: Ansiedad en los procesos de evaluación

Número del ítem	ítem
3	Cuando presento exámenes en esta asignatura, pienso que lo estoy haciendo mal en comparación con la forma en que lo hacen otros estudiantes de la clase.
8	Cuando presento exámenes de la asignatura, me intranquilizo si hay preguntas que no sé responder.
14	Cuando presento una evaluación en esta asignatura, pienso en las consecuencias que tendría sacar una mala nota.
19	Siento preocupación y ansiedad cuando presento evaluaciones en esta asignatura.
27	Cuando presento exámenes en esta asignatura, mi ritmo cardíaco se acelera.

Bibliografía

- ACM & IEEE. (2009). *Curriculum Guidelines for Graduate Degree Programs in Software Engineering* (inf. téc.). ACM, IEEE.
- ACM & IEEE. (2013). *Computer Science Curricula 2013 Curriculum Guidelines for Undergraduate Degree Programs in Computer Science* (inf. téc.). Association for Computing Machinery (ACM), IEEE Computer Society. <https://doi.org/10.1145/2534860>
- ACM & IEEE. (2017). *Information Technology Curricula 2017 IT2017 Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology* (inf. téc.). <https://doi.org/10.1145/3173161>
- Ala-Mutka, K. M. (2005). A Survey of Automated Assessment Approaches for Programming Assignments. *Computer Science Education*, 15(2), 83-102. <https://doi.org/10.1080/08993400500150747>
- Álvarez, B., González, C. & García, N. (2008). La motivación y los métodos de evaluación como variables fundamentales para estimular el aprendizaje autónomo. *Red U. Revista de Docencia Universitaria*, 5(2), 1-12. <https://revistas.um.es/redu/article/view/3371/3271>
- Alves, N. D. C., Wangenheim, C. G. V., Hauck, J. C. R. & Borgatto, A. F. (2020). A large-scale evaluation of a rubric for the automatic assessment of algorithms and programming concepts. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 556-562. <https://doi.org/10.1145/3328778.3366840>
- Aristika, A., Darhim, Juandi, D. & Kusnandi. (2021). The Effectiveness of Hybrid Learning in Improving of Teacher-Student Relationship in Terms of Learning Motivation. *Emerging Science Journal*, 5(4). <https://doi.org/https://doi.org/10.28991/esj-2021-01288>
- Azmi, N. A., Mohd-Yusof, K., Phang, F. A. & Syed Hassan, S. A. H. (2018). Motivating Engineering Students to Engage in Learning Computer Programming. *Advances in Intelligent Systems and Computing*, 143-157. https://doi.org/0.1007/978-3-319-60937-9_12
- Barra, E., López-Pernas, S., Alonso, Á., Sánchez-Rada, J. F., Gordillo, A. & Quemada, J. (2020). Automated Assessment in Programming Courses: A Case Study during the COVID-19 Era. *Sustainability*, 12(18). <https://doi.org/10.3390/su12187451>
- Barrios, T. & Marin, M. B. (2014). Aprendizaje mixto a través de laboratorios virtuales. *Signos Universitarios*.
- Barros Barrios, R. J., Rojas Montero, J. A. & Sánchez Ayala, L. M. (2008). Diseño de instrumentos didácticos para aprendizaje activo basado en teoría de colores. *Revista Educación en Ingeniería*, 3(5), 11-18. <https://doi.org/10.26507/rei.v3n5.148>
- Becerra-Alonso, D., Lopez-Cobo, I., Gómez-Rey, P., Fernández-Navarro, F. & Barbera, E. (2020). EduZinc: A tool for the creation and assessment of student learning activities in complex open,

- online and flexible learning environments. *Distance Education*, 41(1), 86-105. <https://doi.org/10.1080/01587919.2020.1724769>
- Beltrán, J. A. (1993). Procesos, estrategias y técnicas de aprendizaje. *Síntesis*.
- Bennedsen, J. & Caspersen, M. E. (2019). Failure Rates in Introductory Programming: 12 Years Later. *ACM Inroads*, 10(2), 30-36.
- Benotti, L., Aloï, F., Bulgarelli, F. & Gomez, M. J. (2018). The effect of a web-based coding tool with automatic feedback on students' performance and perceptions. *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 2018-Janua*, 2-7. <https://doi.org/10.1145/3159450.3159579>
- Bosse, Y. & Gerosa, M. A. (2017). Difficulties of Programming Learning from the Point of View of Students and Instructors. *IEEE Latin America Transactions*, 15(11), 2191-2199. <https://doi.org/10.1109/TLA.2017.8070426>
- Brito, M. & Goncalves, C. (2019). Codeflex: A web-based platform for competitive programming. *Iberian Conference on Information Systems and Technologies, CISTI, 2019-June*. <https://doi.org/10.23919/CISTI.2019.8760776>
- Bryman, A. (2012). *Social Research Methods* (4a ed). Oxford University Press. <https://global.oup.com/academic/product/social-research-methods-9780190853662?q=Social%5C%20Research&lang=en&cc=co>
- Burgos-Castillo, E. & Sánchez-Abarca, P. (2012). "Adaptación y validación preliminar del cuestionario de motivación y estrategias de aprendizaje (MSLQ)". Universidad del Bío-Bío. Chillán, Chile.
- Byrne, P. & Lyons, G. (2001). The effect of student attributes on success in programming. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 49-52. <https://doi.org/10.1145/507758.377467>
- Cardoso, M., de Castro, A. V., Rocha, Á., Silva, E. & Mendonça, J. (2020). Use of Automatic Code Assessment Tools in the Programming Teaching Process. En R. Queirós, F. Portela, M. Pinto y A. Simões (Eds.), *First International Computer Programming Education Conference (ICPEC 2020)* (4:1-4:10). Schloss Dagstuhl-Leibniz-Zentrum für Informatik. <https://doi.org/10.4230/OASlcs.ICPEC.2020.4>
- Keywords: Teaching programming, APROG, Moodle, VPL, Mooshak, Automatic assessment
- Cardoso, M., Marques, R., De Castro, A. V. & Rocha, Á. (2020). Using Virtual Programming Lab to improve learning programming: The case of Algorithms and Programming. *Expert Systems*, 38(4). <https://doi.org/10.1111/exsy.12531>
- Cheng, L.-C., Li, W. & Tseng, J. C. R. (2021). Effects of an automated programming assessment system on the learning performances of experienced and novice learners. *Interactive Learning Environments*, 0(0), 1-17. <https://doi.org/10.1080/10494820.2021.2006237>
- Chi, H., Allen, C. & Jones, E. (2016). Integrating Computing to STEM Curriculum via CodeBoard, 512-529. https://doi.org/10.1007/978-3-319-42089-9_36
- Chibizova, N. V. (2018). The Problems of Programming Teaching. *2018 4th International Conference on Information Technologies in Engineering Education, Inforino 2018 - Proceedings*. <https://doi.org/10.1109/INFORINO.2018.8581834>

- Christian, M. & Trivedi, B. (2016). A comparison of existing tools for evaluation of programming exercises. *ACM International Conference Proceeding Series*, 04-05-Marc. <https://doi.org/10.1145/2905055.2905350>
- Clifton, J. (2010). A Simple Judging System for the ACM Programming Contest. *Computer Science and Software Engineering, University of Wisconsin – Platteville*. http://micsymposium.org/mics_2010_proceedings/mics2010_submission_7.pdf
- Codeboard. (2020). Recuperado desde: <https://codeboard.io/>.
- Combéfis, S. & de Moffarts, G. (2019). Automated Generation of Computer Graded Unit Testing-Based Programming Assessments for Education. *6th International Conference on Computer Science, Engineering and Information Technology (CSEIT-2019)*. <https://doi.org/10.5121/csit.2019.91308>
- Combéfis, S. & Saint-Marcq, V. (2012). Teaching Programming and Algorithm Design with Pythia, a Web-Based Learning Platform. 6, 31-43.
- Contreras, E. (2004). Evaluación de los aprendizajes universitarios. *Docencia universitaria. Orientaciones para la formación del profesorado*, 129-152.
- Contreras, R., Sierra, E. A., Hernández, H. D. R., Hernández, N. B. E. & Moyotl, V. J. H. (2020). Sistema de evaluación inteligente para medir habilidades de razonamiento matemático. *Revista Iberoamericana de Evaluación Educativa*, 13(1), 251-280.
- Coolican, H. (1997). *Métodos de investigación y estadística en psicología* (2a ed). Editorial Manual Moderno, S.A. de C.V.
- Correia, H., Leal, J. P. & Paiva, J. C. (2017). Improving diagram assessment in Mooshak. *International Conference on Technology Enhanced Assessment*, 69-82.
- Croft, D. & England, M. (2019). Computing with CodeRunner at Coventry University Automated summative assessment of Python and C++ code. *ACM International Conference Proceeding Series*, 1-4. <https://doi.org/10.1145/3372356.3372357>
- Dal Faro, N., Cuenca Pletsch, L. & Maurel, M. (2008). La utilización del Blendend-Learning como aporte a la construcción de conocimientos significativos para los alumnos de Ingeniería en Sistemas. *Primera Conferencia Latinoamericana sobre el Abandono en la Educación Superior*.
- Daradoumis, T., Marqués Puig, J. M., Arguedas, M. & Calvet Liñan, L. (2022). Enhancing students' beliefs regarding programming self-efficacy and intrinsic value of an online distributed Programming Environment. *Journal of Computing in Higher Education*. <https://doi.org/10.1007/s12528-022-09310-9>
- Darejeh, A. & Salim, S. S. (2016). Gamification Solutions to Enhance Software User Engagement—A Systematic Review. *International Journal of Human-Computer Interaction*, 32(8), 613-642. <https://doi.org/10.1080/10447318.2016.1183330>
- De Oliveira Brandão, L., Bosse, Y. & Gerosa, M. A. (2016). Visual programming and automatic evaluation of exercises: An experience with a STEM course. *Proceedings - Frontiers in Education Conference, FIE, 2016-Novem*. <https://doi.org/10.1109/FIE.2016.7757621>
- Del Valle, S. (2004). La programación y las unidades didácticas en Secundaria Obligatoria. *Curso CSI-CSIF Sector de enseñanza*.

- Derval, G., Cégo, A. & Reinbold, P. (2014). *INGINIOUS [software]*. Recuperado desde: <https://github.com/UCL-INGI/INGInious>.
- DomJudge-Online. (2022). Recuperado desde: <https://www.domjudge.org/>.
- Edwards, S. H. & Pérez-Quiñones, M. A. (2008). Web-CAT: Automatically grading programming assignments. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 328. <https://doi.org/10.1145/1384271.1384371>
- Elliott, S. W. (2017). *Computers and the Future of Skill Demand*. OECD Publishing.
- Escamilla, J., Fuentes, K., Venegas, E., Fernández, K., Elizondo, J. & Román, R. (2016). EduTrends Gamificación. *Observatorio de Innovación Educativa*, 1-36. <https://eduteka.icesi.edu.co/pdfdir/edutrends-gamificacion.pdf>
- Fernández, P., Vallejo, G., Livacic-Rojas, P. & Tuero, E. (2014). Validez Estructurada para una investigación cuasi-experimental de calidad. Se cumplen 50 años de la presentación en sociedad de los diseños cuasi-experimentales. *An. psicol.*, 30(2).
- Galan, D., Heradio, R., Vargas, H., Abad, I. & Cerrada, J. A. (2019). Automated Assessment of Computer Programming Practices: The 8-Years UNED Experience. *IEEE Access*, 7, 130113-130119. <https://doi.org/10.1109/ACCESS.2019.2938391>
- Gallardo, K. (2021). The Importance of Assessment Literacy: Formative and Summative Assessment Instruments and Techniques. En R. Babo, N. Dey y A. S. Ashour (Eds.), *Workgroups eAssessment: Planning, Implementing and Analysing Frameworks* (pp. 3-25). Springer Singapore. https://doi.org/10.1007/978-981-15-9908-8_1
- Gallego-Romero, J. M., Alario-Hoyos, C., Estévez-Ayres, I. & Delgado Kloos, C. (2020). Analyzing learners' engagement and behavior in MOOCs on programming with the Codeboard IDE. *Educational Technology Research and Development*. <https://doi.org/10.1007/s11423-020-09773-6>
- García, R., Falkner, K. & Vivian, R. (2018). Systematic literature review: Self-Regulated Learning strategies using e-learning tools for Computer Science. *Computers and Education*, 123, 150-163. <https://doi.org/10.1016/j.compedu.2018.05.006>
- García Méndez, M. & Rivera Aragón, S. (2012). *Aplicación de la estadística a la psicología*. Miguel Ángel Porrúa. <https://maporra.com.mx/product/aplicacion-de-la-estadistica-a-la-psicologia-3>
- Garcia-Duncan, T. & McKeachie, W. J. (2005). The Making of the Motivated Strategies for Learning Questionnaire. *Educational Psychologist*, 40(2), 117-128. https://doi.org/10.1207/s15326985ep4002_6
- Gatica-Saavedra, M. & Rubí-González, P. (2020). The master class in the context of the competency-based educational model. *Revista Electrónica Educare*, 25(1), 1-12. <https://doi.org/10.15359/ree.25-1.17>
- Gomes, A. & Mendes, A. (2015). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. *Proceedings - Frontiers in Education Conference, FIE*. <https://doi.org/10.1109/FIE.2014.7044086>
- González Jaimes, E. I., López Chau, A., Trujillo Mora, V. & Rojas Hernández, R. (2018). Estrategia didáctica de enseñanza y aprendizaje para programadores de software / Teaching and learning

- didactic strategy for software programmers. *RIDE Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 9(17), 688-712. <https://doi.org/10.23913/ride.v9i17.402>
- González-Carrillo, C. D., Restrepo-Calle, F., Ramírez-Echeverry, J. J. & González, F. A. (2021a). Automatic Grading Tool for Jupyter Notebooks in Artificial Intelligence Courses. *Sustainability*, 13(21). <https://doi.org/10.3390/su132112050>
- González-Carrillo, C. D., Restrepo-Calle, F., Ramírez-Echeverry, J. J. & González, F. A. (2021b). Automatic Grading Tool for Jupyter Notebooks in Artificial Intelligence Courses. *Sustainability*, 13(21). <https://doi.org/10.3390/su132112050>
- Gonzalez-Escribano, A., Lara-Mongil, V., Rodriguez-Gutierrez, E. & Torres, Y. (2019). Toward improving collaborative behaviour during competitive programming assignments. *2019 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*, 68-74. <https://doi.org/10.1109/EduHPC49559.2019.00014>
- Gordillo, A. (2019). Effect of an instructor-centered tool for automatic assessment of programming assignments on students' perceptions and performance. *Sustainability (Switzerland)*, 11(20). <https://doi.org/10.3390/su11205568>
- Grover, S. (2021). Toward A Framework for Formative Assessment of Conceptual Learning in K-12 Computer Science Classrooms. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 31-37. <https://doi.org/10.1145/3408877.3432460>
- Guerrero, M., Guamán, D. S. & Caiza, J. C. (2015). Revisión de Herramientas de Apoyo en el Proceso de Enseñanza-Aprendizaje de Programación. *Revista Politécnica*, 35(1), 84. https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/430
- Gupta, S. & Gupta, A. (2018). E-Assessment Tools for Programming Languages: A Review. *First International Conference on Information Technology and Knowledge Management*, 65-70. <https://doi.org/10.15439/2018KM31>
- Hamidah, J., Said, I. & Ratnawati. (2019). Implementing Blended Learning Toward Students' Self Efficacy in Writing Class: Students and Teachers' Voice. *Journal of English Education and Teaching (JEET)*. <https://doi.org/0.33369/jeet.3.3.270-286>
- Hernández Sampieri, R., Fernández Collado, C. & Baptista Lucio, M. d. P. (2016). *Metodología de la investigación* (6a. ed). México D.F.: McGraw-Hill.
- Ibrahim, M. M. & Nat, M. (2019). Blended learning motivation model for instructors in higher education institutions. *International Journal of Educational Technology in Higher Education*, 16(1), 12. <https://doi.org/10.1186/s41239-019-0145-2>
- Ihantola, P., Ahoniemi, T., Karavirta, V. & Seppälä, O. (2010). Review of recent systems for automatic assessment of programming assignments. *Proceedings of the 10th Koli Calling International Conference on Computing Education Research, Koli Calling'10*, 86-93. <https://doi.org/10.1145/1930464.1930480>
- Ion, G., Sánchez Martí, A. & Agud Morell, I. (2019). Giving or receiving feedback: which is more beneficial to students' learning? *Assessment & Evaluation in Higher Education*, 44(1), 124-138. <https://doi.org/10.1080/02602938.2018.1484881>

- Janczewski, R., Kosowski, A., Malafiejski, M. & Noinski, T. (2006). Application of SPOJ cooperative contest management in the university tuition system. *Annals of the Gdansk University of Technology*.
- Janičić, M. V. & Marić, F. (2020). Regression verification for automated evaluation of students programs. *Computer Science and Information Systems*, 17(1), 205-227. <https://doi.org/10.2298/CSIS181220019V>
- Järvelä, S. & Niemivirta, M. (2001). Motivation in context: Challenges and possibilities in studying the role of motivation in new pedagogical culture. En S. Volet y S. Järvelä (Eds.), *Motivation in Learning Contexts* (pp. 105-127). Pergamon Press.
- Jiménez-Toledo, J. A., Collazos, C. & Revelo-Sánchez, O. (2019). Consideraciones en los procesos de enseñanza-aprendizaje para un primer curso de programación de computadores: una revisión sistemática de la literatura. *TecnoLógicas*, 22, 83-117. <https://doi.org/10.22430/22565337.1520>
- Jurado, F., Redondo, M. & Ortega, M. (2014). eLearning standards and automatic assessment in a distributed eclipse based environment for learning computer programming. *Computer Applications in Engineering Education*, 22(4), 774-787. <https://doi.org/https://doi.org/10.1002/cae.21569>
- Kanika, Chakraverty, S. & Chakraborty, P. (2020). Tools and Techniques for Teaching Computer Programming: A Review. *Journal of Educational Technology Systems*, 49(2), 170-198. <https://doi.org/10.1177/0047239520926971>
- Keuning, H., Jeuring, J. & Heeren, B. (2016). Towards a Systematic Review of Automated Feedback Generation for Programming Exercises. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 41-46. <https://doi.org/10.1145/2899415.2899422>
- Khramova, M. V., Nesterov, M. V. & Kurkin, S. A. (2019). Problems of Learning Programming in Introductory Course. *International Conference "Quality Management, Transport and Information Security, Information Technologies"(IT&QM&IS)*, 522-525. <https://doi.org/10.1109/ITQMIS.2019.8928390>
- Kosowski, A., Malafiejski, M. & Noinski, T. (2008). Application of an Online Judge and Contester System in Academic Tuition. *Annals of the Gdansk University of Technology*.
- Krugel, J., Hubwieser, P., Goedicke, M., Striwe, M., Talbot, M., Olbricht, C., Schypula, M. & Zettler, S. (2020). Automated Measurement of Competencies and Generation of Feedback in Object-Oriented Programming Courses. *IEEE Global Engineering Education Conference (EDUCON '20)*, 1(1), 10.
- Krusche, S., von Frankenberg, N. & Affi, S. (2017). Experiences of a Software Engineering Course based on Interactive Learning.
- Law, K. M., Lee, V. C. & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers and Education*, 55(1), 218-228. <https://doi.org/10.1016/j.compedu.2010.01.007>
- Leal, J. P. & Silva, F. (2003). Mooshak: a Web-based multi-site programming contest system. *Software: Practice and Experience*, 567-581. <https://doi.org/10.1002/spe.522>

- Lishinski, A. & Yadav, A. (2019). Motivation, Attitudes, and Dispositions. En S. A. Fincher y A. V. Robins (Eds.), *The Cambridge Handbook of Computing Education Research* (pp. 801-826). Cambridge University Press. <https://doi.org/10.1017/9781108654555.029>
- Lobb, R. & Harlow, J. (2016). Coderunner: A tool for assessing computer programming skills. *ACM Inroads*, 7(1), 47-51. <https://doi.org/10.1145/2810041>
- Loui, M. C. & Borrego, M. (2019). Engineering Education Research. En S. A. Fincher y A. V. Robins (Eds.), *The Cambridge Handbook of Computing Education Research* (pp. 292-322). Cambridge University Press. <https://doi.org/10.1017/9781108654555.012>
- Lovos, E. & González, A. H. (2014). Moodle y VPL como Soporte a las Actividades de Laboratorio de un Curso Introductorio de Programación. *IX Congreso de Tecnología en Educación y Educación en Tecnología*.
- Manniam Rajagopal, M. B. (2018). *Virtual Teaching Assistant to Support Students' Efforts in Programming* (Tesis de maestría). Virginia Polytechnic Institute; State University.
- Marchisio, M., Barana, A., Fioravera, M., Rabellino, S. & Conte, A. (2018). A Model of Formative Automatic Assessment and Interactive Feedback for STEM. *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 01, 1016-1025. <https://doi.org/10.1109/COMPSAC.2018.00178>
- Monereo, C. (2014). Las estrategias de aprendizaje en la Educación formal: enseñar a pensar y sobre el pensar. *Infancia y Aprendizaje*, 13, 3-25. <https://doi.org/10.1080/02103702.1990.10822263>
- Moreira, M. A. (2017). Aprendizaje significativo como un referente para la organización de la enseñanza. *Archivos de Ciencias de la Educación*, 11(12). https://www.memoria.fahce.unlp.edu.ar/art_revistas/pr.8290/pr.8290.pdf
- Muñoz, R., Barría, M., Nöel, R., Providel, E. & Quiroz, P. (2012). Determinando las dificultades en el aprendizaje de la primera asignatura de programación en estudiantes de ingeniería civil informática. *Memorias del XVII Congreso Internacional de Informática Educativa, TISE*.
- Pagano, R. (2006). *Estadística para Las Ciencias Del Comportamiento*. Cengage Learning Latin America. <https://books.google.com/cu/books?id=zU1hmIJ4IrcC>
- Paredes-Daza, J. D. & Sanabria-Becerra, W. M. (2015). Ambientes de aprendizaje o ambientes educativos. Una reflexión ineludible. *Revista de Investigaciones UCM*, 25(15), 144-158. <https://doi.org/10.22383/ri.v15i1.39>
- Patil, A. (2010). *Automatic Grading of Programming Assignments* (Tesis de maestría). San Jose State University. <https://doi.org/0.31979/etd.vnt7-hgnd>
- Pérez Pino, M., Enrique Clavero, J. O., Carbó Ayala, J. E. & González Falcón, M. (2017). La evaluación formativa en el proceso enseñanza aprendizaje. *Edumecentro*, 9(3), 263-283.
- Pham, M. T. & Nguyen, T. B. (2019). The DOMJudge based Online Judge System with Plagiarism Detection. *The University of Danang - University of Science and Technology*.
- Pintrich, P., Smith, D., Garcia, T. & McKeachie, W. (1991). *A Manual for the Use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. <https://files.eric.ed.gov/fulltext/ED338122.pdf>
- Pringuet, P., Friel, A. & Vande Wiele, P. (2021). CodeRunner: A Case Study of the Transition to Online Learning of a Java Programming Course. *Proceedings of the AUBH E-Learning Conference 2021*:

- Innovative Learning and Teaching - Lessons from COVID-19*, 1-10. <https://doi.org/10.2139/ssrn.3874475>
- Keywords: Programming Education, Automated Assessment, CodeRunner, Student feedback, Applied computing, eLearning, Learning Management Systems
- Qoiriah, A., Yamasari, Y., Asmunin, Nurhidayat, A. I. & Harimurti, R. (2021). Exploring Automatic Assessment-Based Features for Clustering of Students' Academic Performance. En A. Abraham, Y. Ohsawa, N. Gandhi, M. Jabbar, A. Haqiq, S. McLoone y B. Issac (Eds.), *Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020)* (pp. 125-134). Springer International Publishing.
- Queirós, R. & Leal, J. P. (2012). PETCHA: A programming exercises teaching assistant. *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 192-197. <https://doi.org/10.1145/2325296.2325344>
- Queirós, R. & Leal, J. P. (2018). Fostering Students-Driven Learning of Computer Programming with an Ensemble of E-Learning Tools. En Á. Rocha, H. Adeli, L. P. Reis y S. Costanzo (Eds.), *Trends and Advances in Information Systems and Technologies* (pp. 289-298). Springer International Publishing.
- Ramírez-Dorantes, M. d. C., Echazarreta-Moreno, A., Bueno-Álvarez, J. A. & Canto-y-Rodríguez, J. E. (2013). Validación Psicométrica del Motivated Strategies for Learning Questionnaire en Universitarios Mexicanos. *Electronic Journal of Research in Educational Psychology*. <https://www.redalyc.org/articulo.oa?id=293125761009>
- Ramírez-Echeverry, J. J. (2017). *La competencia «aprender a aprender» en un contexto educativo de ingeniería* (Tesis doctoral). Universitat Politècnica de Catalunya (UPC). Barcelona, España.
- Ramírez-Echeverry, J. J., García-Carillo, A. & Olarte-Dussán, F. A. (2016). Adaptation and Validation of the Motivated Strategies for Learning Questionnaire-MSLQ-in Engineering Students in Colombia*. *International Journal of Engineering Education*, 32(4), 1-14. https://upcommons.upc.edu/bitstream/handle/2117/107554/ijee3275ns%7B%5C_%7Deditado%7B%5C_%7DIJEE.pdf
- Ramírez-Echeverry, J. J., Restrepo-Calle, F. & González, F. (2022). A case study in technology-enhanced learning in an introductory computer programming course. *Global Journal of Engineering Education*, 24(1), 65-71. <http://www.wiete.com.au/journals/GJEE/Publish/vol24no1/10-Restrepo-Calle-F.pdf>
- Ramírez-Echeverry, J. J., Rosales-Castro, L. F., Restrepo-Calle, F. & Gonzalez, F. A. (2018). Self-Regulated Learning in a Computer Programming Course. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(2), 75-83. <https://doi.org/10.1109/RITA.2018.2831758>
- Restrepo-Calle, F., Ramírez Echeverry, J. J. & González, F. A. (2019). Continuous assessment in a computer programming course supported by a software tool. *Computer Applications in Engineering Education*, 27(1), 80-89. <https://doi.org/10.1002/cae.22058>
- Restrepo-Calle, F., Ramírez-Echeverry, J. J. & Gonzalez, F. A. (2018). UNCODE: Interactive system for learning and automatic evaluation of computer programming skills. In *Proceedings of the 10th annual International Conference on Education and New Learning Technologies EDULEARN 2018*, 1, 6888-6898. <https://doi.org/10.21125/edulearn.2018.1632>

- Restrepo-Calle, F., Ramírez-Echeverry, J. J. & González, F. (2020). Using an interactive software tool for the formative and summative evaluation in a computer programming course: an experience report. *Global Journal of Engineering Education*.
- Revilla, M. A., Manzoor, S. & Liu, R. (2008). Competitive Learning in Informatics: The UVa Online Judge Experience. *Olympiads in Informatics, Institute of Mathematics and Informatics*.
- Rodríguez, J., Rubio Royo, E. & Hernández, Z. (2011). USES OF VPL. *INTED2011 Proceedings*, 743-748.
- Rodriguez-del-Pino, J. (2012). A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features.
- Rubio-Sánchez, M., Kinnunen, P., Pareja-Flores, C. & Ángel Velázquez-Iturbide, J. (2012). Lessons learned from using the automated assessment tool “Mooshak”. *2012 International Symposium on Computers in Education (SIIE)*, 1-6.
- Rubio-Sánchez, M., Kinnunen, P., Pareja-Flores, C. & Velázquez-Iturbide, Á. (2014). Student perception and usage of an automated programming assessment tool. *Computers in Human Behavior*, 31, 453-460. <https://doi.org/https://doi.org/10.1016/j.chb.2013.04.001>
- Ruiz-de-Clavijo, B. N. (2009). Motivación, motivación en el aprendizaje, acción motivacional del profesor en el aula. *Revista Digital Innovación y Experiencias Educativas*.
- Sangwin, C. (2019). Automatic assessment of students' code using CodeRunner. *University of Edinburgh*, 1-20.
- Sanmartín, V. A. G. & Pilco, W. V. Y. (2020). Aprender haciendo”: Aplicación de la metodología por ambientes de aprendizaje. *Polo del Conocimiento: Revista científico-profesional*, 5(7), 188-208.
- Seijo Galán, S., Freire Rodríguez, C. & Ferradás Canedo, M. d. M. (2020). Tipos de motivación en relación a la ansiedad ante los exámenes en el alumnado de educación primaria. *PUBLICACIONES*, 50(1), 265-274. <https://doi.org/10.30827/publicaciones.v50i1.14101>
- Shao, T., Kuang, Y., Huang, Y. & Quan, Y. (2019). PAAA: An implementation of programming assignments automatic assessing system. *ACM International Conference Proceeding Series*, 68-72. <https://doi.org/10.1145/3338147.3338187>
- Shivam, Goswami, N., Baths, V. & Bandyopadhyay, S. (2019). AES: Automated evaluation systems for computer programming course. *ICSOFT 2019 - Proceedings of the 14th International Conference on Software Technologies*, 508-513. <https://doi.org/10.5220/0007951205080513>
- Siegel, S. & Castellan, N. J. (1998). *Estadística no paramétrica: Aplicada a las ciencias de la conducta* (4a ed). Editorial Trillas. https://etrillas.mx/libro/estadistica-no-parametrica-aplicada-a-las-ciencias-de-la-conducta_2051
- Skalka, J., Drlík, M. & Obonya, J. (2019). Automated Assessment in Learning and Teaching Programming Languages using Virtual Learning Environment, 689-697. <https://doi.org/10.1109/EDUCON.2019.8725127>
- Sousa Silva, G. R. (2022). *Impact of a pseudocode online judge on programming language learning*. Universidade de Brasília.
- Souza, D. M., Felizardo, K. R. & Barbosa, E. F. (2016). A systematic literature review of assessment tools for programming assignments. *Proceedings - 2016 IEEE 29th Conference on Software Engineering Education and Training, CSEandT 2016*, 147-156. <https://doi.org/10.1109/CSEET.2016.48>

- Spacco, J., Hovemeyer, D., Pugh, W., Emad, F., Hollingsworth, J. K. & Padua-Perez, N. (2006). Experiences with Marmoset: Designing and Using an Advanced Submission and Testing System for Programming Courses. *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITICSE '06)*, pp. 13-17.
- Spacco, J., Strecker, J., Hovemeyer, D. & Pugh, W. (2005). Software Repository Mining with Marmoset: An Automated Programming Project Snapshot and Testing System. *Proceedings of the Mining Software Repositories Workshop (MSR 2005)*.
- Spacco, J., Winters, T. & Payne, T. (2006). Inferring Use Cases from Unit Testing. *AAAI Workshop on Educational Data Mining*.
- Spacco, J. W. (2006). *MARMOSET: A programming project assignment framework to improve the feedback cycle for students, faculty and researchers* (Tesis doctoral). University of Maryland, College Park. Ann Arbor, United States.
- SPhere-Online-Judge. (2022). Recuperado desde: <https://www.spoj.com/>.
- Sun, Q., Wu, J. & Liu, K. (2020). Toward Understanding Students' Learning Performance in an Object-Oriented Programming Course: The Perspective of Program Quality. *IEEE Access*, 8, 37505-37517. <https://doi.org/10.1109/ACCESS.2020.2973470>
- Sun, Q., Wu, J., Rong, W. & Liu, W. (2019). Formative assessment of programming language learning based on peer code review: Implementation and experience report. *Tsinghua Science and Technology*, 24(4), 423-434. <https://doi.org/10.26599/TST.2018.9010109>
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257-285.
- Tapia, J. A. (2001). Motivación y estrategias de aprendizaje: principios para su mejora en alumnos universitarios.
- Tarek, M., Ashraf, A., Heidar, M. & Eliwa, E. (2022). Review of Programming Assignments Automated Assessment Systems. *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 230-237. <https://doi.org/10.1109/MIUCC55081.2022.9781736>
- Tavares, P. C., Henriques, P. R. & Gomes, E. F. (2017). A computer platform to increase motivation in programming students-PEP. *CSEDU 2017 - Proceedings of the 9th International Conference on Computer Supported Education*, 1, 284-291.
- Thiébaud, D. (2015). Automatic Evaluation of Computer Programs Using Moodle's Virtual Programming Lab (VPL) Plug-In. *J. Comput. Sci. Coll.*, 30(6), 145-151.
- Ullah, Z., Lajis, A., Jamjoom, M., Altalhi, A., Al-Ghamdi, A. & Saleem, F. (2018). The effect of automatic assessment on novice programming: Strengths and limitations of existing systems. <https://doi.org/10.1002/cae.21974>
- Vennila, R., Labelle, D. & Wiendenbeck, S. (2004). Self-efficacy and mental models in learning to program. *ACM SIGCSE Bulletin*, 36(3).
- Verdú, E., Regueras, L. M., Verdú, M. J., Leal, J. P., De Castro, J. P. & Queirós, R. (2012). A distributed system for learning programming on-line. *Computers and Education*, 58(1), 1-10. <https://doi.org/10.1016/j.compedu.2011.08.015>

- Videla, R. L. (2010). Clases pasivas, clases activas y clases virtuales. ¿Transmitir o construir conocimientos? *Revista Argentina de Radiología*. <https://www.redalyc.org/articulo.oa?id=382538482010>
- Wardani, A. D., Gunawan, I., Kusumaningrum, D. E., Benty, D. D. N., Sumarsono, R. B., Nurabadi, A. & Handayani, L. (2020). Student Learning Motivation: A Conceptual Paper. *Proceedings of the 2nd Early Childhood and Primary Childhood Education (ECPE 2020)*, 275-278. <https://doi.org/https://doi.org/10.2991/assehr.k.201112.049>
- Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *International Biometric Society*. <http://www.jstor.org/stable/3001968>
- Wunsche, B. C., Huang, E., Shaw, L., Suselo, T., Leung, K. C., Dimalen, D., Van Der Mark, W., Luxton-Reilly, A. & Lobb, R. (2019). CodeRunnerGL - An interactive web-based tool for computer graphics teaching and assessment. *ICEIC 2019 - International Conference on Electronics, Information, and Communication*. <https://doi.org/10.23919/ELINFOCOM.2019.8706402>
- Yusof, N., Zin, N. A. M. & Adnan, N. S. (2012). Java Programming Assessment Tool for Assignment Module in Moodle E-learning System. *Procedia - Social and Behavioral Sciences*, 56, 767-773.
- Zimmerman, B. (1989). A social cognitive view of self-regulated academic learning. *Journal of Educational Psychology*, 81(3), 329-339. <https://doi.org/10.1037/0022-0663.81.3.329>