



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Analítica de aprendizaje en la asignatura Programación de Computadores: una investigación basada en métodos mixtos

Edna Johanna Chaparro Amaya

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

Analítica de aprendizaje en la asignatura Programación de Computadores: una investigación basada en métodos mixtos

Edna Johanna Chaparro Amaya

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

Director:
Felipe Restrepo Calle, 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

Dedicado a:

A mis padres.

A mi hermane Alelí.

A mi novio Camilo.

"No hay que temer nada en la vida, solo tratar de comprenderlo".

Marie Curie

Agradecimientos

En primer lugar quiero de manera especial agradecer al profesor Felipe Restrepo Calle por poner su confianza en mí para la realización de este proyecto, por haber sido paciente y guiarme a través de cada una de las etapas de este proyecto para culminar mi tesis.

También quiero agradecer al profesor Jhon Jairo Ramírez Echeverry quien con su experiencia y conocimientos fue de gran apoyo en el desarrollo de esta investigación, impulsándome a alcanzar los resultados que buscaba.

Agradezco a los profesores y estudiantes que integran el grupo de investigación PLaS (Programming Languages and Systems) del Departamento de Ingeniería de Sistemas e Industrial, por sus recomendaciones y contribuciones, las cuales me permitieron mejorar tanto académicamente como profesionalmente. En especial, quiero hacer mención a los estudiantes Nicolás Restrepo, Edder Hernández y Nicolai Romero, quienes me brindaron su tiempo y colaboración en la primera fase de esta investigación y quienes fueron parte fundamental del éxito de este trabajo.

Quiero agradecer a Sergio Sarmiento por su invaluable contribución en la segunda fase de esta investigación y quien con sus conocimientos y compromiso permitió el desarrollo satisfactorio de este estudio.

Agradezco a la Universidad Nacional de Colombia por permitirme acceder a todos los recursos y herramientas necesarios para realizar este proceso de investigación y culminar con éxito mis estudios de maestría.

Quiero agradecer a mi novio Camilo por siempre tener palabras de amor y motivación en los momentos de inseguridad y desaliento, quien realmente me dio las fuerzas para continuar hasta la finalización de este proyecto.

Quiero agradecer a mis amigos Joan y Paola, quienes con el ejemplo me han inspirado a seguir mis sueños, y quienes me dieron consejos y ánimos durante el desarrollo de este trabajo.

Agradezco a mi hermane Alelí Gael por ser la persona que me aportó significativamente con sus conocimientos en la etapa de escritura de este documento y quien con su ejemplo me ha enseñado a creer en mí y seguir adelante sin temor.

Finalmente, a mis padres por brindarme su apoyo durante todo este tiempo, sin ellos no hubiese podido culminar mis estudios y lograr mis metas.

Muchas gracias a todos.

Resumen

Análítica de aprendizaje en la asignatura Programación de Computadores: una investigación basada en métodos mixtos

El reciente crecimiento de nuevas formas de datos educativos, ha hecho que la analítica del aprendizaje surja como una solución para identificar información relevante en la toma de decisiones educativas. Un grupo de investigaciones en analítica del aprendizaje se concentra en identificar variables del proceso de aprendizaje relacionadas con el desempeño académico de los estudiantes. Sin embargo, pocas investigaciones consideran el uso de metodologías mixtas o cualitativas, lo que limita el entendimiento sobre los comportamientos de los alumnos. El objetivo general de este trabajo es determinar las relaciones existentes entre las medidas y métricas derivadas del proceso de aprendizaje y el rendimiento académico de los estudiantes en la asignatura Programación de Computadores de la Facultad de Ingeniería en la Universidad Nacional de Colombia durante el 2019 y el 2020. Este trabajo propone un diseño metodológico con enfoque mixto, no experimental, donde la fase cualitativa de la metodología está enfocada hacia el análisis de contenido.

Los resultados evidencian que existe una correlación positiva entre la cantidad de intentos de solución realizados por el alumno y su desempeño académico, lo que posiblemente se puede asociar a las percepciones de los estudiantes sobre la plataforma educativa utilizada en el curso como un ambiente que promueve la práctica constante debido a su disponibilidad en línea. Los errores/veredictos de las soluciones enviadas (respuesta correctas e incorrecta, límite de memoria excedido, errores de compilación y límite de tiempo excedido) también tienen correlaciones positivas, las cuales son corroboradas con las referencias de los estudiantes sobre retroalimentación formativa, consejos orientativos y casos de prueba. Las métricas de software como el conteo de tokens y las líneas de código de los programas diseñados tienen una correlación positiva significativa con la calificación final del alumno, lo cual se puede vincular con las referencias sobre ejercicios estimulantes y motivantes dentro de la plataforma educativa. Por otra parte, el índice de mantenibilidad tiene una correlación negativa, lo que se puede relacionar con las opiniones que resaltan la obtención de habilidades de programación. En contraste, se observan correlaciones negativas entre el uso de las herramientas de la plataforma educativa utilizada en el curso (p. ej. pruebas personalizadas, visualización de la ejecución del código y verificación de buenas prácticas de programación) con el rendimiento académico, las cuales son refutadas con las referencias de los estudiantes a estas herramientas como elementos positivos de la plataforma. En conclusión, se evidencia como el uso de métodos mixtos permite que los hallazgos de la fase cuantitativa sean corroborados, complementados o refutados por medio de las observaciones de los datos cualitativos.

Palabras clave: Analítica del Aprendizaje, Métodos Mixtos, Programación de Computadores, Análisis de correlaciones, Análisis de contenido.

Abstract

Learning analytics in Computer Programming: a mixed-methods investigation

The recent growth of new forms of educational data has led to the emergence of learning analytics as a solution to identify relevant information for educational decision making. A body of research in learning analytics focuses on identifying learning process variables related to students' academic performance. However, little research considers the use of mixed or qualitative methodologies, which limits the understanding of student behaviors. The general objective of this work is to determine the existing relationships between measures and metrics derived from the learning process and the academic performance of students in the Computer Programming courses of the Faculty of Engineering at the National University of Colombia during 2019 and 2020. This work proposes a methodological design with a mixed, non-experimental approach, where the qualitative phase of the methodology is focused on content analysis.

The results show that there is a positive correlation between the number of solution attempts made by the student and their academic performance, which can possibly be associated with the students' perceptions of the educational platform used in the course as an environment that promotes constant practice due to its online availability. Errors/verdicts of submitted solutions (correct and incorrect answer, memory limit exceeded, compilation errors, and time limit exceeded) also have positive correlations, which are corroborated with students' references to formative feedback, guiding hints, and test cases. Software metrics such as token count and lines of code of student-designed programs have a significant positive correlation with the student's final grade, which can be linked to references about stimulating and motivating exercises within the educational platform. On the other hand, the maintainability index has a negative correlation, which can be linked to opinions highlighting the attainment of programming skills. In contrast, negative correlations are observed between the use of the educational platform tools used in the course (e.g., custom input tests, visualization of code execution and verification of good programming practices) with academic performance, which are refuted by the students' references to these tools as positive elements of the platform. In conclusion, it is evident how the use of mixed methods allows the findings of the quantitative phase to be corroborated, complemented or refuted by the observations of the qualitative data.

Keywords: Learning Analytics, Mixed Methods, Computer Programming, Correlation analysis, Content analysis.

Esta tesis de maestría se sustentó el 15 de noviembre de 2022 a las 2:00 pm,
y fue evaluada por los siguientes jurados:

Fabio Augusto Gonzalez Osorio, Ph.D.
Universidad Nacional de Colombia.
Departamento de Ingeniería de Sistemas e Industrial

Sandra Liliana Rojas Martinez
Universidad Nacional de Colombia.
Departamento de Ingeniería de Sistemas e Industrial

Contenido

Resumen	vii
Abstract	viii
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 Alcance	7
1.6 Estructura del documento	8
2 Marco conceptual y trabajos relacionados	9
2.1 Analítica del aprendizaje (<i>Learning analytics</i>)	9
2.2 Analítica de aprendizaje en cursos de programación de computadores	12
2.3 Metodologías de analítica del aprendizaje	14
2.4 Trabajos relacionados	20
2.5 Desafíos de la analítica del aprendizaje	22
3 Diseño metodológico mixto para analítica del aprendizaje	24
3.1 Marco metodológico para analítica de aprendizaje	24
3.2 Fase 1: Datos cuantitativos	25
3.2.1 Recolección de datos cuantitativos	25
3.2.2 Consolidación del conjunto de datos cuantitativos	27
3.2.3 Limpieza del conjunto de datos cuantitativos	27
3.2.4 Identificación de variables de interés	28
3.2.5 Diseño de métricas de interés	28
3.2.6 Cálculo de métricas diseñadas	28

3.2.7	Análisis exploratorio de datos	29
3.2.8	Aplicación de técnicas de análisis y modelación de datos	29
3.3	Fase 2: Datos cualitativos	30
3.3.1	Recolección de datos cualitativos	30
3.3.2	Consolidación del conjunto de datos cualitativos	30
3.3.3	Limpieza del conjunto de datos cualitativos	30
3.3.4	Exploración de temas	31
3.3.5	Codificación abierta	32
3.3.6	Codificación axial	32
3.3.7	Relativización de los datos	33
3.4	Fase 3: Discusión resultados	34
3.4.1	Integración de fases	34
3.4.2	Interpretación de resultados	34
3.4.3	Generación de hipótesis	34
3.4.4	Conclusiones	35
4	Datos cuantitativos	36
4.1	Recolección de datos cuantitativos	36
4.2	Consolidación del conjunto de datos cuantitativos	41
4.3	Limpieza del conjunto de datos cuantitativos	43
4.4	Identificación de variables de interés	47
4.5	Diseño de métricas de interés	48
4.6	Cálculo de métricas diseñadas	50
4.7	Análisis exploratorio de datos	55
4.8	Aplicación de técnicas de análisis y modelación de datos	66
5	Datos cualitativos	73
5.1	Recolección de datos cualitativos	73
5.2	Consolidación del conjunto de datos cualitativos	74
5.3	Limpieza del conjunto de datos cualitativos	75
5.4	Exploración de temas	77
5.5	Codificación abierta	78
5.5.1	Pregunta proceso de aprendizaje	78
5.5.2	Pregunta calificación automática	91
5.5.3	Pregunta retroalimentación inmediata	97
5.6	Codificación axial	100
5.6.1	Pregunta proceso de aprendizaje	101
5.6.2	Pregunta calificación automática	105
5.6.3	Pregunta retroalimentación inmediata	107
5.7	Relativización de datos	111

6	Discusión de resultados	114
6.1	Integración de resultados cuantitativos y cualitativos	114
6.1.1	Resultados de veredictos de error	114
6.1.2	Resultados de intentos realizados	117
6.1.3	Resultados de uso de herramientas	119
6.1.4	Resultados de preguntas percepción	120
6.1.5	Resultados de métricas de software	121
6.2	Interpretación de resultados integrados	121
6.3	Generación de hipótesis	125
7	Conclusiones y trabajos futuros	128
7.1	Conclusiones	128
7.2	Trabajos futuros	132
7.3	Publicaciones	133
	Bibliografía	134

Lista de Figuras

- 1-1 Evolución y espectro de la analítica desde lo explícito hacia lo implícito. Adaptado de Cao (2017). 2
- 2-1 Tipos de datos recolectados en investigaciones de analítica de aprendizaje en cursos de programación. Adaptado de Carter *et al.* (2019). 13
- 3-1 Marco metodológico propuesto para analítica de aprendizaje mediante un enfoque de investigación de métodos mixtos. 26
- 4-1 Colecciones dentro de la base de datos MongoDB de UNCode 39
- 4-2 Tabla con las colecciones seleccionadas de la base de datos de UNCode 40
- 4-3 Cantidad total de actividades realizadas en UNCode en cada curso 44
- 4-4 Cantidad total de soluciones enviadas en cada actividad realizada en UNCode 45
- 4-5 Cantidad total de soluciones enviadas en las actividades seleccionadas 46
- 4-6 Cantidad total de envíos por estudiante 47
- 4-7 Distribución de intentos totales realizados (*Total_Submissions*) por estudiante en cada curso. 55
- 4-8 Tiempo promedio entre los intentos de cada estudiante *Duration_of_Submission*. 56
- 4-9 Tasas totales de veredictos obtenidos en todos los cursos. 57
- 4-10 Tasas totales de veredictos obtenidos en todos los cursos. 57
- 4-11 Tasas totales de uso de herramientas en todos los cursos. 58
- 4-12 Distribución del desempeño académico (*unicode_grade*) por grupo 60
- 4-13 1) Arriba: Porcentaje de estudiantes por sexo, 2) Abajo: Porcentaje de estudiantes por sexo discriminados por programa académico. 61
- 4-14 Porcentaje de estudiantes por carrera 62
- 4-15 1) Arriba: Histograma general de edades, 2) Abajo: Histogramas de edad discriminados por sexo 63
- 4-16 Distribución de edades por curso 64
- 4-17 Porcentaje de respuestas en escala de Likert de las siguientes preguntas: 1) Arriba: “UNCode fue útil en su proceso de aprendizaje de programación de computadores”, 2) Centro: “UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura” y 3) Abajo: “La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas”. 64

4-18	Porcentaje de respuestas en escala de Likert a la pregunta “Las siguientes actividades fueron útiles para su aprendizaje en esta asignatura”	66
4-19	Histograma de las calificaciones finales de los estudiantes de la asignatura	67
4-20	Variables con correlaciones significativas con respecto al desempeño académico	68
4-21	Variables con correlaciones significativas con respecto al desempeño académico de los estudiantes discriminadas por desempeño académico	71
5-1	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema bondades de la plataforma	102
5-2	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema herramientas de UNCode	103
5-3	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema logros pedagógicos	104
5-4	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema mejorable para aprender a programar	105
5-5	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema dificultades de implementación	106
5-6	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema logros pedagógicos	107
5-7	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema mejorable en calificación automática	108
5-8	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema mejorable	109
5-9	Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema resolución de problemas	110
5-10	Representación gráfica del modelo general producto del análisis de contenido	111

Lista de Tablas

2-1	Lenguajes de programación en investigaciones de analítica del aprendizaje	15
2-2	Cuadro comparativo de las metodologías de analítica de aprendizaje revisadas.	19
2-3	Variables correlacionadas con el desempeño académico encontradas en investigaciones de analítica del aprendizaje	21
4-1	Categorías de la escala Likert	37
4-2	Medidas de la interacción con UNCode consideradas en el conjunto de datos	49
4-3	Medidas identificadas de las respuestas de encuestas de percepción de la plataforma UNCode	50
4-4	Métricas construidas con base en las medidas de veredictos obtenidos	51
4-5	Métricas construidas con base en las medidas de uso de herramientas	52
4-6	Métricas de software a partir de los archivos de código fuente enviados por estudiantes	53
4-7	Estadísticas descriptivas de las métricas de software	59
5-1	Preguntas realizadas en las encuestas de percepción de los estudiantes con respecto a UNCode discriminadas por semestre	74
5-2	Códigos asignados a las preguntas consideradas en la fase de datos cualitativos	76
5-3	Cantidad de unidades de análisis agrupadas por códigos, categorías y temas	79
5-4	Ejemplos de unidades de análisis asignadas a la categoría casos de prueba	80
5-5	Ejemplos de unidades de análisis asignadas a la categoría consejos formativos	80
5-6	Ejemplos de unidades de análisis asignadas a la categoría conocer los errores	81
5-7	Ejemplos de unidades de análisis asignadas a la categoría disponibilidad en línea	81
5-8	Ejemplos de unidades de análisis asignadas a la categoría espacio de trabajo	81
5-9	Ejemplos de unidades de análisis asignadas a la categoría facilidad de uso	82
5-10	Ejemplos de unidades de análisis asignadas a facilidad para escribir y corregir código	82
5-11	Ejemplos de unidades de análisis asignadas a la categoría lenguajes de programación	83
5-12	Ejemplos de unidades de análisis asignadas a la categoría práctica constante	83
5-13	Ejemplos de unidades de análisis asignadas a la categoría Custom input	83
5-14	Ejemplos de unidades de análisis asignadas a la categoría Linter	84
5-15	Ejemplos de unidades de análisis asignadas a la categoría Python Tutor	84
5-16	Ejemplos de unidades de análisis asignadas a la categoría aprendizaje autónomo	84
5-17	Ejemplos de unidades de análisis asignadas a la categoría ejercicios estimulantes	85
5-18	Ejemplos de unidades de análisis asignadas al código calificación inmediata	85
5-19	Ejemplos de unidades de análisis asignadas al código calificación objetiva	86

5-20	Ejemplos de unidades de análisis asignadas al código facilita el planteamiento de problemas y ejercicios	86
5-21	Ejemplos de unidades de análisis asignadas a presentación de actividades académicas	86
5-22	Ejemplos de unidades de análisis asignadas a la categoría habilidades de programación	87
5-23	Ejemplos de unidades de análisis asignadas a la categoría resolución de problemas	87
5-24	Ejemplos de unidades de análisis asignadas a la categoría fallas generales UNCode	87
5-25	Ejemplos de unidades de análisis asignadas al código calificación ineficiente	88
5-26	Ejemplos de unidades de análisis asignadas al código casos de prueba	88
5-27	Ejemplos de unidades de análisis asignadas al código herramientas no disponibles	89
5-28	Ejemplos de unidades de análisis asignadas al código pérdida de información	89
5-29	Ejemplos de unidades de análisis asignadas al código registro	89
5-30	Ejemplos de unidades de análisis asignadas al código plataforma caída	89
5-31	Ejemplos de unidades de análisis asignadas al código replicar códigos	90
5-32	Ejemplos de unidades de análisis asignadas al código velocidad de procesamiento	90
5-33	Ejemplos de unidades de análisis asignadas al código visualización	90
5-34	Ejemplos de unidades de análisis asignadas a la categoría inflexibilidad de las comprobaciones	90
5-35	Ejemplos de unidades de análisis asignadas a la categoría falla en el objetivo educativo	91
5-36	Ejemplos de unidades de análisis asignadas a la categoría insuficiencia de la retroalimentación	91
5-37	Ejemplos de la unidad de análisis asignada a la categoría herramienta sustituible	91
5-38	Cantidad de unidades de análisis agrupadas por códigos, categorías y temas	92
5-39	Ejemplos de unidades de análisis asignadas a la categoría participación docente	93
5-40	Ejemplo de la unidad de análisis asignada a la categoría poco uso de la plataforma	93
5-41	Ejemplos de unidades de análisis asignadas a la categoría aprendizaje autónomo	94
5-42	Ejemplos de unidades de análisis asignadas al código evaluación de capacidades	94
5-43	Ejemplos de unidades de análisis asignadas al código retroalimentación formativa	94
5-44	Ejemplos de unidades de análisis asignadas al código seguimiento de calificaciones	94
5-45	Ejemplos de unidades de análisis asignadas a la categoría evaluación optimizada	95
5-46	Ejemplos de unidades de análisis asignadas al código calificación inmediata	95
5-47	Ejemplos de unidades de análisis asignadas al código calificación objetiva	95
5-48	Ejemplos de unidades de análisis asignadas al código retroalimentación inmediata	95
5-49	Ejemplos de unidades de análisis asignadas a la categoría fallas en validaciones	96
5-50	Ejemplos de unidades de análisis asignadas a inflexibilidad en las comprobaciones	96
5-51	Ejemplos de unidades de análisis asignadas a la categoría errores UNCode	96
5-52	Ejemplos de unidades de análisis asignadas a la categoría dificultades para entenderlo	96
5-53	Ejemplos de unidades de análisis asignadas a la categoría retroalimentación insuficiente	97
5-54	Cantidad de unidades de análisis agrupadas por códigos, categorías y temas	97
5-55	Ejemplos de unidades de análisis asignadas a la categoría casos de prueba ocultos	98
5-56	Ejemplos de unidades de análisis asignadas a la categoría orientación insuficiente	98

5-57	Ejemplos de unidades de análisis asignadas al código detalles menores	99
5-58	Ejemplos de unidades de análisis asignadas al código falta de claridad	99
5-59	Ejemplos de unidades de análisis asignadas al código identificación autónoma de errores	99
5-60	Ejemplos de unidades de análisis asignadas a la categoría comparación con salidas esperadas	99
5-61	Ejemplos de unidades de análisis asignadas a la categoría corregir errores	100
5-62	Ejemplos de unidades de análisis asignadas a la categoría especificaciones en la retroalimentación	100
5-63	Ejemplos de unidades de análisis asignadas al tema ambiente de la plataforma	101
5-64	Ejemplos de unidades de análisis asignadas al tema bondades de la plataforma	102
5-65	Ejemplos de unidades de análisis asignadas al tema Google Colaboratory	102
5-66	Ejemplos de unidades de análisis asignadas al tema herramientas de UNCode	103
5-67	Ejemplos de unidades de análisis asignadas al tema logros pedagógicos	104
5-68	Ejemplos de unidades de análisis asignadas al tema mejorable para aprender a programar	104
5-69	Ejemplos de unidades de análisis asignadas al tema dificultades de implementación .	105
5-70	Ejemplos de unidades de análisis asignadas al tema facilidad de uso	106
5-71	Ejemplos de unidades de análisis asignadas al tema logros pedagógicos	107
5-72	Ejemplos de unidades de análisis asignadas al tema mejorable en calificación automática	108
5-73	Ejemplos de unidades de análisis asignadas al tema condiciones iniciales claras	108
5-74	Ejemplos de unidades de análisis asignadas al tema mejorable	109
5-75	Ejemplos de unidades de análisis asignadas al tema resolución de problemas	110
5-76	Ejemplos de unidades de análisis asignadas al tema señalar errores	111

1 Introducción

1.1. Motivación

En la última década, con el auge de la era digital se ha visto un incremento de la tecnología en todas las áreas de la sociedad. Entre estas áreas se encuentran los ambientes educativos, donde se hace uso de herramientas como computadores, tableros electrónicos, ambientes virtuales y sistemas de gestión del aprendizaje (Long y Siemens, 2011; Siemens, 2013). En consecuencia, ha crecido exponencialmente la cantidad de datos producidos durante los procesos de aprendizaje, junto con su potencial de generar conocimiento sobre los factores que determinan el éxito académico (Baker e Inventado, 2014; Charlton *et al.*, 2013; Kumar *et al.*, 2015; Long y Siemens, 2011; Seufert y Meier, 2018; Siemens, 2013). En este sentido, la información generada puede ser empleada como guía para los profesores, instituciones y estudiantes hacia la toma de decisiones relacionadas con los procesos de aprendizaje, enseñanza y administración educativa (Ifenthaler, 2017; Kumar *et al.*, 2015; Long y Siemens, 2011). En específico, le sirve a los docentes para identificar dificultades en el aprendizaje, generar ayudas para los estudiantes y mejorar el diseño curricular de las asignaturas (Charlton *et al.*, 2013). Los datos educativos también sirven para que los alumnos entiendan sus procesos de aprendizaje, identifiquen errores y descubran nuevas técnicas basadas en la auto-reflexión (Seufert y Meier, 2018). Sumado a esto, los docentes y directivos académicos deben monitorear varios estudiantes simultáneamente, mientras generan reportes sobre sus procesos de aprendizaje. En este sentido, se genera un incremento de datos educativos no estructurados o semi-estructurados, originando el desafío de identificar la información relevante para la construcción de indicadores informativos (Charlton *et al.*, 2013; Long y Siemens, 2011).

El análisis de datos específicamente en el área educativa se conoce como analítica del aprendizaje (*learning analytics*, en inglés), y es considerada como el factor más importante en el futuro de la educación, especialmente, en la educación superior (Long y Siemens, 2011). En investigaciones educativas tradicionales los conceptos de seguimiento de estudiantes, análisis de datos educativos y diseño de evaluaciones del aprendizaje para mejorar la enseñanza no son ideas nuevas (Lockyer y Dawson, 2011; Pistilli *et al.*, 2014). La analítica del aprendizaje se basa en los principios de las investigaciones tradicionales, pero se diferencia en que aprovecha nuevas oportunidades como nuevas formas de datos digitales y técnicas de análisis computacionales avanzadas de la ciencia de datos y la inteligencia artificial (Pistilli *et al.*, 2014). De acuerdo con la 1^{ra} Conferencia Internacional de Learning Analytics y Conocimiento, la analítica del aprendizaje se puede definir como la “recolección y análisis de datos generados de los alumno/as y de sus entornos digitales de aprendizaje, con el propósito de entender y mejorar

los procesos de enseñanza y aprendizaje” (Gasevic et al., 2014; Long y Siemens, 2011; Siemens, 2013). En otras palabras, es la convergencia entre el aprendizaje (investigación educativa, ciencias del aprendizaje, tecnología educativa, etc.), la analítica (estadística, visualización, ciencia de datos, modelación computacional, inteligencia artificial, etc.) y el diseño enfocado en personas (Long y Siemens, 2011).

La analítica del aprendizaje también se puede entender desde la perspectiva de la ciencia de datos. Esta perspectiva inicia desde los análisis explícitos por medio de reportes superficiales, análisis estadísticos y valores sin procesar; y termina con análisis implícitos como modelos predictivos, conocimiento aplicable y toma de decisiones (Cao, 2017). La Figura 1-1 detalla este enfoque, donde se hace una división en dos etapas, acorde con el grado de visibilidad de los datos y la complejidad del análisis.

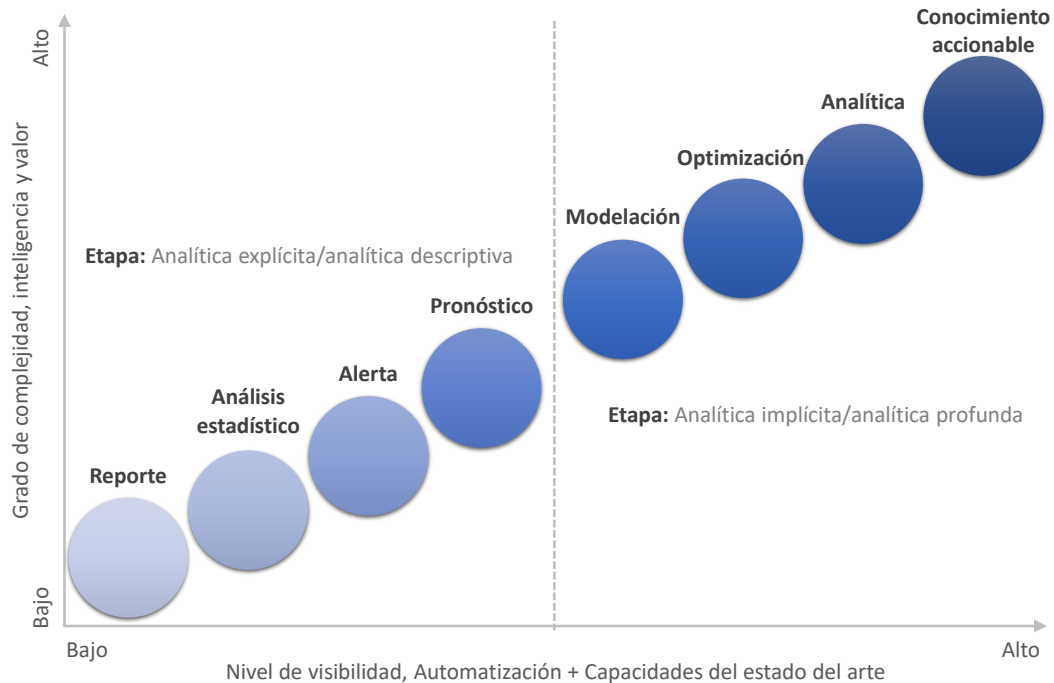


Figura 1-1: Evolución y espectro de la analítica desde los explícito hacia lo implícito. Adaptado de Cao (2017).

- **Analítica explícita:** Esta etapa tiene como objetivo identificar y describir la distribución, tendencias y dinámicas de los datos disponibles. En este sentido, se requiere un grado alto de visibilidad de la información y los problemas que se resuelven son de baja complejidad.
- **Analítica implícita:** Esta etapa se enfoca en entender las causas de los comportamientos de los datos que ocurren u ocurrirán en el futuro. Superando así, las limitaciones que tiene la analítica explícita y avanzando hacia un análisis profundo. Adicionalmente, se puede determinar

los mejores o peores casos posibles y diseñar estrategias óptimas de intervención; a partir de información con un nivel de visibilidad mínimo y utilizando métodos de mayor complejidad.

Aplicando este enfoque, algunos autores han propuesto diferentes modelos metodológicos en los que consideran cíclico el proceso de aplicación de la analítica del aprendizaje. Un ejemplo, es el marco metodológico propuesto por Clow (2012), que inicia con la generación de datos que representan las interacciones y comportamientos de los estudiantes. Con base en estos datos, se toman decisiones que son aplicadas por medio de intervenciones educativas con el objetivo de mejorar el aprendizaje. El ciclo se cierra cuando las intervenciones construidas son aplicadas en el mismo grupo de estudiantes o en uno nuevo. Otro marco es el de Carter *et al.* (2019), el cual divide el proceso de analítica de aprendizaje en cinco actividades principales: en primer lugar, la operacionalización de comportamientos observables, es decir, identificar los datos que representan procesos de aprendizaje; seguido de esto está la recolección de datos, que posteriormente son analizados; con base en los resultados del análisis se diseñan intervenciones; y finalmente, son aplicadas en el mismo contexto educativo o en uno diferente. Por otra parte, el modelo descrito por Siemens (2013) incluye los siguientes siete componentes: recolección, almacenamiento, limpieza, integración, análisis, representación y visualización, y finalmente, acción. La ventaja de este tipo de marcos metodológicos es que aplican un enfoque *top-down*, lo que garantiza que las intervenciones diseñadas puedan ser aprovechadas de manera sistematizada y no estén ajustadas para un solo contexto educativo específico (Pistilli *et al.*, 2014; Siemens, 2013).

En conclusión, la analítica del aprendizaje le entrega los investigadores en educación nuevas herramientas para estudiar los procesos de enseñanza y aprendizaje. Uno de los principales beneficios de la analítica del aprendizaje es darle estructura a datos educativos no estructurados (Dietze *et al.*, 2016), por medio del manejo y procesamiento realizados durante la recolección, análisis y visualización de la información (Siemens, 2013; Zikopoulos *et al.*, 2011). Otro beneficio es que los descubrimientos obtenidos ayudan a tomar decisiones educativas informadas con impactos positivos como retroalimentación más rápida, precisa y aplicable para los estudiantes (Ifenthaler, 2017; Schmitz *et al.*, 2017) o ayudar a promover el aprendizaje y enseñanza de calidad, por medio de innovaciones pedagógicas con evidencia empírica en el éxito (Lockyer y Dawson, 2011). Finalmente, los educadores, diseñadores curriculares y directivos institucionales ganan nuevos conocimientos importantes, cuando el proceso de aprendizaje es claro y visible (Long y Siemens, 2011).

1.2. Trabajos relacionados

La revisión de literatura realizada de analítica del aprendizaje evidencia que estas investigaciones se enfocan en distintas variables educativas. En primer lugar, están el desempeño académico y la retención de estudiantes en el curso, por medio de sistemas de retroalimentación personalizada (Arnold y Pistilli, 2012) y modelos de predicción de estudiantes en riesgo de desistir del curso (Monllaó Olivé *et al.*, 2020). La siguiente variable es la motivación, evidenciando cómo el uso de analítica de datos

puede impactar en la orientación motivacional de los estudiantes (Lonn *et al.*, 2015). Otra variable es la dedicación, visualizando los patrones de compromiso de los estudiantes e identificando los tipos de alumnos presentes en el conjunto de datos (Coffrin *et al.*, 2014). Por otra parte, está la ganancia de aprendizaje adquirida por un estudiante, la cual es cuantificada por medio de análisis de los mensajes publicados en foros de discusión (Wu y Wu, 2018). La satisfacción también es objeto de investigación, en trabajos donde se realiza el análisis textual de sentimientos presentes en los mensajes de los foros educativos (Elia *et al.*, 2019). Por último, varias investigaciones se enfocan en las habilidades metacognitivas, por ejemplo, cuantificar la capacidad de aprendizaje autorregulado de cada estudiante con base en registros individuales de logros académicos, interacciones con el contenido del curso e información personal (Kizilcec *et al.*, 2017).

Los resultados de la analítica del aprendizaje generan muchos beneficios educativos, por esta razón ha sido aplicada en diferentes áreas de conocimiento. Algunos autores se han concentrado en emplear analítica del aprendizaje en el contexto de cursos de programación de computadores (Azcona *et al.*, 2019; Blikstein, 2011; Scheffel *et al.*, 2012). Los tipos de datos que son utilizados en este tipo de investigaciones se dividen en dos: estándar y amplificados (Carter *et al.*, 2019). La primera categoría corresponde al conjunto de interacciones registradas en el sistema de gestión del aprendizaje (LMS, por sus siglas en inglés), ambiente virtual de aprendizaje (VLE, por sus siglas en inglés) o ambiente de desarrollo integrado (IDE, por sus siglas en inglés) durante la realización de actividades de programación. En esta categoría se encuentran trabajos donde se examinan los datos de edición (Blikstein *et al.*, 2014), compilación (Blikstein, 2011) y depuración (Scheffel *et al.*, 2012) del código fuente de los programas desarrollados por los estudiantes. La segunda categoría son los datos amplificados, que se define como la información que solo puede ser recolectada cuando el IDE utilizado en el curso es complementado con módulos adicionales (plugins). Dentro de esta categoría encontramos los datos sociales (Hu y Huang, 2018), las encuestas (Klašnja-Milićević *et al.*, 2011) y los biométricos (Trætterberg *et al.*, 2017).

Las investigaciones de analítica de aprendizaje en cursos de programación aplican diferentes intervenciones para cumplir sus objetivos. Una de las más utilizada es la modelación y predicción de los comportamientos de los estudiantes (Azcona *et al.*, 2019; Berland *et al.*, 2015; Lagus *et al.*, 2018). El segundo tipo de intervención es generar un seguimiento efectivo del progreso del curso, por medio de visualizaciones de estadísticas y métricas que representan los comportamientos de los alumnos (Hsiao y Lin, 2017; Leony *et al.*, 2013; Shen *et al.*, 2020). Por último, está la retroalimentación personalizada, con la construcción de guías adaptativas de aprendizaje (Lu *et al.*, 2017) y sistemas de recomendación automática de contenidos educativos adaptable a cada estudiante (Klašnja-Milićević *et al.*, 2011).

Un grupo de investigaciones en analítica del aprendizaje tienen como objetivo identificar las variables del proceso de aprendizaje que se encuentran relacionadas con desempeño académico de los estudiantes. Un ejemplo es el trabajo realizado por Zacharis (2015) que desarrolla un modelo para

predecir los estudiantes en riesgo de obtener bajos desempeños, a partir de los datos sobre el uso de la plataforma Moodle utilizada en la clase. Otras investigaciones, como la de Azcona *et al.* (2019), presentan una metodología para detectar estudiantes en riesgo de reprobación de las actividades en un curso de programación y apoyar la retroalimentación adaptativa. Este trabajo construye un modelo predictivo usando información estática (e.g. datos demográficos) y datos dinámicos (e.g. registros de actividades y comportamiento de programación). Sumado a esto, autores como Andergassen *et al.* (2014) investigan las relaciones potenciales entre los resultados de aprendizaje y el uso del LMS del curso por parte de los alumnos durante la preparación de exámenes, enfocado en práctica y repetición. En este grupo de investigaciones también se encuentra el modelo predictivo de Barber y Sharkey (2012), el cual emplea información demográfica con el objetivo de calcular la probabilidad de que un estudiante repruebe el curso. Adicionalmente, está la investigación de Macfadyen y Dawson (2010), donde se analizan los datos de seguimiento de uso de un LMS usado en un curso que utiliza Blackboard-Vista. Por último, está el trabajo desarrollado por Vahdat *et al.* (2015) que tiene como objetivo entender el comportamiento de estudiantes de ingeniería de sistemas y computación en un curso donde se usa un simulador de circuitos.

La analítica del aprendizaje en un área de conocimiento relativamente nueva, por eso aun existen varios retos en esta disciplina. Los desafíos en la analítica del aprendizaje se clasifican en dos categorías: sociales y tecnológicos (Ang *et al.*, 2020). En la categoría de problemáticas sociales, el manejo de información personal educativa tiene implicaciones de privacidad (Ang *et al.*, 2020; Dascalu *et al.*, 2018; Pardo y Siemens, 2014). Adicionalmente, se encuentran los desafíos éticos que pueden estar relacionados con: el uso de conjuntos de datos sesgados (Dascalu *et al.*, 2018; Selwyn, 2020), la predicción incorrecta del comportamiento de los alumnos que pueden generar ideas preconcebidas negativas (Ang *et al.*, 2020; Dascalu *et al.*, 2018), la comparación de un grupo experimental con uno de control originando desventajas educativas entre estudiantes (Drachsler y Greller, 2016), y la falta de inclusión de alumnos no tradicionales (e.g. trabajadores de medio o tiempo completo) (Ferguson, 2012; Schmitz *et al.*, 2017). La otra categoría de desafíos son las problemáticas tecnológicas. En este grupo encontramos la falta de políticas inter-institucionales para compartir información e integrar los sistemas educativos construidos (Ang *et al.*, 2020; Dascalu *et al.*, 2018; Schmitz *et al.*, 2017). Adicionalmente, están la dificultad de escalabilidad de intervenciones generadas de investigaciones de analítica de aprendizaje (Najafabadi *et al.*, 2016; Wong y Li, 2020). Finalmente, están la concentración del uso de métodos cuantitativos en comparación con los métodos cualitativos o mixtos en este tipo de investigaciones (Mangaroska y Giannakos, 2017; Tempelaar *et al.*, 2016).

1.3. Problema de investigación

Autores como Mangaroska y Giannakos (2017) basados en revisiones de literatura, llegan a la conclusión de que en investigaciones de analítica del aprendizaje la metodología cuantitativa aun precede los métodos mixtos y la metodología cualitativa. Esta problemática se debe a la abundancia, mayor

disponibilidad y facilidad de recolección de datos cuantitativos en ambientes virtuales y sistemas de gestión del aprendizaje en comparación con los datos de tipo cualitativo (Mangaroska y Giannakos, 2017).

Teniendo en cuenta que en los últimos años los contextos educativos se han transformado hacia clases semi-presenciales o completamente virtuales, las aplicaciones de analítica del aprendizaje basadas exclusivamente en metodologías cuantitativas tienen dificultades en entender los procesos de aprendizaje por completo (Kop *et al.*, 2017; Rienties y Toetenel, 2016). Un ejemplo son los resultados de Macfadyen y Dawson (2010), confirmados por Tempelaar *et al.* (2016), los cuales evidencian que utilizar el seguimiento del tecleado o datos cuantitativos como única fuente de información es una aproximación muy limitada para comprender los procesos de aprendizaje de los estudiantes. Esta limitación en las metodologías de analítica de aprendizaje genera necesidad de información educativa de alta calidad, la cual permita tomar decisiones sobre la generación de intervenciones y su correcta implementación (Hilliger *et al.*, 2020).

Por otra parte, autores como Schmitz *et al.* (2017) identifican como el mayor desafío de la analítica del aprendizaje la habilidad de actuar con base en la información obtenida, es decir, la capacidad no solo de interpretar los descubrimientos obtenidos, sino también de cambiar el comportamiento de los profesores y estudiantes en la práctica. En este sentido, realizar investigaciones educativas que integren tanto datos cuantitativos como cualitativos permite obtener un entendimiento mucho más profundo, inclusivo, y culturalmente consciente del proceso de desarrollo de conocimientos y aprendizaje de la comunidad estudiantil (Gergen *et al.*, 2015; Kop *et al.*, 2017). Teniendo en cuenta esto, esta tesis propone dar respuesta a la siguiente pregunta de investigación: ¿Cómo los métodos mixtos de investigación aplicados en la analítica del aprendizaje, pueden generar un mejor entendimiento de las relaciones existentes entre las variables generadas a lo largo del proceso de aprendizaje y el rendimiento académico de los estudiantes en la asignatura Programación de Computadores?.

1.4. Objetivos

1.4.1. Objetivo general

Determinar las relaciones existentes entre las métricas calculadas del proceso de aprendizaje y el rendimiento académico de los estudiantes en la asignatura Programación de Computadores mediante una estrategia de analítica de aprendizaje fundamentada en métodos mixtos de investigación.

1.4.2. Objetivos específicos

- Establecer una estrategia de analítica de aprendizaje fundamentada en métodos mixtos de investigación para el estudio de los datos de la asignatura Programación de Computadores.

- Preparar el conjunto de datos histórico y las métricas relacionadas con el proceso de aprendizaje de los estudiantes de la asignatura Programación de Computadores para llevar a cabo el proceso de análisis de datos.
- Analizar los datos cuantitativos y cualitativos disponibles de la asignatura Programación de Computadores mediante la estrategia propuesta.
- Identificar las relaciones existentes entre las variables generadas a lo largo del proceso de aprendizaje y el rendimiento académico de los estudiantes de Programación de Computadores por medio del análisis de datos cuantitativos y cualitativos.

1.5. Alcance

La Sociedad de Investigación en Analítica del Aprendizaje (SoLAR) identifica los siguientes cuatro enfoques para este tipo de investigaciones (Society for Learning Analytics Research, 2022):

- **Analítica descriptiva:** Se limita al análisis de datos históricos, entendiendo las distribuciones y tendencias por medio de técnicas estadísticas.
- **Analítica diagnóstica:** Tiene como objetivo encontrar las razones por las cuales se dan los comportamientos de los datos, a través de técnicas como minería de datos y análisis de correlaciones.
- **Analítica predictiva:** Identifica patrones en los datos históricos, aplicando modelos estadísticos y algoritmos para entender relaciones entre variables y predecir tendencias futuras.
- **Analítica prescriptiva:** Se enfoca en generar una o más recomendaciones basadas en las predicciones del comportamiento de los datos. Este tipo de investigaciones utiliza una combinación de aprendizaje de máquina, algoritmos avanzados y modelación computacional.

Este trabajo se encuentra en el marco de la analítica descriptiva y diagnóstica, realizando un análisis univariado tanto de datos cuantitativos como cualitativos y un análisis multivariado para identificar relaciones existentes entre las variables consideradas. Autores como Siemens (2013) definen el proceso de la analítica del aprendizaje como proceso cíclico cerrado. Acorde con este autor, este tipo de investigaciones inicia con la recolección y adquisición de datos, la cual esta determinada por el objetivo del trabajo (aprendizaje, administración, investigación institucional, etc.). Posteriormente, los datos recolectados son almacenados, y se debe realizar un proceso de limpieza y filtrado. El siguiente paso consiste en la integración de conjuntos de datos, obteniendo un consolidado homogéneo de datos. Seguido de esto, se realiza el proceso de análisis utilizando herramientas y técnicas como procesamiento de lenguaje natural, análisis estadísticos, aprendizaje de máquina, etc. Los resultados del análisis generan representaciones y visualizaciones para facilitar la interpretación de la información. Y finalmente, se toman decisiones para realizar acciones educativas aplicables como intervenciones,

optimizaciones o sistemas de alerta temprana. La aplicación de estas acciones genera nuevamente datos relacionados con el proceso de aprendizaje, estos pueden ser recolectados para iniciar un nuevo proceso de analítica del aprendizaje. Los resultados finales de esta investigación se encuentran en la fase de representación y visualización. Adicionalmente, se hace la interpretación de los comportamientos encontrados en el conjunto de datos. Los productos finales de esta investigación pueden ser la base de trabajos futuros relacionados como la creación y aplicación de intervenciones educativas, logrando cerrar el ciclo de la analítica del aprendizaje.

1.6. Estructura del documento

El resto de este documento está estructurado de la siguiente manera: El Capítulo 2 corresponde a la descripción del marco conceptual y los trabajos relacionados de analítica del aprendizaje, tanto de manera general como en el contexto de cursos de programación de computadores. El Capítulo 3 explica el marco metodológico de analítica del aprendizaje fundamentada en métodos mixtos utilizado para el desarrollo de esta investigación. En el Capítulo 4 y el Capítulo 5 se describe la preparación, transformación y análisis de los datos cuantitativos y cualitativos, respectivamente. El Capítulo 6 expone la discusión de resultados, integrando el análisis cuantitativo y cualitativo. Finalmente, el Capítulo 7 presenta las conclusiones y los trabajos futuros derivados de esta investigación.

2 Marco conceptual y trabajos relacionados

En este capítulo se establece un contexto sobre las investigaciones de analítica del aprendizaje realizadas en diversos ambientes educativos, el cual es construido por medio de una revisión de literatura. En primer lugar, la Sección 2.1 especifica los objetivos, fuentes de información, variables de enfoque y áreas de aplicación identificadas en los trabajos de analítica del aprendizaje. En segunda instancia, la Sección 2.2 detalla los tipos de datos, intervenciones y lenguajes de las aplicaciones de analítica del aprendizaje en cursos de programación de computadores. En tercer lugar, dentro de la Sección 2.3 se describen y comparan diferentes metodologías empleadas en este tipo de investigaciones. Posteriormente, la Sección 2.4 detalla los hallazgos obtenidos en investigaciones enfocadas en relacionar variables del proceso de aprendizaje con el desempeño académico de los estudiantes. Finalmente, en la Sección 2.5 de este capítulo se exponen los retos y desafíos identificados por algunos investigadores en los trabajos realizados de analítica del aprendizaje.

2.1. Analítica del aprendizaje (*Learning analytics*)

La definición de analítica del aprendizaje fue por primera vez acuñada durante la 1^{ra} Conferencia Internacional de Learning Analytics y Conocimiento, definiéndola explícitamente como como la “recolección y análisis de datos generados de los alumno/as y de sus entornos digitales de aprendizaje, con el propósito de entender y mejorar los procesos de enseñanza y aprendizaje” (Gasevic *et al.*, 2014; Long y Siemens, 2011; Siemens, 2013). A partir de la revisión de literatura realizada acerca de investigaciones enfocadas en analítica del aprendizaje se evidenció que los objetivos de estos proyectos se pueden dividir en seis categorías. En primer lugar, está la predicción del éxito académico de los estudiantes, el cual ha sido históricamente uno de los usos más comunes (Society for Learning Analytics Research, 2022). En este grupo se encuentra el trabajo de Azcona *et al.* (2019), donde se presenta un modelo que aprovecha datos educativos históricos para detectar automáticamente los alumnos en riesgo de no aprobar las evaluaciones del curso. El segundo objetivo identificado es apoyar el desarrollo de habilidades y estrategias de aprendizaje a largo plazo. Trabajos como el de Tabuenca *et al.* (2015) cumplen con este objetivo, por medio de una herramienta móvil educativa para cursos virtuales, la cual evidencia que la generación de notificaciones a partir de datos de tiempo de uso pueden influir en el desarrollo de habilidades de manejo del tiempo y autorregulación. En tercer lugar, la analítica de aprendizaje se utiliza para el desarrollo de habilidades prácticas como trabajo colaborativo, pensamiento crítico, co-

municación y creatividad. En esta categoría se encuentra la herramienta AMOEBA, la cual ofrece un análisis en tiempo real del progreso y estilo de aprendizaje de los estudiantes, lo que facilita a los docentes la decisión de emparejar estudiantes en grupos de trabajo (Berland *et al.*, 2015). Otro de los usos de la analítica del aprendizaje es generar un proceso de aprendizaje consciente, por medio de la auto-reflexión de los estudiantes. Un ejemplo es el trabajo de Gašević *et al.* (2014), donde se realiza un análisis textual del uso de una herramienta de anotaciones de vídeo utilizada en un contexto educativo; los resultados obtenidos pueden ayudar a los estudiantes a ser conscientes de su propio lenguaje reflexivo y desarrollar habilidades meta-cognitivas. Por otra parte, la analítica del aprendizaje tiene la capacidad de promover la enseñanza de alta calidad, por medio de evidencia empírica en el éxito de innovaciones pedagógicas. Este objetivo es planteado en el trabajo de Martin y Whitmer (2016), donde se estudia el efecto que tiene la planeación de la disponibilidad de los contenidos educativos en el comportamiento y desempeño del estudiante. El último de los objetivos identificados es la generación de retroalimentación automática y personalizada, como en el caso Azcona *et al.* (2019), donde un grupo de estudiantes recibe notificaciones personalizadas sugiriendo materiales del curso, que están basadas en las predicciones de un modelo diseñado para clasificar el desempeño académico.

Otra clasificación que puede ser evidenciada, con base en las investigaciones de analítica del aprendizaje revisadas, es con respecto a las fuentes de información que son utilizadas para analizar los comportamientos de los estudiantes. La mayoría de los estudios utilizan datos sobre el uso de LMS o VLE. Un ejemplo es el trabajo hecho por Martin y Whitmer (2016), donde se intenta evidenciar el efecto del aplicar aprendizaje adaptable, por medio del análisis de datos de uso de la plataforma Blackboard incluyendo registro de entrada, tiempo total de uso, tiempo promedio por sesión, contenido de los módulos accedidos, y tiempo entre apertura y acceso a un módulo. Otro ejemplo es la investigación hecha por Ruipérez-Valiente *et al.* (2015), la cual ayuda a los profesores a tomar decisiones por medio de la visualización de 21 indicadores basados en interacciones de los estudiantes con la plataforma Khan Academy. Entre los indicadores considerados están: uso total de la plataforma, nivel de progreso, distribución del tiempo de uso, hábitos de resolución de ejercicios, hábitos de gamificación y estado emocional. Otra fuente de datos utilizada son los foros de discusión educativos, como en el caso del modelo creado por Wu y Wu (2018) que calcula la ganancia de aprendizaje de cada estudiante, por medio de un análisis de redes sociales basado en datos de un foro educativo. Las encuestas son otro instrumento usado en la recolección de datos en investigaciones de analítica del aprendizaje. En este grupo se encuentra el trabajo de Robinson *et al.* (2016), donde se aplican técnicas de procesamiento de lenguaje natural para el análisis de respuestas a preguntas abiertas sobre motivación e intenciones de los alumnos, y usar estos resultados como indicadores para predecir la persistencia y probabilidad de completar el curso del estudiante. Por otra parte están los datos biométricos, es decir, los que se basan en respuestas físicas de los alumnos. El trabajo de Tam *et al.* (2016) hace uso de esta fuente de información con el sistema PETA, el cual aplica algoritmos de análisis facial para el monitoreo de tiempos de atención, orientación facial y movimientos oculares, notificando a los estudiantes sobre sus comportamientos. Adicionalmente, es común el uso de más de una fuente de información en este tipo de investigaciones. Un ejemplo es el trabajo realizado por Kurilovas (2019), donde se re-

lacionan estilos de aprendizaje identificados por medio de cuestionarios, con el uso de herramientas y/o actividades del VLE usado en el curso. En esta categoría también está el modelo propuesto por Azcona *et al.* (2019), donde se detecta automáticamente a los estudiantes en riesgo académico con base en información recolectada tanto fuera del desarrollo del curso (datos demográficos), como datos generados durante la clase, tales como: registros de ingreso, interacción con el material didáctico y entrega de tareas.

La analítica del aprendizaje dependiendo del objetivo de la investigación puede tener diferentes variables de enfoque. En primer lugar, están los trabajos que se concentran en aumentar el desempeño académico y la retención de estudiantes en el curso. Arnold y Pistilli (2012) son unos de los investigadores que emplean este enfoque, desarrollando el sistema *Course Signals* (CS) que ofrece retroalimentación personalizada en tiempo real a los estudiantes sobre su progreso en la asignatura, teniendo como meta aumentar el desempeño del alumno. Este enfoque también está presente la investigación realizada por Monllaó Olivé *et al.* (2020), donde se propone un modelo predictivo capaz de identificar estudiantes en riesgo de desistir de un curso masivo en línea (MOOC, por sus siglas en inglés), el cual es entrenado con datos de actividades donde los estudiantes adquieren conocimientos resolviendo ejercicios. Otro grupo de trabajos se enfoca en la variable de motivación, como en el caso de Lonn *et al.* (2015) que estudian cómo el uso de analítica de datos puede impactar en la orientación motivacional de los estudiantes. Otra variable considerada es la dedicación, estudiada en trabajos como el de Coffrin *et al.* (2014) que logran clasificar significativamente tipos de estudiantes y visualizar los patrones de dedicación de los alumnos de un MOOC.

Por otra parte, autores como Wu y Wu (2018) cuantifican la ganancia de aprendizaje obtenida por los estudiantes, identificando tipos de comportamientos cognitivos en los conocimientos compartidos entre estudiantes dentro de un foro educativo. La satisfacción también es objeto de investigación en aplicaciones de analítica de aprendizaje, como en trabajos como el de Elia *et al.* (2019) donde se presenta un software que evalúa en tiempo real la satisfacción de un estudiante, por medio de análisis textual de sentimientos presentes en los mensajes de los foros de discusión. Por último, un grupo de variables que son el foco de muchas investigaciones son las habilidades meta-cognitivas, entendidas como capacidades de reflexión, planeación y control del proceso de aprendizaje propio. Por ejemplo, en Kizilcec *et al.* (2017) se propone una metodología para cuantificar las habilidades de aprendizaje autorregulado de cada estudiante con base en registros individuales de logros académicos, interacciones con el contenido del curso e información personal (datos demográficos, dedicación de tiempo, metas personales, historial académico y experiencia previa).

Los resultados de las investigaciones educativas de analítica del aprendizaje generan oportunidades para mejorar los procesos de aprendizaje, por esta razón han sido aplicados en asignaturas de diversas áreas de conocimiento. Se encontraron estudios de analítica de aprendizaje en cursos de ciencias exactas como física, química y matemáticas (Ruipérez-Valiente *et al.*, 2015), cursos avanzados como macroeconomía y optimización discreta (Coffrin *et al.*, 2014) o cursos de artes escénicas (Gašević *et*

al., 2014). Por otra parte, algunas investigaciones consideran datos de diferentes contextos educativos; como en el caso de Tabuenca *et al.* (2015), que estudia un curso de psicología junto con uno de sistemas de información geográfica. Finalmente, una de las áreas de conocimiento donde se han realizado varios trabajos de analítica del aprendizaje son las ciencias de la computación, como en cursos de introducción a la programación de computadores (Azcona *et al.*, 2019).

2.2. Analítica de aprendizaje en cursos de programación de computadores

Algunos autores se han concentrado en emplear analítica del aprendizaje en el contexto de cursos de programación de computadores. La Figura 2-1 muestra los tipos de datos que son utilizados en este tipo de investigaciones habitualmente, los cuales están basados en la taxonomía propuesta por Carter *et al.* (2019). En primera instancia, están los datos estándar que corresponden al conjunto de interacciones registradas en un LMS, VLE o IDE durante la realización de actividades de programación. En esta categoría se encuentran por ejemplo los datos de edición de código fuente; usados por ejemplo en la investigación de Blikstein *et al.* (2014), donde se emplean métodos de aprendizaje de máquina para encontrar patrones en el comportamiento de programación de los estudiantes. En este trabajo la medición se hace comparando dos capturas consecutivas del código fuente y determinando el número de caracteres o líneas adicionadas, eliminadas o modificadas. Otra investigación realizada por Blikstein (2011) aborda la subcategoría correspondiente a los datos de compilación, donde se calculan métricas que incluyen frecuencia y cantidad de compilaciones correctas e incorrectas, y se emplean técnicas cuantitativas para la categorización de los comportamientos en términos de habilidades en programación. La última subcategoría de los datos estándar son los relacionados con hábitos de depuración; un ejemplo es la investigación de Scheffel *et al.* (2012), donde se identifican cuáles son las acciones más relevantes en el proceso de aprendizaje de los estudiantes de programación de computadores. El objetivo de ese trabajo es entender las dinámicas o dificultades presentes en el curso; entre las actividades monitoreadas está la apertura del depurador *gdb*, el archivo compilado y las advertencias o mensajes de error generadas.

Por otra parte, algunas investigaciones de analítica del aprendizaje en cursos de programación utilizan datos amplificados (Figura 2-1), que corresponden a la información que solo puede ser recolectada cuando el IDE utilizado en el curso es complementado con módulos adicionales (*plugins*). Dentro de esta categoría encontramos los datos sociales; como en el caso de Hu y Huang (2018), quienes analizan las publicaciones de los estudiantes en el foro educativo del curso y relacionan estas actividades con el desempeño académico. La siguiente fuente de datos son las encuestas, las cuales son utilizadas por ejemplo en el trabajo de Klačnja-Milićević *et al.* (2011), quienes identifican diferentes estilos de aprendizaje utilizando un instrumento de 44 preguntas relacionadas con preferencias, intereses y hábitos de los estudiantes; y basados en esto, construyen un sistema de tutoría en programación que se adapta automáticamente a los intereses y niveles de conocimientos de los estudiantes. La

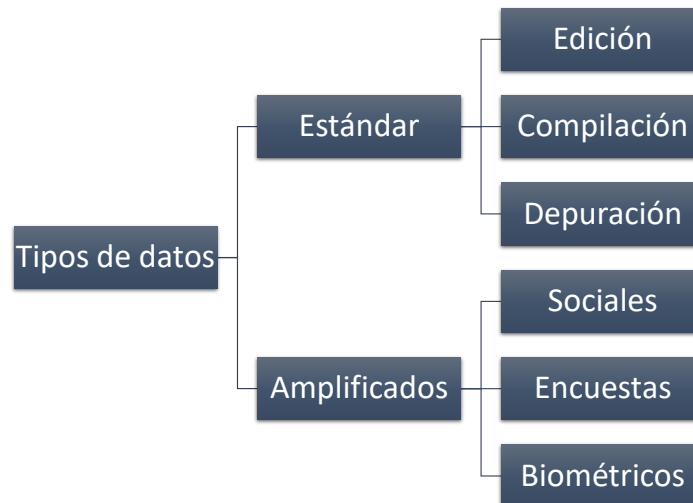


Figura 2-1: Tipos de datos recolectados en investigaciones de analítica de aprendizaje en cursos de programación. Adaptado de Carter *et al.* (2019).

última subcategoría de tipos de datos amplificados son los biométricos, donde se encuentra la investigación de Trættemberg *et al.* (2017) que emplean una combinación de información de entrevistas semi-estructuradas con técnicas de seguimiento ocular, e identifican los hábitos de depuración de los estudiantes; esto permite monitorear y visualizar el progreso y comportamientos de los alumnos durante actividades de programación.

Las investigaciones de analítica de aprendizaje en cursos de programación aplican diferentes intervenciones para cumplir sus objetivos. Una de las más utilizadas es la modelación y predicción de los comportamientos de los estudiantes. Un ejemplo es el trabajo realizado por Azcona *et al.* (2019), quienes proponen un modelo para detectar automáticamente alumnos en riesgo de no aprobar un curso de programación de computadores en Python, y a partir de las predicciones, ofrecer retroalimentación personalizada. Este tipo de intervenciones también se han aplicado en el aprendizaje de otros lenguajes de programación como C, utilizando indicadores de similitud entre los programas de código fuente construidos por los alumnos con el objetivo de identificar grupos de estudiantes con características similares (Berland *et al.*, 2015). El lenguaje Java también está presente en este tipo de intervenciones como en el trabajo realizado por Lagus *et al.* (2018), donde se compara el impacto que tiene utilizar algoritmos de transferencia de conocimiento en la predicción de estudiantes exitosos o en riesgo de no aprobar, y se comparan los resultados con técnicas tradicionales de aprendizaje de máquina.

El segundo tipo de intervenciones utilizadas consiste en realizar un seguimiento efectivo del progre-

so del curso, por medio de visualizaciones de estadísticas y métricas que representan los comportamientos de los estudiantes. Por ejemplo, en la investigación realizada por Shen *et al.* (2020), donde se visualiza con un mapa de calor la intensidad de acceso de los estudiantes a los recursos educativos y actividades en un MOOC introductorio en programación con Python, y por medio de análisis de redes sociales se examinan similitudes y diferencias. Otro ejemplo es el conjunto de visualizaciones propuestas por Leony *et al.* (2013) en un curso de programación del lenguaje C, el cual tiene como objetivo informar a los profesores sobre las emociones de los estudiantes con base en la interacción de los alumnos con las herramientas tecnológicas educativas utilizadas en la asignatura. A este tipo de intervenciones se suma el trabajo de Hsiao y Lin (2017), donde presentan un sistema inteligente capaz de extraer la información semántica de las respuestas de exámenes escritos en un curso de programación en Java. Adicionalmente, en este trabajo se implementa un algoritmo de indexación automática, el cual relaciona cada pregunta del examen a un conjunto de conceptos y se construye una interfaz de visualización interactiva donde se evidencian las asociaciones identificadas.

Por último, están las investigaciones que tienen como objetivo generar retroalimentación personalizada. El efecto de este tipo de intervenciones se puede evidenciar en el trabajo de Lu *et al.* (2017), donde se aplica analítica del aprendizaje para identificar estudiantes que necesitan intervención inmediata en un MOOC de Python, y los docentes construyen guías adaptativas de aprendizaje que son aplicadas en un grupo experimental y comparadas con un grupo de control. En otros casos se utilizan técnicas avanzadas de aprendizaje de máquina para construir sistemas de recomendación automática personalizada, con la capacidad de sugerir contenidos educativos a partir del análisis de patrones y comportamientos de los estudiantes (Klašnja-Milićević *et al.*, 2011). La Tabla 2-1 evidencia que la mayoría de las investigaciones educativas de analítica de aprendizaje revisadas fueron desarrolladas en cursos de programación del lenguaje Java. Sin embargo, en la revisión de literatura también se encuentran aplicaciones en cursos de C y Python.

2.3. Metodologías de analítica del aprendizaje

Varios autores han propuesto marcos metodológicos para la aplicación de investigaciones educativas de analítica del aprendizaje. En primer lugar, está el modelo cíclico propuesto por Clow (2012) que está basado en tres teorías educativas: el ciclo de aprendizaje experimental (Kolb, 1984), la teoría de la práctica reflexiva (Argyris y Schon, 1974; Schön, 1983, 1991) y el diseño de marco conversacional (Laurillard, 2002). Adicionalmente, se incorporan conceptos del área de ingeniería electrónica, específicamente las características de los sistemas de control de ciclo cerrado. El modelo propuesto es un ciclo cerrado donde los resultados de la investigación son las salidas, las cuales se comparan con un punto de referencia; por ejemplo, datos previos o resultados esperados. Posteriormente, con base en la comparación se diseñan y ejecutan intervenciones que modifican el mismo proceso de aprendizaje estudiado. En específico, el modelo metodológico cíclico propuesto está compuesto por las siguientes 4 etapas:

Tabla 2-1: Lenguajes de programación en investigaciones de analítica del aprendizaje

Lenguaje de programación	Referencias
Java	Blikstein <i>et al.</i> (2014) Blikstein (2011) Klašnja-Milićević <i>et al.</i> (2011) Trætteberg <i>et al.</i> (2017) Lagus <i>et al.</i> (2018) Hsiao y Lin (2017)
C	Scheffel <i>et al.</i> (2012) Hu y Huang (2018) Berland <i>et al.</i> (2015) Leony <i>et al.</i> (2013)
Python	Azcona <i>et al.</i> (2019) Shen <i>et al.</i> (2020) Lu <i>et al.</i> (2017)

1. Identificación del grupo de estudiantes: en esta etapa se considera el área del conocimiento, la institución donde se dicta el curso y la modalidad (presencial, virtual o semi-presencial).
2. Generación y captura de datos educativos: la información recolectada puede corresponder a información sobre los estudiantes o datos generados durante actividades educativas. Por lo general, estos datos son generados y almacenados en un LMS o VLE. Sin embargo, en algunos casos la recolección y almacenamiento de datos requiere herramientas adicionales externas.
3. Creación de métricas, visualizaciones y análisis: esta etapa consiste en transformar los datos disponibles en métricas o analíticas claras y concretas. Cabe resaltar, que estas métricas creadas deben generar un conocimiento profundo sobre el proceso de aprendizaje; de lo contrario, es posible promover comportamientos indeseados.
4. Creación y ejecución de intervenciones: este último paso es de gran importancia, dado que cierra el ciclo, retornando el producto generado al proceso inicial. Las intervenciones son diseñadas con base en las métricas y visualizaciones obtenidas, con el fin de generar un efecto en el comportamiento de los estudiantes que está definido por el objetivo de la investigación.

En segundo lugar, está el marco metodológico propuesto por Aljohani *et al.* (2019) el cual se basa en conceptos de ciencias del comportamiento, aprendizaje y computación cognitiva. El objetivo principal es el diseño de aplicaciones de analítica del aprendizaje que se adapten a los requerimientos específicos del curso. Este marco está dividido en los siguientes 4 niveles:

1. Nivel instructor: el docente especifica las herramientas del LMS que se utilizarán en el curso, basado en los objetivos, metodologías de enseñanza y métodos de evaluación de la asignatura. Seguido de esto, se identifica la información que va a ser recolectada. Posteriormente, se comunica y justifica a los estudiantes el uso de las herramientas seleccionadas, junto con una guía para su correcta utilización. Finalmente, se informa a los alumnos el proceso de seguimiento de sus interacciones con las herramientas.
2. Nivel datos: el objetivo es extraer datos del LMS, generados del uso de los estudiantes de las herramientas. El principal objetivo es aumentar el conocimiento sobre las actividades y comportamientos de los estudiantes.
3. Nivel de analítica de datos: dentro de este nivel se puede utilizar la computación cognitiva, la cual ofrece múltiples técnicas de análisis. La técnica aplicada es seleccionada con base en el objetivo de la investigación. Las técnicas disponibles incluyen análisis estadístico, aprendizaje de máquina, análisis de redes sociales y analítica visual.
4. Nivel de presentación: el análisis de datos es retornado a los usuarios interesados, como los estudiantes o profesores, y presentado de diferentes maneras. Entre los tipos de presentación están los tableros virtuales, los cuales visualizan la información más relevante, consolidada y organizada en una misma pantalla para facilidad de monitoreo. La retroalimentación presentada puede ser de diferente tipo como estadísticas, texto o escrito, pero lo importante es que se adapte al tipo de interfaz con la que están familiarizados los estudiantes.

Otra de las metodologías es la propuesta por Carter *et al.* (2019) que se basa en conceptos del área de reconocimiento del comportamiento humano y experiencias de cursos que emplean IDEs. El modelo propuesto es un proceso cíclico conformado por las siguientes 5 etapas:

1. Operacionalización de comportamientos observables: en esta etapa se identifican los datos que representan los comportamientos y resultados de los estudiantes. Esta fase permite establecer qué información permite lograr el objetivo de la investigación.
2. Recolección de datos: teniendo en cuenta la gran variedad de mecanismos para la generación y captura de información, se establecen dos categorías de tipos de datos. En primer lugar, los datos estándar, es decir, los generados durante el desarrollo de actividades de programación. La taxonomía propuesta establece cuatro clases: edición, compilación, ejecución y depuración. La otra categoría son los datos amplificados, que corresponden a la información generada cuando las funcionalidades de IDE son extendidas por medio de herramientas externas. La taxonomía propuesta divide estos datos en las siguientes cuatro sub-categorías: sociales, testeo, encuestas y físicos del individuo.
3. Análisis de datos: en esta etapa los datos obtenidos se transforman en analíticas o métricas, las cuales generan información útil para diseñar intervenciones que mejoren el proceso de aprendizaje. Se identifican los siguientes 6 tipos de métricas: habilidades de programación, contenido del código fuente, comportamiento social, conocimientos, actitudes e información ocular.

Adicionalmente, las técnicas disponibles para el análisis de los datos son: conteo, métodos matemáticos, algoritmos de predicción, visualizaciones y aplicaciones de aprendizaje de máquina.

4. Diseño de intervenciones: en esta etapa se crean espacios donde información relevante, sugerencias y/o retroalimentación es compartida, con el objetivo de influenciar positivamente el comportamiento o actitud de los alumnos. Las tres características fundamentales en el diseño de intervenciones son: contenido, presentación y tiempo.
5. Ejecución de intervenciones: en esta última fase se aplican las intervenciones diseñadas, con el fin de generar el impacto esperado en el proceso de aprendizaje de los estudiantes. Finalmente, se resalta que los procesos de analítica de aprendizaje son iterativos, puesto que los cambios educativos esperados no son inmediatos sino graduales.

Por otra parte, el trabajo realizado por Ithantola *et al.* (2015) establece una arquitectura de los sistemas y subsistemas presentes en investigaciones de analítica del aprendizaje aplicadas en cursos de ciencias de la computación. La fuente principal de información es una revisión sistemática de literatura de trabajos publicados en el área, enfocados en la enseñanza de programación de computadores y que aplican herramientas de recolección automática de datos. En este marco se identifican todos los elementos e interacciones tanto de los estudiantes cuando desarrollan soluciones a problemas de programación, como de los docentes o investigadores cuando analizan los datos generados. Las cinco dimensiones especificadas en el marco propuesto son:

- **Acciones:** Corresponde a la primera interacción que tienen los estudiantes con un IDE o VLE, el cual permite la construcción, ejecución y envío de soluciones de ejercicios o actividades de programación.
- **Retroalimentación:** cuando la solución construida es enviada, el alumno puede recibir una calificación numérica o retroalimentación detallada sobre la calidad del programa construido. Esta retroalimentación puede ser entregada por medio de una plataforma de calificación automática externa o dentro del mismo VLE o IDE utilizado para el envío.
- **Almacenamiento:** muchos IDE y VLE incorporan herramientas de recolección de datos. La información puede estar almacenada en diferentes formatos, desde archivos planos simples hasta bases de datos soportadas por servidores. Adicionalmente, se debe identificar cuál es la información o atributos de los datos que son relevantes para el objetivo de la investigación.
- **Exportación:** la mayoría de herramientas de almacenamiento disponen también de mecanismos de exportación de datos. En este caso, se simplifica el flujo de trabajo para el proceso de análisis.
- **Visualización:** finalmente, se pueden aprovechar los sistemas que permiten generar resúmenes, análisis exploratorio o visualizaciones de los datos recolectados. Este procesamiento de los datos es bastante útil tanto para profesores como para investigadores en el área educativa.

Por último, está el modelo propuesto por Siemens (2013). Este trabajo tiene como objetivo generar una arquitectura generalizable, la cual utilice un enfoque *tow-down* que permita la sistematización de los recursos educativos utilizados. La fuente de información utilizada es la literatura de aplicaciones de analítica del aprendizaje hallada, donde se considera el desarrollo histórico, y las herramientas y técnicas utilizadas en este tipo de investigación. El modelo propuesto es cíclico e incluye los siguientes siete componentes:

1. **Recolección y adquisición:** en esta primera etapa, las especificaciones dependen del objetivo establecido de la aplicación de analítica de aprendizaje. Entre los objetivos están: mercadeo, publicidad, aprendizaje, administración o investigación institucional. Entre las fuentes de datos disponibles encontramos: Bases de datos, LMS, sistemas de información de estudiantes (SIS, por sus siglas en inglés), sensores y entradas manuales.
2. **Almacenamiento:** en esta etapa se debe considerar la ubicación y formato de la información recolectada.
3. **Limpieza de datos:** teniendo en cuenta que los datos almacenados pueden ser tanto estructurados como no estructurados, se debe considerar un proceso de depuración de la información.
4. **Integración:** considerando que varias aplicaciones de analítica del aprendizaje consideran múltiples fuentes de datos, es pertinente realizar la consolidación de todos los conjuntos de datos considerados. Adicionalmente, el formato de almacenamiento de la información debe ser homogéneo, con el objetivo de facilitar el manejo y procesamiento de datos.
5. **Análisis:** para realizar este paso, se tienen disponibles varias herramientas y técnicas como: análisis de redes sociales (SNA, por sus siglas en inglés), procesamiento de lenguaje natural (NLP, por sus siglas en inglés), desarrollo de concepto, predicción, determinación del riesgo, secuenciación de curso y búsqueda de ayuda.
6. **Representación y visualización:** en esta etapa, el análisis de los datos se hace visible a los actores interesados (e.g. estudiantes, docentes, administradores educativos, etc.).
7. **Acción:** finalmente, los resultados obtenidos de los pasos anteriores son utilizados en el proceso de toma de decisiones educativas; esto se puede generar en forma de intervenciones, optimizaciones, sistemas de alerta, guías, sistematizaciones, modificaciones en el diseño curricular o métodos de enseñanza.

La revisión de metodologías de analítica del aprendizaje realizada sirve como fuente de información para la construcción del diseño metodológico del presente proyecto. En este sentido, se realiza una comparación de los modelos revisados considerando los siguientes criterios: objetivo, contexto, fuentes de información y año de publicación. El aspecto de objetivo se refiere a la finalidad del diseño presentado, es decir, el valor agregado de la propuesta metodológica. El criterio de contexto indica el área donde el modelo propuesto puede ser aplicado. Las fuentes de información son las bases teóricas o evidencias experimentales utilizadas en la construcción de la metodología propuesta. La Tabla

2-2 resume las características de cada modelo revisado con respecto a los 4 criterios de comparación establecidos.

Tabla 2-2: Cuadro comparativo de las metodologías de analítica de aprendizaje revisadas.

Referencia	Criterios de comparación			Año
	Objetivo	Contexto	Fuentes	
Clow (2012)	Mayor efectividad, velocidad y escala de aplicación de intervenciones.	Multidisciplinar	<ul style="list-style-type: none"> ■ Teorías educativas. ■ Sistemas de control electrónicos. 	2012
Aljohani <i>et al.</i> (2019)	Método específico para diseño de herramientas de analítica educativa.	Multidisciplinar	<ul style="list-style-type: none"> ■ Ciencias del comportamiento. ■ Ciencias del aprendizaje. ■ Computación cognitiva. 	2019
Carter <i>et al.</i> (2019)	Metodología cíclica en cursos de programación con IDEs.	Ciencias de la computación.	<ul style="list-style-type: none"> ■ Reconocimiento de comportamientos humanos. ■ Expertos en educación computacional. ■ Investigadores en analítica del aprendizaje. 	2019
Ihantola <i>et al.</i> (2015)	Estructura de interacción con herramientas de recolección automática de datos.	Ciencias de la computación	<ul style="list-style-type: none"> ■ Revisión sistemática de literatura. ■ Investigadores en analítica del aprendizaje. 	2015
Siemens (2013)	Modelo con enfoque generalizado	Multidisciplinar	<ul style="list-style-type: none"> ■ Revisión de literatura de aplicaciones de analítica del aprendizaje. 	2013

En primer lugar, el objetivo del modelo de Clow (2012) está encaminado a aumentar la efectividad de la aplicación de la analítica del aprendizaje por medio intervenciones de mayor impacto. Sin embargo, esta propuesta se concentra en el diseño y ejecución de intervenciones educativas, las cuales no se encuentran dentro del alcance de este proyecto. Los diseños de Aljohani *et al.* (2019) y Carter *et al.* (2019) plantean etapas similares y los planteamientos se acoplan a las necesidades del presente proyecto. No obstante, en el caso de Aljohani *et al.* (2019) el objetivo está encaminado hacia la construcción de una herramienta para analítica de los datos, lo que limita la generalización y aplicación del modelo. Por otro lado, la metodología de Carter *et al.* (2019) está complementada con taxono-

mías de tipos de datos y técnicas de análisis disponibles para este tipo de investigación. La ventaja de este modelo es que las métricas, técnicas de análisis y modelación propuestas pueden usarse en las especificaciones del modelo final construido. Con respecto al modelo propuesto por Ithantola *et al.* (2015), se plantea una estructura general con las interacciones entre los estudiantes y docentes con los sistemas y herramientas de recolección automática de datos educativos. No obstante, no se especifica un proceso secuencial a seguir en aplicaciones de analítica del aprendizaje. En este aspecto del objetivo, el método propuesto por Siemens (2013) tiene la mayor ventaja, ya que se plantea como una metodología generalizable; intentando utilizar un enfoque desde lo general hacia lo específico para garantizar su aplicabilidad en diferentes contextos educativos.

Con respecto al contexto de aplicación, la mayoría de las metodologías están construidas de manera general, es decir, que se pueden adaptar a cursos de distintas áreas del conocimiento. Por otra parte, dos de los modelos revisados tienen un dominio de aplicación específico en cursos de ciencias de la computación, los cuales corresponden a los propuestos por Carter *et al.* (2019) e Ithantola *et al.* (2015). Adicionalmente, estas últimas dos metodologías consideran cursos que utilizan IDE y herramientas de recolección automática de datos educativos, respectivamente.

Las fuentes de información de los modelos consideran tanto bases conceptuales robustas como experiencias empíricas expertas. Algunos autores como Aljohani *et al.* (2019) y Clow (2012) y Carter *et al.* (2019) se basan en conceptos específicos de teorías educativas sobre los procesos de aprendizaje, pero solo los dos últimos consideran los fundamentos de las ciencias relacionadas con el comportamiento humano. Sumado a esto, los modelos de Carter *et al.* (2019) e Ithantola *et al.* (2015) tienen el valor agregado de considerar la opinión de expertos en aplicaciones de analítica del aprendizaje e investigaciones educativas. Adicionalmente, las metodologías de Ithantola *et al.* (2015) y Siemens (2013) son construidas con base en revisiones de literatura de modelos aplicados en investigaciones similares. El hecho de considerar la opinión de investigadores y búsqueda de referencias como fuentes de información es una ventaja, puesto que el diseño metodológico de este proyecto requiere un enfoque práctico.

Por último, acorde con el año de publicación los diseños cíclicos presentados por Clow (2012) y Siemens (2013) son los trabajos de mayor antigüedad en comparación con las demás propuestas. En contraste, los modelos de Aljohani *et al.* (2019) y Carter *et al.* (2019) son los más actuales puesto que ambos fueron publicados hace menos de tres años. La ventaja de las metodologías más recientes es que están basadas en un estado del arte más actualizado.

2.4. Trabajos relacionados

De forma similar al objetivo de la presente tesis, varias de las investigaciones en analítica del aprendizaje se concentran en identificar cuáles de las variables del proceso de aprendizaje (e.g. interacción con

plataformas virtuales educativas, uso de herramientas disponibles para el desarrollo de actividades, datos demográficos, etc.) se encuentran relacionadas con el desempeño académico de los estudiantes. La Tabla 2-3 resume los hallazgos de algunas investigaciones que buscan correlaciones entre la información sobre el proceso de aprendizaje con el desempeño final de los estudiantes.

Tabla 2-3: Variables correlacionadas con el desempeño académico encontradas en investigaciones de analítica del aprendizaje

Variable	Referencia					
	Zacharis (2015)	Azcona et al. (2019)	Andergassen et al. (2014)	Barber y Sharkey (2012)	Macfadyen y Dawson (2010)	Vahdat et al. (2015)
Total de intentos enviados	✓ ⁺					
Total de respuestas correctas		✓ ⁺				
Tiempo promedio entre intentos			✓ ⁺			
Género				✗		
Edad		✗		✗		
Accesos a calificaciones	✗				✗	
Complejidad ciclomática						✓ ⁻

✓⁺ Correlación positiva estadísticamente significativa.

✓⁻ Correlación negativa estadísticamente significativa.

✗ Correlación no significativa a nivel estadístico.

En este grupo se encuentra el trabajo realizado por Zacharis (2015) que desarrolla un modelo para predecir los estudiantes en riesgo de obtener bajos desempeños académicos, a partir de los datos sobre el uso de la plataforma Moodle utilizada en la clase. El análisis realizado obtiene que el número de entregas realizadas por estudiante tiene una correlación positiva de 0.34 ($p\text{-value} = 0.01$) con la calificación final de los alumnos. Por otra parte, la cantidad de accesos a la herramienta de calificaciones no evidencia una correlación significativa con el desempeño académico ($p\text{-value} = 0.26$). Otras investigaciones, como la de Azcona et al. (2019), presentan una metodología para detectar estudiantes en riesgo de reprobación de las actividades en un curso de programación y apoyar la retroalimentación adaptativa. Este trabajo construye un modelo predictivo usando información estática (e.g. datos demográficos) y datos dinámicos (e.g. registros de actividades y comportamiento de programación). Los

resultados de esta investigación evidencian que el porcentaje acumulado de ejercicios correctos tiene un coeficiente de correlación de Pearson de 0.67 ($p\text{-value} < 0.0001$). Por otra parte, no se encuentra una correlación significativa entre la edad y las calificaciones finales de los estudiantes. Otros autores como Andergassen *et al.* (2014) investigan las correlaciones potenciales entre los resultados de aprendizaje y el uso de LMS durante la preparación de exámenes, enfocado en práctica y repetición. Los resultados de este trabajo muestran una correlación positiva de 0.18 ($p\text{-value} < 0.0053$) entre el tiempo promedio de diferencia entre repeticiones (i.e. intentos) de ejercicios con la calificación del examen final.

En este grupo de investigaciones también se encuentra el modelo predictivo de Barber y Sharkey (2012). Estos autores utilizan información demográfica en la arquitectura del modelo para calcular la probabilidad de que un estudiante repruebe un curso. Esta investigación evidencia que el género y la edad de los estudiantes no son variables estadísticamente significativas para predecir el desempeño académico. Otro ejemplo de este tipo de investigaciones es la investigación de Macfadyen y Dawson (2010), donde se analizan los datos de seguimiento de uso de un LMS usado en un curso que utiliza *Blackboard-Vista*. Los resultados de esta investigación evidencian que el número de accesos a la herramienta de calificaciones *MyGrades* no tiene una correlación significativa ($p\text{-value} = 0.06$) con la nota final de los estudiantes. Finalmente, está el trabajo desarrollado por Vahdat *et al.* (2015) que tiene como objetivo entender el comportamiento de estudiantes de ingeniería de sistemas y computación en un curso donde se usa un simulador de circuitos. Los resultados de esta investigación evidencian que la complejidad ciclomática calculada por sesión de clase, tiene una correlación negativa con las calificaciones promedio de los estudiantes.

2.5. Desafíos de la analítica del aprendizaje

La analítica del aprendizaje en un área de conocimiento relativamente nueva, que a tomado forma en la última década, por esta razón aun existen varios retos a superar en esta disciplina. Acorde con el estado del arte construido por Ang *et al.* (2020), los desafíos en la analítica del aprendizaje se clasifican en dos categorías: sociales y tecnológicos. En el área de problemáticas sociales, el análisis de información educativa tiene implicaciones de privacidad, debido a que el manejo de datos personales de los estudiantes puede ser utilizado con fines diferentes a los de investigación (Ang *et al.*, 2020; Dascalu *et al.*, 2018). En este sentido, autores como Pardo y Siemens (2014) identifican tres subcategorías dentro de los desafíos de privacidad:

1. Transparencia sobre la información recolectada y el proceso que se va a realizar.
2. Control de los datos obtenidos donde los estudiantes puedan corregir la información.
3. Política de seguridad que define un conjunto usuarios con derechos de acceso y las operaciones permitidas.

Por otra parte, en el conjunto de problemáticas sociales se encuentran los desafíos éticos. En primer lugar, se encuentran desafíos relacionados con el uso de conjuntos de datos sesgados, los cuales pueden generar modelos predictivos incompletos, porque no se consideran todas las variables necesarias para garantizar la exactitud de los resultados (Dascalu *et al.*, 2018; Selwyn, 2020). Sumado a esto, está la problemática que predecir incorrectamente el comportamiento de los alumnos, lo que genera ideas preconcebidas que pueden impactar negativamente la motivación y las oportunidades educativas disponibles para los estudiantes (Ang *et al.*, 2020; Dascalu *et al.*, 2018). Las implicaciones éticas también se evidencian en aplicaciones de analítica del aprendizaje donde se compara un grupo experimental con uno de control, puesto que un grupo puede perder oportunidades de aprendizaje potencialmente beneficiosas (Drachler y Greller, 2016). Por último, otro desafío ético es no considerar grupos de estudiantes de alta diversidad, debido a que el avance en la educación ha permitido la inclusión de nuevos alumnos no tradicionales (e.g. trabajadores de medio o tiempo completo), los cuales tienden a ser poco estudiados en este tipo de investigaciones (Ferguson, 2012; Schmitz *et al.*, 2017).

La segunda categoría de desafíos en analítica de aprendizaje son las problemáticas tecnológicas. En primer lugar, está la falta de interoperabilidad entre instituciones. En otras palabras, debido a que varias instituciones han empezado a construir sus propios conjuntos de datos educativos con diferentes tipos de información, almacenamiento y contextos, se ha generado una carencia de políticas inter-institucionales para compartir información e integrar los sistemas educativos construidos (Ang *et al.*, 2020; Dascalu *et al.*, 2018; Schmitz *et al.*, 2017). Adicionalmente, las aplicaciones de analítica de aprendizaje tienen dificultades en la escalabilidad de las intervenciones diseñadas, debido a las limitaciones en las condiciones de implementación y en la evaluación de la efectividad (Najafabadi *et al.*, 2016; Wong y Li, 2020).

Finalmente, dentro de los desafíos técnicos se ha evidenciado una concentración del uso de métodos de investigación cuantitativos en comparación con los métodos cualitativos o mixtos en investigaciones de analítica del aprendizaje (Mangaroska y Giannakos, 2017). Esta problemática concuerda con las conclusiones de Tempelaar *et al.* (2016), los cuales afirman que la alta dependencia de resultados de una única fuente de datos es uno de los principales desafíos que necesitan ser solucionados en el futuro de las investigaciones educativas. La presente investigación se enfoca precisamente en este último reto.

3 Diseño metodológico mixto para analítica del aprendizaje

En este capítulo se presenta el diseño metodológico propuesto para la aplicación de analítica del aprendizaje mediante un enfoque de investigación de métodos mixtos. En primer lugar, la Sección 3.1 del capítulo especifica los fundamentos conceptuales y el esquema general de la propuesta metodológica. Las secciones posteriores describen las fases, etapas y actividades presentes en la metodología propuesta de investigaciones educativas de analítica del aprendizaje con enfoque de investigación de métodos mixtos. En este sentido, la Sección 3.2 detalla las etapas de preparación, transformación y análisis de la fase de datos cuantitativos. Posteriormente, la Sección 3.3 especifica las mismas tres etapas pero en la fase de datos cualitativos. Por último, la Sección 3.4 describe cómo llevar a cabo la fase de discusión de resultados, donde se realiza una integración e interpretación de los hallazgos de las dos fases anteriores.

3.1. Marco metodológico para analítica de aprendizaje

Establecer el marco metodológico de un proyecto es de gran importancia, dado que delimita el alcance de las etapas de la investigación. A partir de la revisión de literatura de metodologías de analítica del aprendizaje presentada en la Sección 2.3, se identifican las características fundamentales que debe tener el diseño metodológico propuesto. Se incluyen características de las 6 primeras etapas del modelo de Siemens (2013), ya que es el más generalizable. Estas etapas se refieren a: recolección y adquisición de datos, almacenamiento, limpieza de datos, integración, análisis de datos, y finalmente, representación y visualización de datos. También, se incorporan elementos propios de la segunda y tercera etapa de los modelos propuestos por Clow (2012), Aljohani *et al.* (2019) y Carter *et al.* (2019); las cuales reúnen actividades encaminadas a la recolección de datos educativos y al análisis y visualización de los datos. Cabe resaltar que las taxonomías de tipos de datos disponibles, métricas y técnicas de análisis propuestas por Carter *et al.* (2019) son también consideradas en la construcción de la metodología de este proyecto. Adicionalmente, a partir del diseño de Ihantola *et al.* (2015) se adoptan algunas especificaciones de los niveles de almacenamiento, exportación y visualización de datos.

Además, el diseño propuesto para investigaciones educativas se complementa mediante un enfoque de investigación de métodos mixtos, y en particular, se sugiere un tipo de estudio secuencial expli-

cativo para este tipo de investigaciones (Bryman, 2015; Creswell, 2014). La metodología secuencial explicativa utiliza los resultados encontrados mediante el uso de métodos cualitativos para encontrar una explicación probable a los hallazgos encontrados mediante el uso de métodos cuantitativos. Permite explorar los resultados en mayor detalle y, en muchas ocasiones, sirve de ayuda para interpretar resultados no esperados.

A partir de los criterios de comparación utilizados en la Tabla 2-2 se define el objetivo de la metodología construida en este capítulo. El objetivo es proponer un marco metodológico donde se aplique la analítica del aprendizaje utilizando un enfoque mixto de investigación. El contexto de aplicación son cursos de programación de computadores donde se utilicen plataformas educativas que permitan la recolección y almacenamiento de datos sobre las interacciones de los estudiantes. Las fuentes de información utilizadas son la revisión de literatura sobre metodologías de analítica de aprendizaje en diversos contextos y el marco metodológico utilizado en investigaciones con enfoque mixto. Con respecto al año de publicación, esta metodología correspondería a la más actualizada en comparación con los trabajos analizados en la Tabla 2-2.

La Figura 3-1 ilustra el marco metodológico propuesto, el cual consiste en tres fases globales secuenciales representadas por los recuadros grandes de colores en la figura. La primera fase corresponde al enfoque cuantitativo, la segunda fase engloba el enfoque cualitativo y la tercera fase abarca la integración de las dos anteriores fases junto con la discusión de resultados. A su vez, las dos primeras fases se dividen en tres etapas que son representadas dentro de la figura con las líneas negras punteadas; estas etapas corresponden a preparación, transformación y análisis de datos. Las fases y etapas contienen actividades específicas, las cuales son representadas por los recuadros pequeños de colores en la figura. Las actividades son definidas acorde con la naturaleza de la fase correspondiente. A continuación se describen en detalle cada una de las actividades de la metodología propuesta.

3.2. Fase 1: Datos cuantitativos

3.2.1. Recolección de datos cuantitativos

La recolección de información es la primera actividad de la etapa de preparación dentro de la fase de datos cuantitativos. Cabe resaltar que, la recopilación de datos depende del objetivo de la aplicación de analítica de aprendizaje propuesta en el trabajo. En primer lugar, se especifica la temporalidad de la generación de información, donde se tienen dos opciones: datos históricos o datos actuales. Los datos históricos pueden ser utilizados en investigaciones donde el objetivo es observar los comportamientos de los estudiantes sin modificar su contexto. Por otra parte, los datos actuales pueden ser usados para comparar el efecto de una intervención educativa en un grupo de estudiantes. En segundo lugar, se identifica el tipo de datos que van a ser recolectados, donde se tienen dos categorías: estáticos y dinámicos.

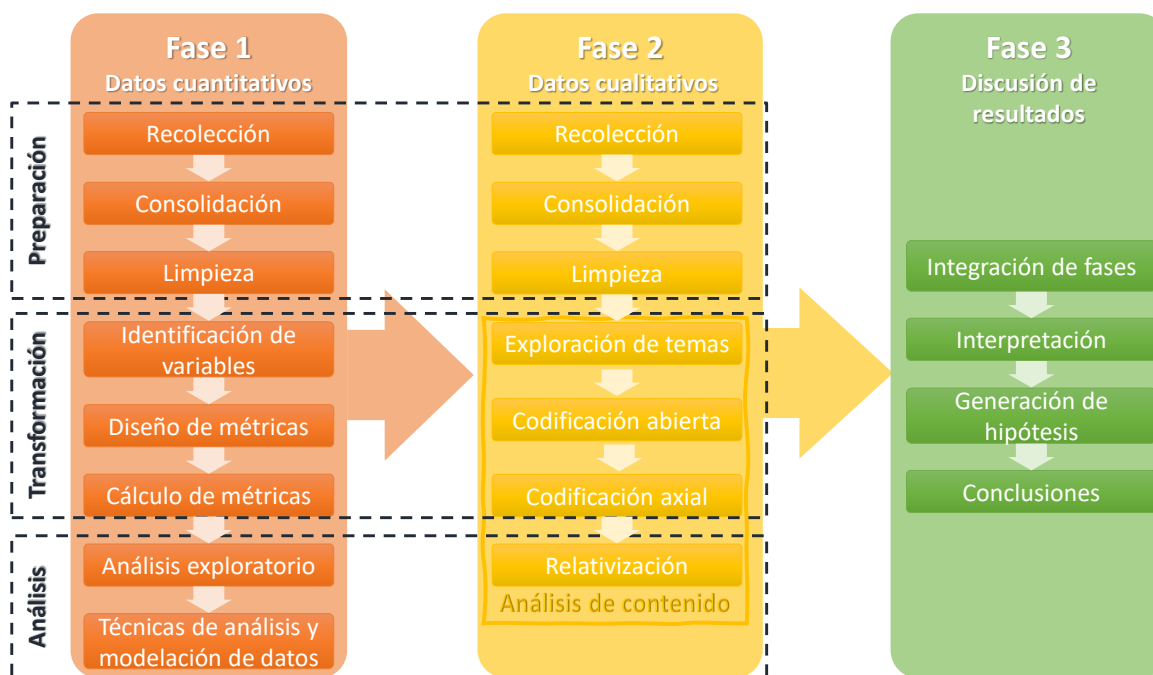


Figura 3-1: Marco metodológico propuesto para analítica de aprendizaje mediante un enfoque de investigación de métodos mixtos.

La información estática corresponde a los datos que son adquiridos en un punto fijo del proceso de aprendizaje, como información personal y académica sobre los alumnos por medio de fuentes de información como bases de datos, repositorios académicos y sistemas de información de estudiantes (SIS, por sus siglas en inglés). Cabe resaltar que cuando se utilizan fuentes de datos donde se consideran demasiadas variables como en bases de datos educativas o conjuntos de datos públicos, es necesario filtrar las variables que son relevantes para el objetivo de la investigación. En contraste, los datos dinámicos son los generados durante actividades educativas, como las interacciones y uso de herramientas del LMS, VLE o IDE durante el curso. Adicionalmente, las fuentes de información pueden ser clasificadas acorde con la taxonomía propuesta por Carter *et al.* (2019), donde se establecen dos categorías de tipos de datos: 1) estándar, que se divide en cuatro clases: edición, compilación, ejecución y depuración; y 2) amplificados, que tienen las siguientes cuatro sub-categorías: sociales, pruebas, encuestas y biométricas.

Una vez especificado el tipo de información a recolectar para cumplir con el objetivo del trabajo, se define la ubicación y formato de almacenamiento de los datos. Generalmente, los datos educativos son almacenados en la misma plataforma donde son generados, puesto que algunos LMS, IDE, VLE poseen herramientas de recolección. No obstante, en algunos casos el almacenamiento de datos re-

quiere instrumentos adicionales para permitir el manejo y análisis de datos. Estos instrumentos deben ser seleccionados en función de las necesidades del grupo de trabajo, entre las posibilidades está el almacenamiento local o las plataformas de almacenamiento de archivos en la nube (e.g., Google Drive, Dropbox, Mega, etc.). Finalmente, se debe establecer el formato de los archivos almacenados que puede ser desde archivos de texto plano hasta bases de datos complejas. Acorde con el tipo de datos seleccionado se determina el formato más adecuado para su manejo, entre los cuales están archivos de texto, imágenes, vídeos, audio, datos comprimidos y códigos fuente.

3.2.2. Consolidación del conjunto de datos cuantitativos

La consolidación de la información es la segunda actividad dentro de la etapa de preparación de la fase de datos cuantitativos. En este sentido, se tiene en cuenta que las investigaciones de analítica del aprendizaje pueden utilizar múltiples tipos y fuentes de datos cuantitativos. Inclusive en los casos en lo que se usa solo una fuente de información, los datos pueden estar desagregados en función del tiempo, grupo de estudiantes o institución. Teniendo en cuenta esto, se deben consolidar todos los datos recolectados y almacenados. Los investigadores deben establecer un formato homogéneo de almacenamiento de los datos, con el objetivo de mejorar la organización, manejo y procesamiento del conjunto de datos completo, evitando la pérdida de información.

3.2.3. Limpieza del conjunto de datos cuantitativos

La limpieza del conjunto de datos corresponde a la tercera actividad de la etapa de preparación dentro de la fase de datos cuantitativos. Esta actividad tiene como objetivo depurar la información no relevante o significativa para el trabajo. Por lo general, las fuentes de datos disponibles para este tipo de investigación poseen información no estructurada o sin procesar, esto se debe a que los procesos de adquisición de grandes volúmenes de datos tienden a tener errores (e.g. digitación incorrecta, almacenamiento insuficiente, funcionamiento incorrecto de la plataforma, etc). En este orden de ideas, primero es necesario realizar una revisión del conjunto de datos consolidado, identificando datos faltantes, incompletos, incorrectos, atípicos o no relevantes para entender los procesos de aprendizaje. Este análisis inicial de los datos se puede realizar por medio del cálculo de medidas de la estadística descriptiva, tales como: media, desviación estándar, rango, mínimo, máximo, moda; y representaciones gráficas como histogramas, diagramas de barras, diagramas de cajas, diagramas de dispersión.

Una vez identificados los datos con problemas de calidad o erróneos, las posibles acciones para su tratamiento son: substituir, modificar o eliminar. Los investigadores deben decidir la acción correctiva más apropiada acorde con las características de los datos. Por ejemplo, en una investigación que considere las edades de los estudiantes, los valores faltantes podrían llegar a ser substituidos por el promedio de los datos de los demás estudiantes. Esto permitiría conocer una aproximación a la medida real.

3.2.4. Identificación de variables de interés

La identificación de variables de interés es la primera actividad de la etapa de transformación de la fase de datos cuantitativos. Esta actividad consiste en realizar una revisión más detallada de la información disponible en el conjunto de datos consolidado y depurado, identificando la información de interés para la investigación. En este orden de ideas, se debe examinar individualmente cada una de las variables presentes en el conjunto de datos, reflexionando sobre su posible relación sobre el proceso de aprendizaje y su utilidad en el desarrollo de la investigación. Esta selección se debe realizar en función del objetivo de la aplicación de analítica de aprendizaje y en las variables de interés, tales como: desempeño académico, motivación, compromiso, satisfacción, capacidades meta-cognitivas, etc.

En el caso de investigaciones fenomenológicas, es decir, que tienen como fin examinar los comportamientos de los estudiantes en su contexto normal, sin intervenciones, el conjunto de variables tiende a considerar un rango más amplio; en comparación con trabajos enfocados en evidenciar el efecto de una intervención educativa en un aspecto específico de los alumnos.

3.2.5. Diseño de métricas de interés

El diseño de métricas es la segunda actividad de la etapa de transformación de la fase de datos cuantitativos. Esta actividad consiste en transformar las variables identificadas en la actividad anterior en métricas que generen conocimiento específico sobre el proceso de aprendizaje de los estudiantes. El proceso de generación de métricas puede estar complementado por una revisión de literatura de métricas utilizadas en otras aplicaciones de analítica del aprendizaje, ya que de su correcto diseño depende la relevancia de la información adquirida sobre los estudiantes.

En primer lugar, se determina el tipo de métricas de interés con base en la taxonomía presentada por Carter *et al.* (2019). Esta taxonomía establece que las métricas se pueden categorizar en: habilidades de programación (e.g., manejo del tiempo, número de ejecuciones, cantidad de código agregado o modificado, etc.), contenido del código fuente (e.g., tamaño del programa construido, cantidad de palabras clave, similitud con otros programas, resultados de casos de prueba, etc.), comportamiento social (e.g., participación en foros, contenido de mensajes publicados, cantidad de preguntas realizadas o solucionadas), conocimientos (e.g., número de respuestas correctas o incorrectas), actitudes (e.g., motivación, emociones reportadas, etc.) y movimientos oculares (e.g., tiempo de fijación en pantalla, tiempo de distracción, etc.).

3.2.6. Cálculo de métricas diseñadas

El cálculo de métricas es la tercera actividad de la etapa de transformación de la fase de datos cuantitativos. Cabe resaltar que posterior a la selección de las métricas, se identifican las técnicas que

necesitan ser aplicadas para su cálculo, que acorde con Carter *et al.* (2019) pueden ser las siguientes cinco: conteo, ecuaciones matemáticas, algoritmos, visualizaciones y aprendizaje de máquina. En este sentido, se plantean en concreto las ecuaciones o procedimientos necesarios para la cuantificación de las métricas diseñadas. Los valores obtenidos pueden ser almacenados en la misma ubicación y en el mismo formato utilizado en la recolección de datos.

3.2.7. Análisis exploratorio de datos

El análisis exploratorio de datos corresponde a la primera actividad de la etapa de análisis de la fase de datos cuantitativos. Esta actividad tiene como objetivo entender en detalle los comportamientos y tendencias de las variables y métricas de interés para la investigación. Sin embargo, esta es una primera aproximación, dado que se enfoca en realizar un análisis univariado de cada variable y métrica, sin tener en cuenta aun el efecto que tienen las demás variables en su comportamiento. Para esto, es necesario seleccionar una o múltiples técnicas para realizar el análisis, entre las cuales están: estadística descriptiva (i.e., medidas de tendencia central, medidas de dispersión, medidas de forma de distribución, etc.) y visualizaciones (e.g., diagramas de cajas, histogramas, etc.). Posteriormente, a partir de los resultados obtenidos, los investigadores deben obtener conclusiones parciales sobre los procesos de aprendizaje de los estudiantes, y disponerse a realizar nuevos análisis más complejos involucrando análisis multivariados.

3.2.8. Aplicación de técnicas de análisis y modelación de datos

Las técnicas de análisis de modelación corresponden a la última actividad dentro de la etapa de análisis de la fase de datos cuantitativos. Esta actividad está determinada por el objetivo del proyecto. En este sentido, se determina la finalidad de la aplicación de analítica del aprendizaje, el cual puede ser el análisis detallado del comportamiento evidenciado de una o varias variables, o la predicción de una variable de interés con base en datos históricos. En el caso del análisis profundo, es decir, investigaciones que tienen como finalidad entender las relaciones entre las variables y métricas consideradas, hay herramientas disponibles como: pruebas de hipótesis, análisis de correlaciones, de cadena causal o de redes sociales (SNA, por sus siglas en inglés), procesamiento de lenguaje natural (NLP, por sus siglas en inglés), entre otras. En contraste, cuando el trabajo está enfocado en obtener un modelo de clasificación o de regresión es necesario especificar la variable dependiente y el conjunto de variables independientes; algunas de las técnicas que pueden ser aplicadas son: *clustering*, máquinas de soporte vectorial (SVM, por sus siglas en inglés), redes neuronales recurrentes (RNN, por sus siglas en inglés), regresión múltiple y algoritmos genéticos.

3.3. Fase 2: Datos cualitativos

3.3.1. Recolección de datos cualitativos

La recolección de datos es la primera actividad de la etapa de preparación dentro de la fase de datos cualitativos. Esta primera actividad se define con base en el objetivo general de la aplicación de analítica de aprendizaje de la investigación. En este orden de ideas, se debe especificar primero el diseño de investigación que va a ser aplicado. Acorde con Hernández-Sampieri *et al.* (2014) los diseños de investigación disponibles para metodologías cualitativas son: teoría fundamentada, estudio etnográfico, narrativo, fenomenológico e investigación acción - participativa. En segundo lugar, se identifica la muestra poblacional de interés. En este caso, puesto que la metodología general es mixta, el conjunto de individuos del que se recolectan datos deben corresponder al mismo grupo seleccionado en la fase de los métodos cuantitativos, puesto que los datos recolectados deben corresponder al mismo contexto de estudio para que los resultados puedan ser integrados posteriormente. Finalmente, se identifica la fuente de datos que van a ser recolectados, entre las cuales están: anotaciones y bitácoras de observaciones, entrevistas, encuestas, grupos focales, testimonios, documentos (escritos, audio, vídeos, etc).

3.3.2. Consolidación del conjunto de datos cualitativos

La consolidación del conjunto de datos es la segunda actividad de la etapa de preparación dentro de la fase de datos cualitativos. Esta actividad inicia con el almacenamiento de todos los datos recolectados, para lo cual se define la ubicación de almacenamiento entre las posibilidades está almacenamiento local o plataformas de almacenamiento de archivos compartidos (e.g. Google Drive, Dropbox, Mega, etc). Adicionalmente, se debe establecer el formato de los archivos almacenados que puede ser desde archivos planos simples hasta bases de datos complejas. Acorde con la fuente de datos utilizada se determina el formato más adecuado para el manejo de la información, entre los cuales están archivos de texto, imagen, vídeo, audio. Al igual que en la fase de datos cuantitativos, se considera que este tipo de investigación pueden usar múltiples fuentes de datos o inclusive en casos en lo que se usa solo una fuente de datos, estos pueden estar desagregados en función del tiempo, grupos intrínsecos, organizaciones, etc. Teniendo en cuenta esto, se debe realizar un proceso de consolidación de todos los datos almacenados, definiendo un formato homogéneo para todo el conjunto de datos.

3.3.3. Limpieza del conjunto de datos cualitativos

La limpieza del conjunto de datos es la tercera actividad de la etapa de preparación dentro de la fase de datos cualitativos. La actividad de limpieza tiene como objetivo tanto descartar la información no relevante del conjunto de datos cualitativos con el fin de obtener un conjunto de datos que extienda o complemente los datos cuantitativos. Teniendo en cuenta la naturaleza de los datos cualitativos, es posible encontrar problemas de calidad en el conjunto de datos; por ejemplo, participantes fal-

tantes, digitación incorrecta, respuestas incompletas, etc. En primer lugar, se debe hacer una revisión preliminar del conjunto de datos consolidado, identificando información faltante, incompleta, atípica o irrelevante para el trabajo. Al identificar los datos inconsistentes esta información solo puede ser descartada porque la información cualitativa no puede ser modificada porque existe el riesgo de cambiar la intención o interpretación del individuo estudiado. Por otra parte, dado que esta actividad hace parte de una metodología mixta, donde se integran los resultados de la fase cuantitativa y cualitativa, se debe realizar una comparación de los datos cualitativos con los cuantitativos, asegurando que sean los mismos individuos los considerados en el conjunto de datos. Finalmente, en los caso en lo que la cantidad de datos sea muy grande, se puede necesitar usar herramientas adicionales como programas computacionales, algunos ejemplos son: Atlas.ti, Decision Explorer, Ethnograph y NVivo.

3.3.4. Exploración de temas

La exploración de temas corresponde a la primera actividad dentro de la etapa de transformación de la fase de datos cualitativos. Cabe resaltar que, dentro de la metodología propuesta esta actividad da inicio al proceso de análisis de contenido, el cual está representado con el recuadro de línea amarilla en la Figura 3-1. El análisis de contenido originalmente es definido por Berelson (1952) como *“una técnica de investigación para la descripción objetiva, sistemática y cuantitativa del contenido manifiesto de la comunicación”*. Una definición más actualizada corresponde a *“una técnica sistemática y reproducible utilizada para comprimir un gran volumen de palabras de texto a una cantidad reducida de categorías de contenido, por medio de la aplicación de reglas explícitas de codificación”* (Stemler, 2000). Por otra parte, Bryman (2015) propone definir el análisis de contenido como la cuantificación sistemática y reproducible en términos de categorías predeterminadas de documentos y textos, los cuales pueden ser tanto impresos como visuales. En este sentido, el análisis de contenido se vuelve aplicable no solo para fuentes de información textuales sino también para otros medios como acciones observadas en vídeo, imágenes y discursos (Bryman, 2015; Stemler, 2000).

El análisis de contenido es un proceso no lineal e iterativo, puesto que las tareas de codificar y categorizar no son eventos únicos dentro del procedimiento (Hernández-Sampieri *et al.*, 2014). En este sentido, a veces es necesario revisar nuevamente los datos iniciales para reflexionar sobre los análisis realizados o inclusive volver a recolectar más información. Cabe resaltar, que los datos recolectados deben ser analizados sin ninguna modificación de forma (p. ej. puntuación u ortografía) o redacción, debido a que estas correcciones podrían poner en riesgo la veracidad de los resultados. Con respecto a la actividad de exploración de temas, la cual da inicio al análisis de contenido, el procedimiento es el siguiente:

1. Elegir la unidad de análisis o significado básica con base en los datos cualitativos recolectados.
2. Hacer una lectura rápida de todas la información recolectada e identificar conceptos claves que ayuden a entender el conjunto de datos, como temas recurrentes, atípicos, relacionados o similares.

3.3.5. Codificación abierta

La codificación abierta es la segunda actividad dentro de la etapa de transformación de la fase de datos cualitativos y también corresponde al segundo paso dentro del análisis de contenido. Esta codificación requiere que los investigadores sean receptivos a perspectivas y posibilidades teóricas nuevas que permitan interpretar los fenómenos reflejados en los datos recolectados, dentro de este proceso interpretativo la información es fragmentada y analizada a profundidad con el objetivo de etiquetar los conceptos, propiedades y dimensiones identificados (Corbin y Strauss, 1990). En otras palabras, durante el proceso de codificación abierta la información recolectada debe ser dividida en fragmentos puntuales, los cuales son etiquetados por medio de códigos que surgen de la interpretación de los datos. El proceso de generación de códigos debe tener en cuenta que el dato original representa el ejemplo de un fenómeno general, por lo tanto la etiqueta creada debe indicar una idea global y no específica. El uso de códigos para etiquetar la información permite comparar y contrastar objetivamente los datos e identificar eventos similares, evitando que se generen nociones preconcebidas y sesgos dentro de la investigación (Hernández-Sampieri *et al.*, 2014). El paso a paso de la codificación abierta es detallado a continuación:

1. Dividir los datos en fragmentos pequeños de información, los cuales pueden ser párrafos, secciones de texto, frases, oraciones o palabras.
2. Leer cada fragmento de información de manera reflexiva, identificando las propiedades y conceptos presentes y establecer un código adecuado para etiquetar el fenómeno ejemplificado.
3. Compilar todos los fragmentos de información que tienen asignado la misma etiqueta y realizar comparaciones entre grupos, con el objetivo de asegurar que todos los segmentos de dato relacionados con el mismo tema están etiquetados con el mismo código.

3.3.6. Codificación axial

La codificación axial es la tercera actividad dentro de la etapa de transformación de la fase de datos cualitativos y también corresponde al tercer paso dentro del análisis de contenido. Esta codificación inicia identificando conexiones entre los códigos generados en la codificación abierta, a partir de las cuales los códigos son agrupados en categorías (Corbin y Strauss, 1990). Cabe resaltar que, una categoría puede ser construida con base en un solo código o una nueva categoría puede surgir como un conjunto de diferentes códigos. Adicionalmente, en esta actividad también es posible explorar los vínculos entre categorías, como relaciones temporales o de causalidad; lo que permite identificar categorías más generales o temas centrales que generan explicaciones más precisas y completas sobre los fenómenos presentes en los datos. El resultado de esta codificación son los temas centrales y categorías generados, los cuales son considerados los ejes axiales alrededor de los cuales se enlazan conjuntos de códigos (Hernández-Sampieri *et al.*, 2014). El proceso para realizar la codificación axial de los datos es detallada a continuación:

1. A partir del conjunto de códigos establecido en el proceso de codificación abierta, se deben identificar vínculos entre las etiquetas y establecer categorías donde puedan ser agrupadas.
2. Buscar asociaciones entre las categorías generadas como relaciones de causalidad, contexto detrás de observaciones o consecuencias del fenómeno y agrupar las categorías en temas generales.

3.3.7. Relativización de los datos

La relativización de datos o codificación selectiva es la primera actividad dentro de la etapa de análisis de la fase de datos cualitativos y también corresponde al último paso dentro del análisis de contenido. La codificación selectiva es considerada el proceso donde se identifica el fenómeno o categoría central que unifica todas las demás categorías y temas resultantes de las codificaciones anteriores (Corbin y Strauss, 1990). Adicionalmente, la categoría central captura la esencia de la investigación permitiendo al investigador generar una teoría unificada, expresada en pocas palabras o frases, la cual puede ser tanto una nueva teoría como la modificación de una existente con base en los resultados obtenidos (Hernández-Sampieri *et al.*, 2014). El fenómeno central puede surgir tanto de resaltar una de las categorías o temas resultantes de la codificación axial o puede ser una nueva categoría derivada de la unificación de las demás. Esta codificación puede señalar la necesidad de refinamiento de algunos códigos, los cuales pueden resultar en la creación, mezcla, división o eliminación de etiquetas (Corbin y Strauss, 1990). El paso a paso de la relativización de los datos es el siguiente:

1. Con base en las tendencias identificadas en los datos, se define la categoría central que agrupa todos los temas y categorías de la codificación axial y que captura la idea general de la investigación.
2. Identificar los vínculos entre la categoría general y el resto de temas y categorías, con el objetivo de determinar la narrativa final del reporte de la investigación.
3. Identificar los temas, categorías y códigos que parecen no tener relación con el fenómeno central identificado y verificar si la cantidad de datos de estas etiquetas es suficiente para considerar relevantes los resultados. En el caso de que la información sea insuficiente la etiqueta debe ser eliminada.
4. Revisar nuevamente los datos originales y codificar los fragmentos de información teniendo en cuenta la categoría general generada.

3.4. Fase 3: Discusión resultados

3.4.1. Integración de fases

La integración de los resultados de las fases anteriores corresponde a la primera actividad de la fase de discusión de resultados. Teniendo en cuenta que el diseño metodológico propuesto corresponde a una metodología de métodos mixtos de carácter secuencial explicativa, esta actividad consolida los resultados de la fase de métodos cuantitativos con los de los métodos cualitativos. En este orden de ideas, los hallazgos de la primera fase intentan ser explicados por los descubrimientos de la segunda fase. Ya que ambas fases terminan con el análisis de datos, se hace una comparación de las conclusiones obtenidas en cada fase. En este sentido, esa comparación verifica si los comportamientos encontrados con los datos cuantitativos son corroborados o desmentidos por medio de los datos cualitativos. La ventaja de esta actividad es el hecho de que permite ampliar el alcance de los resultados de la fase cuantitativa, generando aclaraciones sobre los comportamientos encontrados desde un enfoque cualitativo.

3.4.2. Interpretación de resultados

La interpretación de los resultados integrados corresponde a la segunda actividad de la fase de discusión de resultados. Esta actividad consiste en retornar a las preguntas de investigación planteadas para el trabajo y estas preguntas deben ser respondidas en detalle con base en resultados específicos obtenidos. Las preguntas deben ser abordadas en el orden en el que fueron planteadas, y se debe especificar cuáles son los valores y análisis en concreto que han respuesta directa a estas. Posteriormente, los resultados obtenidos deben ser comparados con los hallazgos de los trabajos relacionados, que fueron estudiados en la revisión de literatura. La comparación de resultados puede resultar en la confirmación de hallazgos de otras investigaciones o en su contradicción. En el caso de obtener resultados diferentes a la literatura, se deben plantear las posibles razones por las cuales se generan estas diferencias, las cuales pueden estar relacionadas con el conjunto de datos, las características del curso, el contexto, entre otras.

3.4.3. Generación de hipótesis

La generación de hipótesis corresponde a la tercera actividad de la fase de discusión de resultados. Esta actividad parte de un análisis profundo sobre los resultados obtenidos, describiendo en detalle las posibles razones de los comportamientos encontrados. De esta manera, es posible plantear hipótesis que describan aspectos de procesos de aprendizaje que posiblemente se están dando en el ambiente educativo estudiado. Con base en estas hipótesis, los investigadores deben describir cuáles son las implicaciones de los resultados obtenidos en el contexto de investigación. Y finalmente, se debe reflexionar sobre cómo contribuyen los hallazgos de la investigación en la comunidad científica, específicamente, la relacionada con analítica del aprendizaje en cursos de programación. Las hipóte-

sis planteadas pueden servir como base para nuevos proyectos de investigación y ser abordadas en trabajos futuros.

3.4.4. Conclusiones

El planteamiento de conclusiones corresponde a la última actividad de la fase de discusión de resultados. Dentro de las conclusiones del trabajo se deben mencionar las contribuciones más importantes de la investigación. Esta actividad inicia con el resumen de la motivación, problemática y diseño metodológico propuesto. Posteriormente, se especifican los hallazgos más importantes tanto de la fase cuantitativa como de la cualitativa, y las interpretaciones de la integración de los resultados. Adicionalmente, es necesario contestar las preguntas de investigación planteadas y los resultados específicos que dan resolución a estas. Sumado a esto, se debe reflexionar sobre las limitaciones y amenazas a la validez encontradas en el desarrollo de la investigación, como información insuficiente, sesgos o restricciones en la aplicabilidad de los resultados. Para finalizar, es necesario plantear ideas para trabajos futuros que puedan ser desarrollados a partir de los hallazgos del presente proyecto.

4 Datos cuantitativos

Este capítulo especifica todas las actividades realizadas en la fase de datos cuantitativos. Las primeras tres secciones del capítulo especifican la etapa de la preparación de los datos. En este sentido, la Sección 4.1 detalla las tareas de la recolección de datos de tipo cuantitativo. Seguido de esto, la Sección 4.2 explica las actividades realizadas para la consolidación de los datos recolectados en un solo conjunto de datos. Posteriormente, la Sección 4.3 especifica los filtros aplicados para la depuración de los datos consolidados. Las siguientes tres secciones del capítulo corresponden a la etapa de transformación. En primer lugar, la Sección 4.4 resume las medidas del conjunto de datos que son de interés para la investigación. En segunda instancia, la Sección 4.5 describe el proceso de construcción de métricas, especificando las expresiones y ecuaciones para su cálculo. En tercer lugar, en la Sección 4.6 se especifican los procedimientos utilizados para la cuantificación de las medidas y métricas. Finalmente, las últimas dos secciones del capítulo describen las actividades de la etapa de análisis de datos. Así pues, en la Sección 4.7 se realiza un análisis univariado de cada una de las medidas identificadas y métricas calculadas. Por último, la Sección 4.8 resume los resultados del análisis de correlaciones, el cual establece como variable de especial interés el desempeño académico.

4.1. Recolección de datos cuantitativos

Teniendo en cuenta que el objetivo de la investigación es observar los comportamientos de los estudiantes sin modificar el contexto normal de desarrollo, se hace uso de un conjunto de datos históricos recolectado desde el primer periodo académico del 2019 hasta el segundo periodo del 2020 (2 años - 4 semestres académicos). La muestra poblacional corresponde a estudiantes que cursaron la asignatura *Programación de Computadores* en la Universidad Nacional de Colombia durante ese periodo de tiempo y utilizaron la plataforma UNCode en la asignatura¹.

UNCode es una plataforma desarrollada en la Universidad Nacional de Colombia que es utilizada en algunos cursos de la Facultad de Ingeniería para la evaluación automática de ejercicios de programación (Restrepo-Calle *et al.*, 2019; Restrepo-Calle *et al.*, 2020). En específico, UNCode es una plataforma educativa que permite a los estudiantes enviar múltiples intentos de solución a tareas de programación por medio de programas de computador (código fuente o Jupyter notebooks). En cada intento de solución enviado por el estudiante la plataforma almacena el archivo del programa entre-

¹UNCode está disponible en: <https://uncode.unal.edu.co/> y su documentación se puede encontrar en este enlace: <https://juezun.github.io/>

gado, la fecha y hora de entrega. La plataforma adicionalmente ofrece evaluación automática, a través de veredictos que generan retroalimentación sobre errores de sintaxis, semántica y eficiencia del programa. UNCode también ofrece calificación numérica, con base en el número de casos de prueba que el programa enviado logre satisfacer. Sumado a esto, en la plataforma UNCode se han incluido un conjunto de herramientas de apoyo para los estudiantes, tales como: resaltado de sintaxis, autocompletado de código, verificación de buenas prácticas de programación (i.e., *linter*), visualización de la ejecución del código paso a paso (por medio de Python Tutor (Guo, 2013)) y la posibilidad de realizar pruebas personalizadas (i.e., *custom input*). En la plataforma también existen herramientas de monitoreo, por medio de reportes estadísticos interactivos tanto para estudiantes como para profesores. En Restrepo-Calle *et al.* (2018) se pueden consultar más detalles acerca de las funcionalidades de UNCode.

Por otra parte, durante este mismo periodo de tiempo (desde el segundo semestre del 2019 hasta el segundo semestre del 2020), se realizaron encuestas a los estudiantes para conocer sus percepciones con respecto al uso de UNCode en la asignatura programación de computadores. Las encuestas fueron aplicadas a 17 grupos de los 22 cursos de Programación de Computadores, por medio de *Google Forms* y las respuestas fueron almacenadas en hojas de cálculo por periodo académico. Los datos recolectados por medio de las encuestas incluyen información demográfica sobre los estudiantes y las respuestas a las preguntas sobre el uso de la plataforma. Los tipos de pregunta de las encuestas corresponden a los siguientes:

- Preguntas cerradas, donde se plantea una afirmación y los estudiantes deben responder con su nivel de acuerdo o desacuerdo en escala Likert. La Tabla 4-1 especifica las categorías de la escala Likert, donde el valor 6 es el máximo indicador de acuerdo y 1 es el mayor nivel de desacuerdo.

Tabla 4-1: Categorías de la escala Likert

Valor numérico	Categoría
6	Totalmente de acuerdo.
5	De acuerdo.
4	Algo de acuerdo.
3	Algo en desacuerdo.
2	En desacuerdo.
1	Totalmente en desacuerdo.

- Preguntas abiertas, donde los estudiantes responden las razones por las cuales están de acuerdo o en desacuerdo con las afirmaciones de las preguntas cerradas.

En esta fase se tienen en cuenta solo las respuestas de las preguntas cerradas de la encuesta, puesto que los datos recolectados son de tipo cuantitativo.

De esta manera, las fuentes de datos consideradas para el presente estudio son las siguientes:

1. Registro de las interacciones de los estudiantes con la plataforma UNCode utilizada en los cursos de Programación de Computadores en la Universidad Nacional de Colombia - sede Bogotá,
2. Encuestas de percepción de los estudiantes acerca del uso de la plataforma educativa.

Los datos de las interacciones de los alumnos son de tipo dinámico, dado que son registradas durante el desarrollo del curso. En contraste, las respuestas de las encuestas de percepción son datos estáticos, puesto que las encuestas son aplicadas posterior a la finalización del curso.

Una vez identificadas las fuentes de datos, se procedió a definir la ubicación y formato de almacenamiento de los datos recolectados. Teniendo en cuenta que la información recolectada es de alta magnitud, esta fase de la investigación se realiza con la ayuda de otros integrantes del grupo de investigación PLaS² con conocimientos en manejo y procesamiento de datos. En este sentido, se tomó la decisión de almacenar la información en una unidad compartida de Google Drive, dado que permite el manejo de archivos de manera colaborativa. Adicionalmente para el procesamiento y posterior análisis de la información se utilizaron Jupyter Notebooks en Google Collaboratory, dado que estas herramientas usan los recursos computacionales de Google lo que facilita el proceso de análisis de datos. Adicionalmente, para el formato de almacenamiento en la unidad compartida se decidió trabajar con archivos de texto con valores separados por comas con extensión *.csv* y hojas de cálculo con extensión *.xlsx*, puesto que este tipo de archivos pueden ser fácilmente leídos y accedidos desde Google Collaboratory.

Los datos sobre las interacciones con UNCode se encuentran almacenados en una base de datos MongoDB. El conjunto de datos contiene 16 colecciones de datos, la Figura 4-1 especifica los datos que posee cada una de ellas. Las colecciones *aggregations* y *users* contienen datos sobre los cursos y los usuarios registrados en la plataforma, respectivamente. La información sobre el registro de ingresos a las herramientas de UNCode se encuentra almacenada en las colecciones *analytics* y *services*. Las colecciones *batch_jobs*, *lis_outcome_queue*, *nonce* y *sessions* contienen información específica para la funcionalidad y estructura de la base de datos. La colección *problem_banks* corresponde al banco de preguntas con los ejercicios de programación propuestos en cada curso. Las especificaciones de las actividades realizadas en cada curso se encuentran en *task_cache* y *user_tasks*. Las calificaciones finales de cada estudiante en la asignatura corresponden a *students_grades*. Entre estas colecciones se destaca *submissions*, donde están todos los envíos hecho por los estudiantes como intentos de solución a las tareas de programación propuestas en la plataforma. Finalmente, las colecciones *fs.chunks* y *fs.files* poseen información específica sobre cada uno de los archivos enviados por los estudiantes en cada intento de solución.

²<http://plas.unal.edu.co/>

aggregations	analytics	batch_jobs	fs.chunks	fs.files	lis_outcome_queue	nonce	problem_banks
_id	_id	_id	_id	_id	_mongo	_id	_id
tutors	date	group_name	n	uploadDate	_db	expiration	courseid
description	username	group_hash	data	chunkSize	_shortName	timestamp	course_name
students	service	courseid	files_id	length	fullName	nonce	
courseid	session_id	container_name		md5			
default	course_id	submitted_on					
groups		result					

student_grades	services	sessions	submissions	tasks_cache	user_tasks	users
_id	_id	_id	_id	_id	_id	_id
uncode_grade	service_name	data	taskid	course_id	courseid	username
final_course_grade	service_key	atime	username	task_id	taskid	realname
username			input	tags	username	language
courseid			submitted_on	task_name	grade	password
			courseid	task_context	tried	email
			status	task_author	succeeded	bindings
			response_type	course_name	submissionid	
			stderr		tokens	
			archive			
			custom			
			result			
			stdout			
			tests			
			text			
			grade			
			problems			

Figura 4-1: Colecciones dentro de la base de datos MongoDB de UNCode

En primer lugar, se genera una copia actualizada de la base de datos de UNCode, con el objetivo de proteger los datos originales. La copia de la base de datos es exportada como un archivo comprimido, el cual contiene las 16 colecciones descritas en la Figura 4-1. El notebook 1) Dataset builder se utiliza con el objetivo de generar el conjunto de datos organizados como sistema de archivos *csv*, a partir de las 16 colecciones de la copia de la base de datos. La ejecución del notebook debe ser realizada localmente, puesto que la memoria disponible en Google Collaboratory es menor al tamaño del archivo de la copia de la base de datos (17 GB aprox.). Dentro del notebook se utiliza la librería *Pymongo* para hacer la conexión y recopilación del sistema de archivos de la base de datos. También se utilizan *Gridfs* y *Bson* para el procesamiento y lectura de archivos codificados. Cabe resaltar que, no toda la información contenida en las 16 colecciones de la copia de la base de datos es considerada relevante para la investigación, por lo cual deben realizarse un proceso de selección de datos. La Figura 4-2 especifica las colecciones y los datos que según el criterio de los investigadores aportan conocimiento sobre el proceso de aprendizaje de los estudiantes. Las colecciones seleccionadas junto con la información utilizada en este trabajo, son listadas a continuación:

- Colección *aggregations*: Contiene los nombres de usuario e identificador del curso al que pertenece cada estudiante.
- Colección *analytics*: Consolida la información sobre el uso de las herramientas disponibles en la plataforma, las cuales ayudan a los estudiantes en el desarrollo de actividades.

aggregations	analytics	student_grades	submissions	tasks_cache	user_tasks	users
_id	_id	_id	_id	_id	_id	_id
tutors	date	unicode_grade	taskid	course_id	courseid	username
description	username	final_course_grade	username	task_id	taskid	realname
students	service	username	input	tags	username	language
courseid	session_id	courseid	submitted_on	task_name	grade	password
default	course_id		courseid	task_context	tried	email
groups			status	task_author	succeeded	bindings
			response_type	course_name	submissionid	
			stderr		tokens	
			archive			
			custom			
			result			
			stdout			
			tests			
			text			
			grade			
			problems			

Figura 4-2: Tabla con las colecciones seleccionadas de la base de datos de UNCode

- Colección *students_grades*: Contiene las calificaciones de los estudiantes, la cuales corresponden al promedio ponderado de las actividades realizadas en la plataforma.
- Colección *submissions*: Contiene toda la información relacionada con los intentos de solución construidos y enviados por los estudiantes para la resolución de problemas de programación planteados en el curso. Los datos que contiene esta colección son: la fecha y hora de entrega, identificador de actividad y curso, archivo cargado en la plataforma, pruebas realizadas, veredictos obtenidos y calificación numérica.
- Colección *task_cache*: Posee la descripción, enunciado e identificador de cada actividad planteada por el docente en el curso.
- Colección *user_tasks*: Resume la cantidad de intentos de solución enviados por estudiante y calificación final obtenida en cada actividad.
- Colección *users*: Corresponde a la lista de usuarios registrados en la plataforma identificados por nombre de usuario y correo electrónico.

Cada una de estas colecciones son almacenadas en un formato que pueda ser posteriormente manipulado y utilizado para análisis. En este sentido, el notebook se utiliza la librería *pandas* para transformar cada colección en un DataFrame. La estructura de datos DataFrame es utilizada, puesto que facilita

el análisis y visualización de datos de alto volumen. Posteriormente, cada uno de los DataFrames generados es transformado en archivos con extensión **.csv**. Teniendo en cuenta que, la ejecución del notebook es realizada localmente, el conjunto de datos es generado dentro del almacenamiento interno del servidor local utilizado para la ejecución. En este sentido, los archivos del conjunto de datos debe ser cargados manualmente a la unidad compartida de Google Drive Learning Analytics 1/Recolección/UNCode.

Por otra parte, las encuestas de percepción son realizadas por medio de Google Forms y las respuestas son exportadas como archivos de Excel (**.xlsx**). Cabe resaltar, que las encuestas son realizadas al final de cada semestre académico desde el 2019-1, en algunos cursos donde UNCode es utilizado. En este sentido, se obtienen cinco archivos con las respuestas de cada semestre, los cuales son almacenados en una unidad compartida Learning Analytics 1/Recolección/Encuestas.

4.2. Consolidación del conjunto de datos cuantitativos

En esta actividad se utiliza el notebook UNCode - Data consolidation, el cual tiene como objetivo consolidar el conjunto de datos definitivo que se va a utilizar en la investigación. Teniendo en cuenta que la información recolectada incluye todos los cursos que utilizan UNCode (43), es necesario delimitar los datos exclusivamente a cursos de programación de computadores, los cuales corresponden a 24. En este sentido, los archivos almacenados en la unidad compartida del proceso de recolección (Learning Analytics 1/Recolección/UNCode.) son copiados en una nueva unidad compartida, excluyendo los directorios correspondientes a cursos diferentes a la asignatura *Programación de Computadores*. Dentro del notebook se utiliza el módulo `os.path` de Python, el cual permite acceder a los archivos y directorios de las unidades compartidas por medio de rutas de acceso. Adicionalmente, se emplea el módulo `pathlib` para la generación de directorios nuevos en la unidad compartida de destino. Los directorios y archivos copiados son almacenados dentro de la unidad compartida en la carpeta denominada Learning Analytics 2/Consolidado/UNCode. La información está organizada en carpetas individuales, las cuales corresponden a cada curso de programación de computadores donde se utiliza UNCode, dentro de las cuales se almacenan los siguientes archivos:

1. Students: Listado de los usuarios identificados con el rol de estudiante que pertenecen al curso.
2. Analytics: información relacionada con el uso de las herramientas de UNCode por parte de los usuarios. Las herramientas disponibles para apoyar a los estudiantes en la solución de las actividades del curso son las cinco siguientes:
 - Custom input: que es una herramienta que permite a los estudiantes construir sus propias pruebas personalizadas para evaluar los programas construidos.
 - Python tutor: es una herramienta que permite visualizar paso a paso el flujo de ejecución del programa enviado.

- **Multiple languages code:** le da la opción a los estudiantes de someter a evaluación código fuente escrito en diferentes lenguajes de programación (C/C++, Java, Python).
- **Linter:** es una herramienta de resaltado que identifica errores y recomendaciones en el código fuente con base en principios de buenas prácticas de programación.
- **User statistics:** corresponde a reportes estadísticos sobre las calificaciones obtenidas por cada estudiante. Permite a los estudiantes conocer su progreso en la asignatura.

Por otra parte, la columna correspondiente a la fecha en la que se registra el acceso es desglosada en las columnas: día (*day*), mes (*month*), año (*year*), día de la semana (*day_of_week*) y hora (*time*). Adicionalmente, el archivo contiene el nombre de usuario del estudiante (*username*), el nombre de la herramienta utilizada (*service*) y el identificador del curso (*course_id*).

3. **Unicode_grades:** Archivo con las calificaciones finales asignadas a cada estudiante. Este archivo está conformado principalmente por la nota final (*unicode_grade*), la cual corresponde al promedio ponderado de las notas obtenidas de las actividades realizadas dentro de la plataforma. En algunos casos, el archivo contiene una columna llamada (*final_course_grade*) que corresponde a las calificaciones finales reportadas por los profesores, las cuales tienen en cuenta tanto las actividades realizadas en UNCode como las desarrolladas fuera de la plataforma. Estos mismos cursos tienen una columna llamada (*average_grade*) que corresponde al promedio entre *unicode_grade* y *final_course_grade*. Sumado a esto, están las columnas con los nombres de usuario (*username*) y el identificador del curso (*course_id*).
4. **Submissions:** registros de los intentos de solución enviadas por los estudiantes en las actividades del curso. Este archivo está compuesto por la fecha y hora en la que fue enviada la solución (*submitted_on*), la cual es dividida en día (*day*), mes (*month*), año (*year*), día de la semana (*day_of_week*) y hora (*time*). Este archivo tiene los registros de cada intento de solución, donde se especifica el identificador de la actividad (*taskid*), el estudiante que realiza el envío (*username*) y el curso (*course_id*). Finalmente, el archivo contiene el identificador del archivo enviado (*input*), la calificación numérica (*grade*) en un escala de 0 a 100, las pruebas realizadas (*tests*) y el veredicto obtenido (*custom_summary_result*).
5. **Input:** Este archivo especifica las características de los archivos enviados por cada estudiante en cada intento de solución. Las columnas corresponden al identificador del archivo (*index*), el nombre del archivo cargado en la plataforma (*file_name*) y el lenguaje de programación utilizado (*language*).
6. **Task_cache:** Contiene información específica sobre las actividades del curso que son desarrolladas dentro de la plataforma UNCode. Cada archivo tiene una columna con el identificador del curso (*course_id*), el nombre del curso (*course_name*), la descripción del problema planteado (*task_context*), el identificador y el nombre de la actividad (*task_id* y *task_name*, respectivamente).

7. *User_tasks*: Este archivo resume la cantidad de intentos realizados por cada estudiante en las actividades del curso. Las columnas de esta tabla están compuestas por el identificador del curso (*courseid*), la calificación obtenida en el mejor intento realizado (*grade*), el identificador del envío realizado (*submissionid*), el identificador de la actividad (*taskid*), la cantidad total de intentos realizados (*tried*) y el nombre de usuario del estudiante (*username*).

Adicionalmente, cada una de estas carpetas por curso contiene un directorio llamado *submissions*, el cual contiene todos los archivos enviados por los estudiantes en cada intento de solución. Estos archivos están organizados en directorios identificados con los nombres de usuario de los estudiantes, los cuales contienen subdirectorios que representan cada actividad del curso.

Por otra parte, las respuestas de las encuestas de percepción son consolidadas en un único archivo *csv*, el cual reúne la información de todos los cursos donde es realizada la encuesta. En este sentido, en el notebook *Survey - Data consolidation* cada archivo de Excel es transformado en *DataFrames*, los cuales son concatenados formando un *DataFrame* con toda la información. El archivo final contiene las columnas con el nombre de usuario del estudiante (*username*), la fecha de nacimiento, sexo, programa académico y el grupo del curso (*course_id*). Adicionalmente, se seleccionan las respuestas a las preguntas cerradas de las encuestas de satisfacción. Las respuestas a estas preguntas están basadas en la escala Likert, la cual tiene siete categorías que expresan el nivel de acuerdo o desacuerdo del encuestado con una declaración. Finalmente, el *Dataframe* consolidado es importado como un archivo *csv* y almacenado en la carpeta *Learning Analytics 2/Consolidado/Encuestas*.

4.3. Limpieza del conjunto de datos cuantitativos

Esta actividad corresponde a una depuración de los datos que presentan fallas o están incompletos, la cual se realiza por medio del notebook *UNCode - Data selection*. En primer lugar, se hace un filtro de los cursos de programación de computadores, seleccionando solo los que realizan una alta cantidad de actividades en la plataforma. El número de actividades realizadas en cada curso es cuantificado a partir del archivo *task_cache.csv*. La Figura 4-3 resume la cantidad total de actividades realizadas en UNCode por cada grupo. Acorde con la gráfica la cantidad máxima de actividades es de 102 realizadas en los grupos *pc-group15-2019-2* y *pc-group16-2019-2*. Por otra parte, el grupo con la menor cantidad de actividades realizadas es el grupo *pc-group8-2020-1* con 15 actividades. Los resultados evidencian que todos los cursos tienen suficiente información registrada sobre la interacción de los estudiantes con la plataforma. Sin embargo, los cursos *group-5* y *group-6* que corresponden al semestre académico 2018-II son descartados, debido a que estos grupos son cursos piloto en el uso de UNCode. En este sentido, estos dos grupos son grupos donde la plataforma fue utilizada para el estudio preliminar de su funcionalidad y diseño, lo que hace que la información no sea comparable con los datos recolectados en los demás cursos. En conclusión, 22 grupos de los 24 iniciales son seleccionados, los cuales comprenden un total de 1404 actividades. El promedio de actividades por curso es de 63.8 con una

desviación estándar de 27.7.

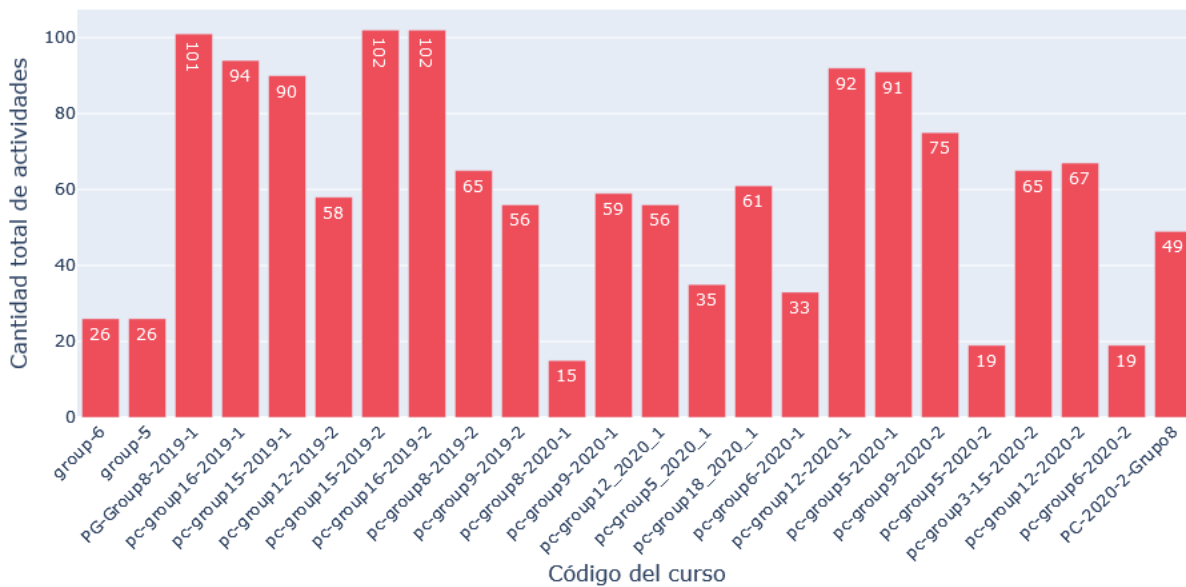


Figura 4-3: Cantidad total de actividades realizadas en UNCode en cada curso

En segundo lugar, se hace un filtro de las actividades con base en la cantidad de envíos totales registrados en la plataforma. El total de envíos por actividad es cuantificado a partir del archivo *submissions.csv*. Antes de realizar el conteo de envíos por actividad, se eliminan del archivo *submissions.csv* los usuarios que tienen rol de administradores, es decir, profesores y asistentes docentes. En este sentido, se utiliza la lista de estudiantes del curso contenida en el archivo *students.csv*. Seguido a esto, se crea un DataFrame con las siguientes columnas: código del curso (*course_id*), nombre de la actividad (*task_name*), código de la actividad (*task_id*) y la cantidad total de soluciones enviadas. La Figura 4-4 resume la cantidad total de envíos hechos por actividad en cada grupo. En la gráfica se pueden identificar actividades con baja o nula cantidad de envíos, las cuales corresponden a actividades creadas como pruebas del funcionamiento de sistema o como actividades opcionales en el curso; en este sentido, los datos de estas actividades deben ser descartados, puesto que no son relevantes para el entendimiento de los procesos de aprendizaje de los estudiantes. Para aplicar este filtro se establece un límite mínimo de 15 envíos por actividad, dado que este valor corresponde a la mínima cantidad de actividades por curso (Figura 4-3). En este sentido, se identifican 49 actividades de 1404 que no poseen suficiente número de envíos, es decir, menos de 15 entregas y por esta razón son eliminadas del archivo de datos.

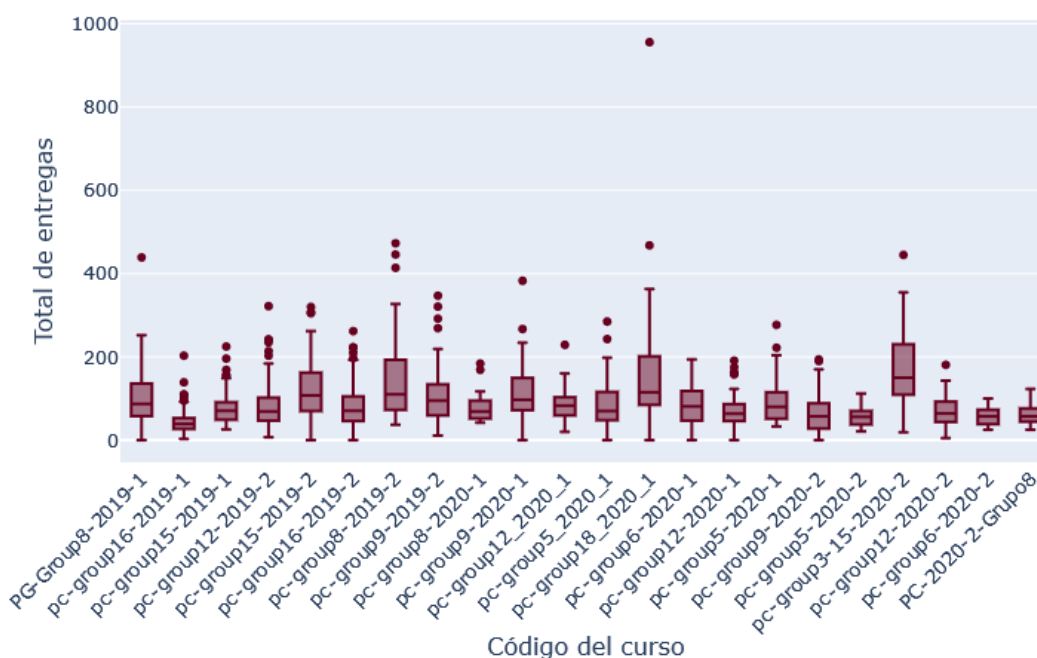


Figura 4-4: Cantidad total de soluciones enviadas en cada actividad realizada en UNCode

En la Figura 4-4 destaca una de las actividades del curso *pc-group18_2020_1*, la cual posee cerca de 1000 envíos registrados. Este dato es relevante porque la siguiente actividad con más envíos (curso *pc-group8-2019-2*) tiene 473, la cual es menos de la mitad de las soluciones registradas en la actividad del grupo *pc-group18_2020_1*. Este valor atípico está relacionado con una actividad especial donde los archivos enviados son de tipo notebook, que requirió más esfuerzo por parte de los estudiantes, y por consiguiente, una mayor cantidad de intentos. Teniendo en cuenta esto, y que la gran mayoría de las demás actividades son de *multiple languages* y no de notebooks, se toma la decisión de descartar las actividades donde los archivos son de tipo notebook, puesto que no son comparables con los archivos de código fuente enviados en otras actividades. Estos archivos son identificados utilizando la columna *type* de *input.csv*, la cual especifica la extensión del archivo enviado por el estudiante. En total se identifican tres actividades con este tipo de archivos, de las cuales todas se encuentran en el curso *pc-group18_2020_1*. La Figura 4-5 resume la cantidad de envíos totales de las actividades que cumplen los filtros descritos anteriormente. La cantidad final de actividades consideradas es 1352 de las 1404 iniciales, las cuales tienen un promedio de 96.2 envíos, con una desviación estándar de 64.4.

Posteriormente, se hace un filtro de estudiantes con base en la cantidad total de envíos registrados en la plataforma por estudiante. La cuantificación de envíos se hace utilizando el archivo *submissions.csv*, a partir del cual se hace una agrupación de los intentos de solución por estudiante. Así pues, se construye un DataFrame con las siguientes columnas: código del curso (*course_id*), nombre de usuario (*username*) y la cantidad total de soluciones enviadas. Los datos del DataFrame evidencian que algunos

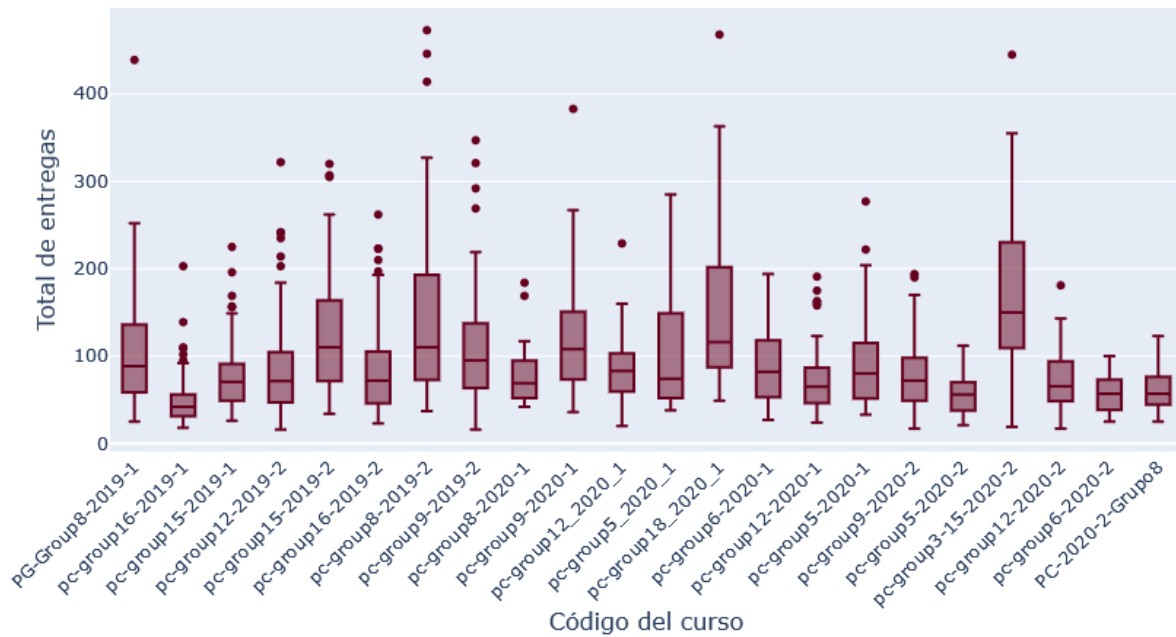


Figura 4-5: Cantidad total de soluciones enviadas en las actividades seleccionadas

usuarios tienen pocos o ningún envío registrado, lo que indica la existencia de estudiantes que posiblemente retiraron la asignatura y por esta razón tienen poca actividad en la plataforma. De manera similar al proceso de selección de actividades, en este filtro se emplea un límite mínimo de 15 envíos por estudiante. En total son identificados 37 estudiantes con menos del límite mínimo de envíos y por esta razón son eliminados del conjunto de datos. La Figura 4-6 visualiza el número de soluciones totales enviadas por los estudiantes que cumplen el criterio de envíos mínimos. El conjunto de datos final considera 735 estudiantes con un promedio de 176.6 envíos y una desviación estándar de 120.8.

El conjunto de datos con los directorios de los 22 cursos seleccionados, junto con la información depurada de las actividades y estudiantes, es almacenado dentro de la unidad compartida en la carpeta Learning Analytics 2/ Filtrado/UNCode. Adicionalmente, en esta carpeta se almacenan la lista de los cursos seleccionados en el archivo *Gropus.csv* y la lista de actividades y estudiantes en los archivos *Tasks.csv* y *Students.csv*, respectivamente. Por otra parte, se utiliza el notebook Demographic filtering con el objetivo de depurar la información de los datos demográficos, la cual está almacenada en el archivo consolidado de las encuestas de percepción llamado *Surveys.csv*. Teniendo en cuenta que las encuestas donde se recolecta la información demográfica fueron realizadas en 17 de los 22 cursos, no es necesario realizar un filtro en estos. No obstante, se requiere filtrar los estudiantes con insuficiente información para ser considerada en el conjunto de datos, lo cual se hace con base en el archivo *Students.csv*. En este filtro son eliminados 33 estudiantes, generando un conjunto final de 349 estudiantes. El conjunto final es transformado en el archivo *datos demográficos.csv*, que es almacenado en

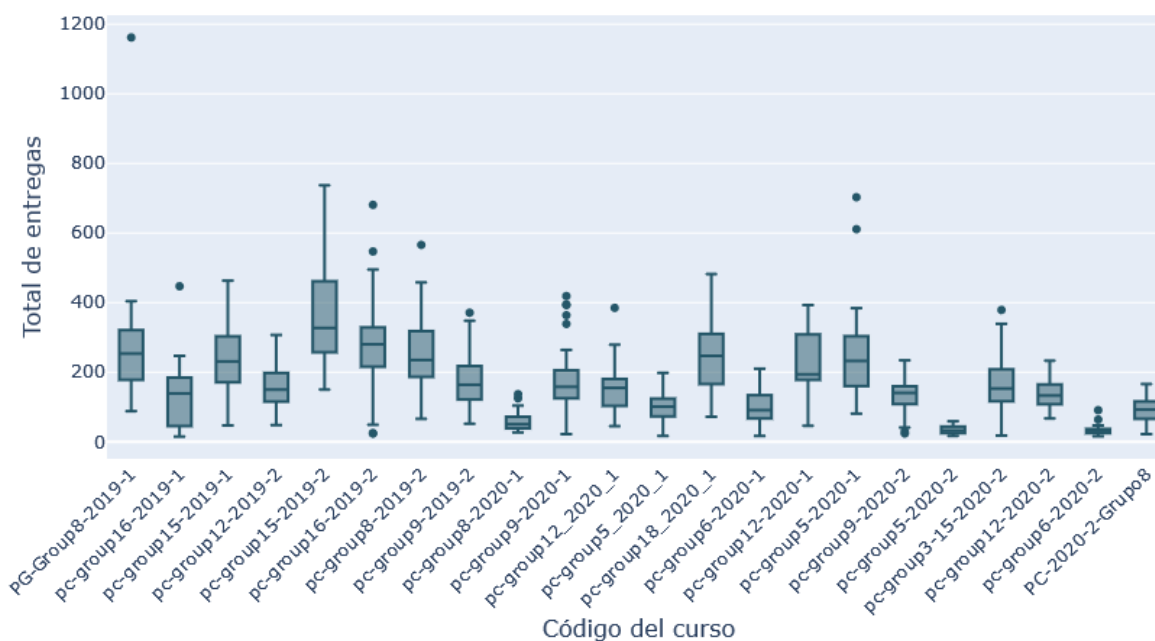


Figura 4-6: Cantidad total de envíos por estudiante

la carpeta Learning Analytics 2/3) Filtrado/Encuestas.

Sumado a esto, se utiliza el notebook Likert filtering para depurar los datos de las preguntas cerradas de la encuesta de percepción. La encuesta realizada está compuesta por diez preguntas cerradas, las cuales están relacionadas con la utilidad de la plataforma en el aprendizaje de los estudiantes. Cabe resaltar, que tres de las preguntas son descartadas, puesto que poseen 84 respuestas las cuales representan menos del 25 % de estudiantes del archivo de datos demográficos. Dado que las respuestas están basadas en las categorías de la escala Likert, los datos son transformados en los valores numéricos correspondientes a cada categoría que están especificados en la Tabla 4-1. Posteriormente, se genera el archivo *encuestas.csv* que contiene columnas con el código del curso (*course_id*), nombre de usuario de los estudiantes (*username*) y siete columnas con las preguntas cerradas de la encuesta. Los cursos y estudiantes son depurados utilizando los archivos *Gropus.csv* y *Students.csv*. Y por último, el archivo *encuestas.csv* es almacenado dentro de la unidad compartida de Google Drive en la carpeta Learning Analytics 2/3) Filtrado/Encuestas.

4.4. Identificación de variables de interés

En esta sección se determina cuáles son las variables específicas del conjunto de datos filtrado que generan conocimiento sobre el proceso de aprendizaje de los estudiantes. La Tabla 4-2 especifica 15

medidas derivadas de las interacciones registradas en UNCode que son de interés para esta investigación. Estas medidas se clasifican en cuatro categorías: 1) los datos relacionados con los intentos de soluciones realizados por los estudiantes en cada actividad, 2) veredictos obtenidos en los intentos de solución enviados, 3) información sobre el uso de las herramientas de la plataforma y 4) las calificaciones numéricas. Adicionalmente, se tienen en cuenta los archivos enviados como solución a las tareas de programación, a partir de los cuales se obtienen características de los códigos diseñados denominados *métricas de software*.

Por otra parte, la Tabla 4-3 describe 13 medidas identificadas en el conjunto de datos generado de las encuestas de percepción sobre el uso de UNCode. Estas medidas se dividen en dos categorías: datos demográficos y preguntas cerradas. Los datos demográficos corresponden a la información que permite la caracterización de la muestra de estudiantes consideradas en el conjunto de datos. Adicionalmente, la categoría de preguntas cerradas corresponde a las respuestas de los estudiantes sobre la utilidad de la plataforma en el proceso de aprendizaje en escala Likert.

4.5. Diseño de métricas de interés

En esta actividad se diseñan métricas derivadas de las medidas identificadas en la Sección 4.4, con el objetivo de obtener un mayor entendimiento sobre el proceso de aprendizaje. En total se diseñan 25 métricas que están divididas en las siguientes cuatro categorías: veredictos, uso de herramientas, métricas de software y datos demográficos. En la primera categoría están las tasas de veredictos para cada uno de los siete tipos de veredictos especificados en la Tabla 4-2. Estas tasas relacionan la cantidad de veredictos de un tipo específico con respecto al total de veredictos obtenidos por estudiante (Tabla 4-4).

La segunda categoría son las tasas de uso de herramientas de la plataforma, las cuales son definidas para cada una de las 5 herramientas presentes en la Tabla 4-2. Estas tasas de uso se definen como el porcentaje de cantidad de accesos registrados a una herramienta específica en relación con el total de accesos registrados por estudiante (Tabla 4-5).

La tercera categoría corresponde a las métricas de software, las cuales representan características específicas de los programas construidos por los estudiantes. La Tabla 4-6 describe las métricas y las ecuaciones para su cálculo. Estas métricas son definidas a partir de la cantidad de operandos, operadores, líneas de código ejecutables y número de palabras reservadas utilizadas en los códigos construidos como solución a las actividades del curso.

La última categoría corresponde a los datos demográficos, la cual tiene una métrica que corresponde a la edad de los estudiantes. Las edades son calculadas con base en la fecha de nacimiento registrada en las encuestas. Para el cálculo de las edades se emplean dos ecuaciones. Cuando el mes de nacimiento

Tabla 4-2: Medidas de la interacción con UNCode consideradas en el conjunto de datos

Categoría	Medidas	Descripción
Intentos realizados	<i>Total_Submissions</i>	Cantidad de intentos enviados por estudiante.
	<i>Duration_of_Submission</i>	Tiempo promedio empleado por los estudiantes entre intentos de envíos.
Veredictos obtenidos	Respuesta correcta (<i>Accepted</i>)	Cantidad de soluciones con respuestas correctas.
	Respuesta incorrecta (<i>Wrong_Answer</i>)	Cantidad de soluciones con respuestas incorrectas.
	Error de compilación (<i>Compilation_Error</i>)	Cantidad de intentos enviados que fallan al intentar compilar.
	Error de ejecución (<i>Runtime_Error</i>)	Cantidad de intentos que logran compilar pero fallan durante la ejecución.
	Límite de tiempo excedido (<i>Time_Limit_Exceeded</i>)	Cantidad de intentos que toman demasiado tiempo en ejecución.
	Límite de memoria excedido (<i>Memory_Limit_Exceeded</i>)	Cantidad de intentos que exceden la memoria disponible para su ejecución.
	Límite de resultado excedido (<i>Output_Limit_Exceeded</i>)	Cantidad de intentos que exceden el tamaño esperado en la salida del programa.
Uso de herramientas	<i>Python_Tutor</i>	Cantidad de accesos registrados a la herramienta Python tutor, la cual permite la visualización de la ejecución paso a paso de su programa.
	<i>Custom_Input</i>	Cantidad de accesos registrados a la herramienta Custom input donde los estudiantes realizan pruebas personalizadas a su programa.
	<i>Linter</i>	Cantidad de accesos registrados a la herramienta Linter, la cual resalta problemas sintácticos y de estilo en el código fuente.
	Estadísticas de usuario (<i>User_Statistics</i>)	Cantidad de accesos registrados al tablero interactivo de reporte individual de estadísticas de envíos y calificaciones del estudiante.
	<i>Multiple_Languages_Code</i>	Cantidad de accesos registrados a la herramienta Multiple Languages Code, la cual permite seleccionar diferentes lenguajes de programación y someter a evaluación una posible solución.
Desempeño académico	<i>uncode_grade</i>	Promedio ponderado de calificaciones de las actividades realizadas por estudiante en UNCode.

es menor al mes cuando se realizó la encuesta, la edad es calculada como la diferencia entre el año de aplicación de la encuesta y el año de nacimiento. Cuando el mes de nacimiento es mayor al mes de la realización de la encuesta, la fórmula para el cálculo de la edad es la diferencia entre años menos uno. Por último, si los meses son iguales se comparan los días. En el caso de que el día de nacimiento

Tabla 4-3: Medidas identificadas de las respuestas de encuestas de percepción de la plataforma UNCode

Categoría	Medidas	Descripción
Demográficos	Fecha de nacimiento	Día, mes y año de nacimiento del estudiante.
	Sexo	Variable que representa el sexo del alumno, tiene como opciones: Hombre o Mujer.
	Programa académico	Corresponde a la carrera universitaria del estudiante, entre las opciones hay 15 programas académicos, principalmente de ingeniería.
Preguntas cerradas	PREGUNTA: Proceso de aprendizaje	Nivel de acuerdo o desacuerdo en escala Likert del estudiante con la sentencia: <i>“UNCode fue útil en su proceso de aprendizaje de programación de computadores”</i> .
	PREGUNTA: Calificación automática	Nivel de acuerdo o desacuerdo en escala Likert del estudiante con la sentencia: <i>“UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura”</i> .
	PREGUNTA: Retroalimentación	Nivel de acuerdo o desacuerdo en escala Likert del estudiante con la sentencia: <i>“La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas”</i> .
	A_Custom_input	Nivel de acuerdo o desacuerdo en escala Likert del estudiante con la sentencia: <i>“La prueba de programas usando entradas provistas por el usuario (custom input) fue útil para su aprendizaje en esta asignatura”</i> .
	A_Linter	Nivel de acuerdo o desacuerdo en escala Likert del estudiante con la sentencia: <i>“La herramienta de verificación de buenas prácticas de programación (Linter) fue útil para su aprendizaje en esta asignatura”</i> .
	A_PythonTutor	Nivel de acuerdo o desacuerdo en escala Likert del estudiante con la sentencia: <i>“La visualización de la ejecución del programa (PythonTutor) fue útil para su aprendizaje en esta asignatura”</i> .
	A_Statistics	Nivel de acuerdo o desacuerdo en escala Likert del estudiante con la sentencia: <i>“Los reportes de desempeño (estadísticas) fueron útiles para su aprendizaje en esta asignatura”</i> .

sea menor al día de la realización de la encuesta la edad es la diferencia entre años. En contraste, si el día de nacimiento es mayor o igual, la edad se calcula como la diferencia entre años menos uno.

4.6. Cálculo de métricas diseñadas

En este paso se construye un DataFrame para cada una de las categorías de medidas y métricas detalladas en las secciones 4.4 y 4.5. La primera categoría son las medidas relacionadas con los intentos

Tabla 4-4: Métricas construidas con base en las medidas de veredictos obtenidos

Categoría	Métrica	Ecuación
Veredictos	Tasa de respuestas correctas (<i>Success_rate</i>)	$\frac{Accepted}{\sum_i Veredictos_i} \cdot 100$
	Tasa de respuestas incorrectas (<i>Error_rate_Wrong_Answer</i>)	$\frac{Wrong_Answer}{\sum_i Veredictos_i} \cdot 100$
	Tasa de errores de compilación (<i>Error_rate_Compilation_Error</i>)	$\frac{Compilation_Error}{\sum_i Veredictos_i} \cdot 100$
	Tasa de errores de ejecución (<i>Error_rate_Runtime_Error</i>)	$\frac{Runtime_Error}{\sum_i Veredictos_i} \cdot 100$
	Tasa de límite de tiempo excedido (<i>Error_rate_Time_Limit_Exceeded</i>)	$\frac{Time_Limit_Exceeded}{\sum_i Veredictos_i} \cdot 100$
	Tasa de límite de memoria excedido (<i>Error_rate_Memory_Limit_Exceeded</i>)	$\frac{Memory_Limit_Exceeded}{\sum_i Veredictos_i} \cdot 100$
	Tasa de límite de resultado excedido (<i>Error_rate_Output_Limit_Exceeded</i>)	$\frac{Output_Limit_Exceeded}{\sum_i Veredictos_i} \cdot 100$

realizados, las cuales están especificadas en la Tabla 4-2. En primer lugar, se cuantifican los intentos totales realizados por cada estudiante, por medio del notebook 4.1 Total de entregas. En este notebook se utiliza el archivo *submissions.csv* de cada curso, con el objetivo de crear un único archivo con la cantidad total de entregas realizadas por estudiante de todos los grupos. El archivo final está compuesto por las columnas: código del curso (*course_id*), nombre de usuario del estudiante (*username*) y total de entregas enviadas (*Total_Submissions*).

En segundo lugar, el tiempo promedio entre entregas (notebook 4.2 Tiempo de entrega), utiliza para el cálculo también el archivo *submissions.csv* de cada curso. En este sentido, con base en la columna de fecha y hora de entrega (*submitted_on*) se calcula la diferencia de tiempo entre entregas realizadas. Posteriormente, se cuantifica el promedio por alumno de las diferencias de tiempos entre intentos de solución. El DataFrame final está compuesto por una columna con el código del curso (*course_id*), el nombre de usuario del estudiante *username* y el promedio de tiempo en minutos (*Du-*

Tabla 4-5: Métricas construidas con base en las medidas de uso de herramientas

Categoría	Métrica	Ecuación
Uso de herramientas	Tasa de uso Python Tutor (<i>Python_Tutor_rate</i>)	$\frac{Python_Tutor}{\sum_i Herramientas_i} \cdot 100$
	Tasa de uso Custom input (<i>Custom_input_rate</i>)	$\frac{Custom_input}{\sum_i Herramientas_i} \cdot 100$
	Tasa de uso Linter (Linter_rate)	$\frac{Linter}{\sum_i Herramientas_i} \cdot 100$
	Tasa de uso estadísticas de usuario (<i>User_Statistics_rate</i>)	$\frac{User_Statistics}{\sum_i Herramientas_i} \cdot 100$
	Tasa de uso Multiple Languages Codes (<i>Multiple_Languages_Codes_rate</i>)	$\frac{Multiple_Languages_Code}{\sum_i Herramientas_i} \cdot 100$

ration_of_Submission). Sumado a esto, se agregan tres columnas con el promedio de tiempo en unidades de segundos (*Duration_of_Submission [seg]*), horas (*Duration_of_Submission [hour]*) y días (*Duration_of_Submission [days]*).

La segunda categoría son las medidas y métricas relacionadas con los veredictos obtenidos por los estudiantes, los cuales son cuantificados dentro del notebook 4.3 Tasas de veredictos. En este notebook se calculan los siete veredictos especificados en la Tabla 4-2 y las siete tasas de obtención aplicando las ecuaciones de la Tabla 4-4. Los cálculos se basan en la columna de veredictos obtenidos (*custom_summary_result*) del archivo *submissions.csv* de cada curso. En el notebook se genera un DataFrame con las siguientes columnas: código del curso (*course_id*), nombre de usuario del estudiante (*username*) y 14 columnas con la cantidad total y tasas de obtención de veredictos.

La tercera categoría corresponde a las medidas y métricas relacionadas con el uso de herramientas. Para esta categoría se calcula la cantidad de accesos a las cinco herramientas especificadas en la Tabla 4-2 y las cinco tasas de uso, las cuales son cuantificadas por medio de las ecuaciones de la Tabla 4-5. El cálculo de estas medidas y métricas se hace con base en los datos del archivo *analytics.csv*. Cabe resaltar que, solo 17 cursos de los 22 totales poseen el archivo *analytics.csv*, puesto que la implementación de estas herramientas en UNCode fue hecha a partir del segundo semestre del 2020. En este sentido, dentro del notebook 4.4 Tasas uso servicios se genera un DataFrame con las siguientes

Tabla 4-6: Métricas de software a partir de los archivos de código fuente enviados por estudiantes

Categoría	Métrica	Descripción/Ecuación
Métricas de software	Líneas de código (NLOC)	Número de líneas de código fuente excluyendo comentarios.
	Conteo de tokens (<i>Conteo_tokens</i>)	Número de palabras reservadas del lenguaje del programación utilizadas en el programa construido.
	Complejidad ciclomática (G)	Número de bloques de decisión que contiene el código más uno.
	Vocabulario (n) ^{a,b}	$n = n_1 + n_2$
	Longitud (N) ^{c,d}	$N = N_1 + N_2$
	Longitud calculada (L)	$L = n_1 \cdot \log_2(n_1) + n_2 \cdot \log_2(n_2)$
	Volumen (V)	$V = N \cdot \log_2(n)$
	Dificultad (D)	$D = \frac{n_1}{2} \cdot \frac{N_2}{n_2}$
	Esfuerzo (E)	$E = D \cdot V$
	Tiempo de programación (T)	$T = E/18$
Errores (B)	$B = V/3000$	
	Índice de mantenibilidad (MI) ^{e,f,g}	Medida de la facilidad de modificación y mantenimiento del código fuente. Se encuentra en una escala de 0 a 100.

^a n_1 : Número de operadores distintos.

^b n_2 : Número de operandos distintos.

^c N_1 : Número total de operadores utilizados.

^d N_2 : Número total de operandos utilizados.

^e $MI = \max[0.1 \cdot \sqrt{171 - 5.2 \cdot \ln(V) - 0.23 \cdot G - 16.2 \cdot \ln(SLOC) + 50 \cdot \sin(2.4 \cdot C)} / 171]$

^f *SLOC* : Número de líneas de código.

^g *C* : Porcentaje de líneas de comentarios convertidos a radianes.

columnas: código del curso (*course_id*), nombre de usuario del estudiante (*username*) y 10 columnas con la cantidad de accesos y tasas de uso de cada herramienta disponible en la plataforma.

La siguiente categoría es el desempeño académico, el cual es especificado en la Tabla 4-2 y calculado en el notebook 4.5 Rendimiento académico. La cuantificación se hace a partir del archivo *uncode_grades.csv*. En el notebook se crea un DataFrame con columnas correspondientes al código del curso (*course_id*), el nombre de usuario del estudiante (*username*) y la calificación final del estudiante (*uncode_grade*).

La quinta categoría son las medidas y métricas vinculadas a los datos demográficos, las cuales son descritas en la Tabla 4-3 y la sección 4.5. Los cálculos se realizan en el notebook 4.6 Datos demográficos, utilizando el archivo *datos demográficos.csv*. Las columnas de sexo y programa académico de cada estudiante son utilizadas sin modificaciones. En contraste, la fecha de nacimiento es utilizada para calcular la edad del alumno en el momento en que cursaba la asignatura. Sin embargo, cabe resaltar que debido a errores de digitación en las respuestas de los estudiantes, algunos cálculos de edades son iguales a cero. Estas edades erróneas son reemplazados por el promedio de los datos el cual es 17.4. La siguiente categoría son las medidas relacionadas con las preguntas cerradas de la encuesta de percepción sobre el uso de UNCode, las cuales son especificadas en la Tabla 4-3. El archivo con las preguntas cerradas es generado en el notebook 4.7 Escala likert , donde a partir del archivo *encuestas.csv* se crea un DataFrame, el cual posee 7 columnas correspondientes al nivel de acuerdo o de desacuerdo de los estudiantes en la escala numérica de Likert en relación con las afirmaciones de cada pregunta cerrada.

La última categoría son las métricas de software, que son cuantificadas dentro del notebook 4.8 Métricas de software. Estas métricas son especificadas en la Tabla 4-6 y calculadas utilizando las librerías *lizard* y *radon* de Python. La librería *lizard* se utiliza para cuantificar el número de líneas sin comentarios (NLOC) y la cantidad de tokens utilizados en el programa (*Conteo_tokens*). La librería *radon* es aplicada para calcular la complejidad ciclomática (*G*), el índice de mantenibilidad (*MI*) y las métricas de Halstead (vocabulario, longitud, longitud calculada, volumen, dificultad, esfuerzo, tiempo de programación y errores). Estas librerías emplean los archivos enviados por los estudiantes en las actividades de clase como parámetro para el cálculo de las métricas, los cuales se encuentran almacenados en el directorio *submission* de cada curso. Primero, las métricas de software son calculadas para cada archivo almacenado en la plataforma. Posteriormente, se estima el promedio de las métricas para cada alumno del conjunto de datos. Este procedimiento genera un DataFrame con 12 columnas, las cuales corresponden a los promedios de las métricas por estudiante. Finalmente, los DataFrame generados en cada notebook son transformados en archivos **csv**, que a su vez son almacenados en la carpeta denominada Learning Analytics 2/4) Análisis/Cuantitativo.

4.7. Análisis exploratorio de datos

En esta sección se realiza un análisis univariable de cada una de las medidas y métricas calculadas. La Figura 4-7 visualiza la distribución de intentos realizados por estudiante discriminados por cursos. En la gráfica se evidencia que *pc-group5-2020-2* y *pc-group6-2020-2* son los cursos con menor registro de intentos de solución de actividades con promedios de 34.5 y 32.8. Estos valores son relevantes, dado que los demás cursos tienen más del doble del promedio de intentos. La baja cantidad de envíos posiblemente se relaciona con una poca cantidad de actividades realizadas en UNCode, ya que acorde con los datos de la Figura 4-3 estos cursos son de los que menor cantidad de actividades realizaron en la plataforma; lo anterior hace que la cantidad total de intentos enviados por los estudiantes sea mucho menor en comparación con cursos que se realizan la mayoría de sus tareas en UNCode.

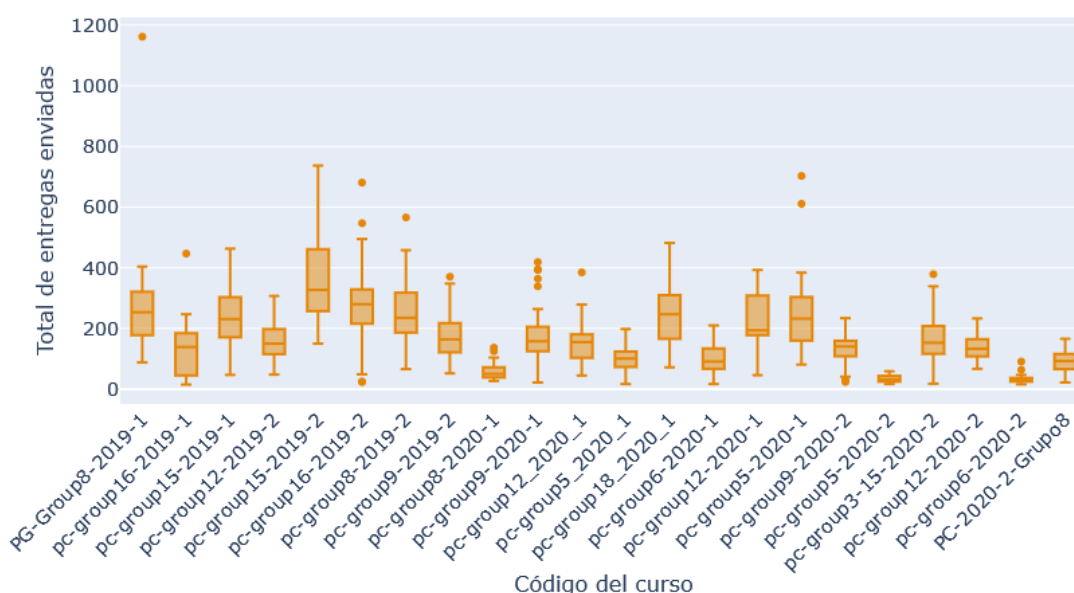


Figura 4-7: Distribución de intentos totales realizados (*Total_Submissions*) por estudiante en cada curso.

Los intentos totales realizados por cada estudiante en promedio son 176.6 durante todo el semestre académico. La desviación estándar de los datos es de 120.8, el valor mínimo es de 15 y el máximo de 1162. Adicionalmente, la desviación estándar corresponde a más del 68 % del promedio de los datos, lo que indica una alta dispersión de los datos. Esta alta dispersión de los valores puede indicar las diferentes técnicas que los estudiantes utilizan en la resolución de problemas. En este sentido, es probable que algunos alumnos optan por realizar muchos envíos con cambios pequeños en cada intento. En contraste, otros estudiantes posiblemente realizan modificaciones extensas en sus códigos de manera local, lo que resulta en pocos intentos dentro de la plataforma.

En el caso del tiempo promedio entre intentos, el diagrama de cajas mostrado en la Figura 4-8 evidencia de manera gráfica una concentración de los datos en valores menores a 2500 min (1.7 días). La concentración de datos puede estar relacionada principalmente con dos factores. Primero, puede indicar que la mayoría de las actividades realizadas en la plataforma son de plazo corto de entrega, como en el caso de quices o evaluaciones. En segundo lugar, es posible que la mayoría de estudiantes intente resolver los ejercicios de programación realizando cambios pequeños en el código diseñado, lo que hace que los intervalos de tiempo entre intentos sean cortos.

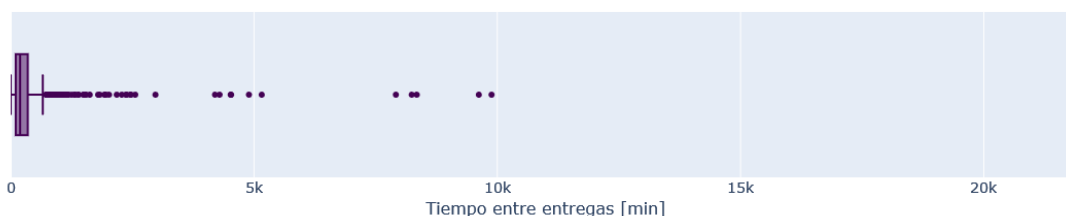


Figura 4-8: Tiempo promedio entre los intentos de cada estudiante *Duration_of_Submission*.

El promedio del tiempo registrado entre entregas es 423.5 min (7.1 h) con una desviación estándar de 1185.8 min (19.8 h). Estos valores evidencian una dispersión muy alta de los datos, puesto que la desviación estándar es más de 100 % del promedio. El máximo tiempo promedio calculado es de 21807.4 min (15.1 días). En este sentido, la variación de los datos se puede justificar teniendo en cuenta que las actividades realizadas en los cursos son de diferentes características; es decir, las actividades consideradas son tanto quices y exámenes con tiempo de resolución corto, como parciales y proyectos que permiten plazo más largos como días o semanas.

Por otra parte, con respecto a los veredictos obtenidos, la Figura 4-9) evidencia que el juicio con mayor cantidad de obtención es respuesta incorrecta con un 48.9 % del total de veredictos registrados. Este resultado puede indicar que una cantidad alta de estudiantes tiene dificultades con el entendimiento de los objetivos de las actividades dado que logran construir programas ejecutables pero no obtienen la respuesta esperada. El segundo veredicto es respuesta correcta con un 31.7 %. Los veredictos que le siguen en magnitud son error de ejecución y límite de tiempo excedido con 14.6 % y 3.9 %, respectivamente. Y los veredictos con menor cantidad de obtención son error de compilación, límite de memoria excedido y límite de resultado excedido con valores menores al 1.0 %. Los veredictos con porcentajes al 15 %, corresponden a los juicios relacionados con dificultades para aplicar de manera práctica los conocimientos del curso. Estos resultados indican que la mayoría de estudiantes son exitosos en el diseño de soluciones ejecutables, aplicando de conocimientos en programación de computadores.

La Figura 4-10 muestra la distribución de los porcentajes de obtención de veredictos en cada curso de programación. En general se evidencia una distribución similar a la de la Figura 4-9, donde los veredictos de mayor obtención son respuesta correcta e incorrecta.

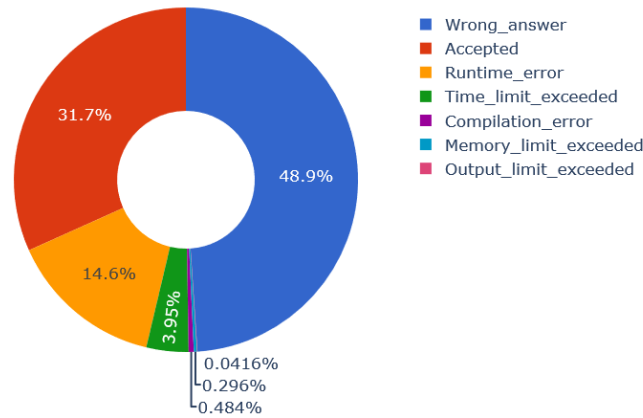


Figura 4-9: Tasas totales de veredictos obtenidos en todos los cursos.

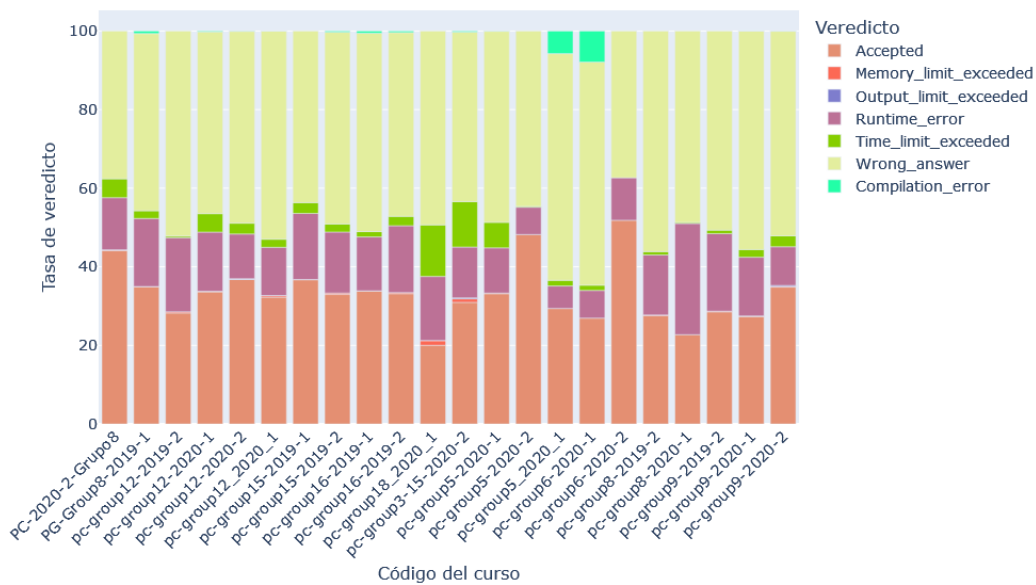


Figura 4-10: Tasas totales de veredictos obtenidos en todos los cursos.

En la gráfica se puede evidenciar para los cursos *pc-group18_2020_1* y *pc-group3-15-2020-2* el porcentaje de tiempo límite excedido es mayor en comparación con otros cursos. Los errores de límite de tiempo excedido indican dificultades para la construcción de programas eficientes, esto puede evidenciar la necesidad de reforzar las habilidades de los estudiantes para identificar algoritmos sencillos para la resolución de problemas. Por otra parte, en los cursos *pc-group5_2020_1* y *pc-group6-2020-1* se evidencia un alto porcentaje de errores de compilación en comparación con los demás cursos. Estos resultados puede representar falencias más concretos en los conocimientos de programación como en el uso correcto de comas, paréntesis, comillas o palabras clave.

El uso de las herramientas disponibles en la plataforma está representado en la Figura 4-11, donde

se especifican los porcentajes de acceso en el conjunto de datos. En este sentido, la herramienta más utilizada por los estudiantes es *Custom_input* con el 65.0 % del total de accesos registrados en la plataforma. Este resultado indica que la mayoría de los alumnos prefieren probar la efectividad de sus programas con pruebas diseñadas por ellos mismos.

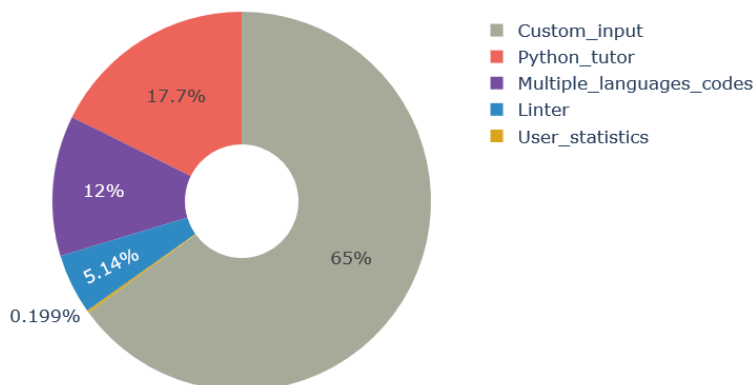


Figura 4-11: Tasas totales de uso de herramientas en todos los cursos.

Adicionalmente, están las tasas *Python_Tutor* y *Multiple_Languages_Codes* con porcentajes de 17.7 % y 12.0 %, respectivamente. El uso de *Python_Tutor* indica que algunos estudiantes consideran una ayuda significativa observar paso a paso el flujo de ejecución del programa construido, posiblemente para la ubicación de errores y afianzamiento de conocimientos. Por su parte, el uso de *Multiple_Languages_Codes* indica cuándo los estudiantes someten sus soluciones a evaluación. Por último, las herramientas menos utilizadas en la plataforma son *Linter* y las estadísticas de usuario con 5.1 % y 0.5 %. El poco uso de estas herramientas, puede indicar que los estudiantes consideran que la información que aportan estas herramientas no generan una retroalimentación suficiente para ayudar al estudiante a mejorar la solución construida.

La Tabla 4-7 resume los valores estadísticos que describen las métricas de software, que son calculadas con base en los programas construidos por los estudiantes como soluciones a las actividades del curso. Los resultados de la tabla evidencian que la métrica con mayor promedio corresponde al esfuerzo con 1290.0 y la métrica de menor promedio es errores con 0.1. Con respecto a la desviación estándar, se considera alta dispersión en los datos cuando la desviación estándar es mayor que el promedio estadístico. En este sentido, las métricas con mayor dispersión de datos son esfuerzo y tiempo de programación con desviaciones de 1603.8 y 89.1, respectivamente. Esta alta dispersión en estas métricas muestran la diversidad de soluciones que son construidas por los estudiantes. En este sentido, los algoritmos diseñados por los estudiantes pueden diferir en gran magnitud con respecto a la complejidad, aumentando o disminuyendo en otros casos el esfuerzo y el tiempo que representa cada programa enviado, de acuerdo a las métricas de Halstead.

Tabla 4-7: Estadísticas descriptivas de las métricas de software

Métrica	Promedio estadístico	Desviación estándar	Mínimo	Máximo
Líneas de código (NLOC)	20.7	9.4	2.0	92.1
Conteo de tokens (<i>Conteo_tokens</i>)	150.2	66.1	25.0	565.4
Complejidad ciclomática (G)	7.8	4.4	1.0	42.2
Vocabulario (<i>n</i>)	20.8	10.0	3.0	82.0
Longitud (N)	42.0	25.7	3.0	225.2
Longitud calculada (L)	88.3	66.3	2.0	534.7
Volumen (V)	214.6	117.5	4.8	1740.8
Dificultad (D)	3.7	1.3	0.5	12.6
Esfuerzo (E)	1290.0	1603.8	2.4	20181.8
Tiempo de programación (T)	71.7	89.1	0.1	1121.2
Errores (B)	0.1	0.1	0.002	0.6
Índice de mantenibilidad (MI)	60.9	6.4	40.1	86.1

Por otra parte, las métricas con menor dispersión de datos son el índice de mantenibilidad y la dificultad con dispersiones de 6.4 y 1.3, que corresponden al 10.5 % y 35.1 % de sus respectivos promedios. La baja dispersión de estos datos posiblemente indica que los estudiantes de este curso tienen capacidades y habilidades similares para la construcción de programas; estos resultados tienen sentido, debido a que la asignatura *Programación de Computadores* es un curso de tipo introductorio y en algunos casos el primer acercamiento del alumno a los lenguajes de programación. En específico, el índice de mantenibilidad adicional a tener una desviación estándar baja tiene un promedio estadístico de 60.9, el cual es mayor a la mediana del rango de esta métrica de software con un valor mínimo de 0 y un valor máximo de 100.

La Figura 4-12 muestra la distribución de desempeños académicos promedio en cada grupo de la asignatura. Los valores se encuentran en un rango de 0.0 a 5.0, con nota mínima aprobatoria de 3.0. El promedio total es de 4.1 con una desviación estándar de 0.9. Se evidencia que el 50 % de los datos centrales se concentran en un rango intercuartílico de 1.0, el cual se encuentra entre 3.7 (1° cuartil) y 4.7 (3° cuartil). Esta concentración de los datos indica que el rendimiento académico en el curso de la mayoría de los estudiantes es un desempeño medio-alto, dado que esta por encima de la nota mínima aprobatoria y un poco por debajo de la calificación máxima. Teniendo en cuenta que la mayor parte de los estudiantes aprueban la asignatura, es posible afirmar que al finalizar el curso una gran parte de los alumnos adquieren conocimientos y habilidades para la resolución de ejercicios de programación de computadores.

Adicionalmente se realiza un análisis de las medidas y métricas de la categoría de datos demográficos de los estudiantes. En primer lugar, se analiza la distribución de sexos de los alumnos considerados

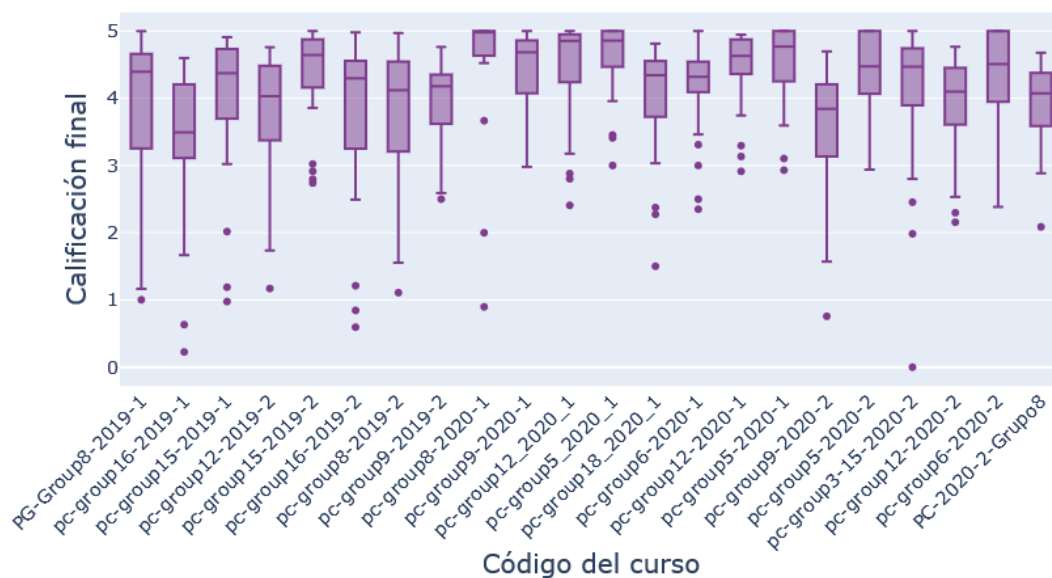


Figura 4-12: Distribución del desempeño académico (*uncode_grade*) por grupo

en el conjunto de datos. La Figura 4-13 resume los resultados de los porcentajes de estudiantes discriminados por sexo. En el gráfico circular (arriba Figura 4-13) se evidencia que la cantidad de mujeres presentes en el conjunto de datos es bajo, dado que es menor al 20 % del total de la muestra. En este sentido, el alto porcentaje de hombres (84 %) puede estar relacionado con los programas académicos de los estudiantes que cursan la asignatura de Programación de Computadores.

La gráfica de barras discriminada por programas académicos (abajo Figura 4-13) muestra que para los ocho programas académicos de ingeniería el porcentaje de mujeres es siempre menor al 50 % de los estudiantes. En este sentido, el programa con mayor porcentaje de mujeres es ingeniería química con 40.5 %. Sumado a esto, las carreras con menor porcentaje de mujeres son ingeniería mecánica e ingeniería civil con 6.7 % y 6.2 % de estudiantes mujeres. En general, ocho de las nueve opciones de programa académico corresponde a ingenierías, las cuales son carreras donde generalmente existe carencia de mujeres. Adicionalmente, inclusive en la opción Otro, donde podrían estar incluidas carreras donde la mayoría son mujeres, la totalidad de los estudiantes son hombres.

Por otra parte, con respecto a los programas académicos de los estudiantes considerados en el conjunto de datos, la Figura 4-14 muestra la distribución general de carreras. La gráfica evidencia que ingeniería de sistemas es el programa con el porcentaje más alto de estudiantes con 37.8 %. Sumado a esto, los programas con porcentajes entre 10 % y 20 % son ingeniería electrónica e ingeniería química. Tiene sentido que estas tres ingenierías sean los programas con mayor cantidad de estudiantes, dado que son carreras donde los conocimientos de programación de computadores son indispensables en otras asignaturas del programa. En orden de magnitud, las siguientes carreras son ingeniería in-

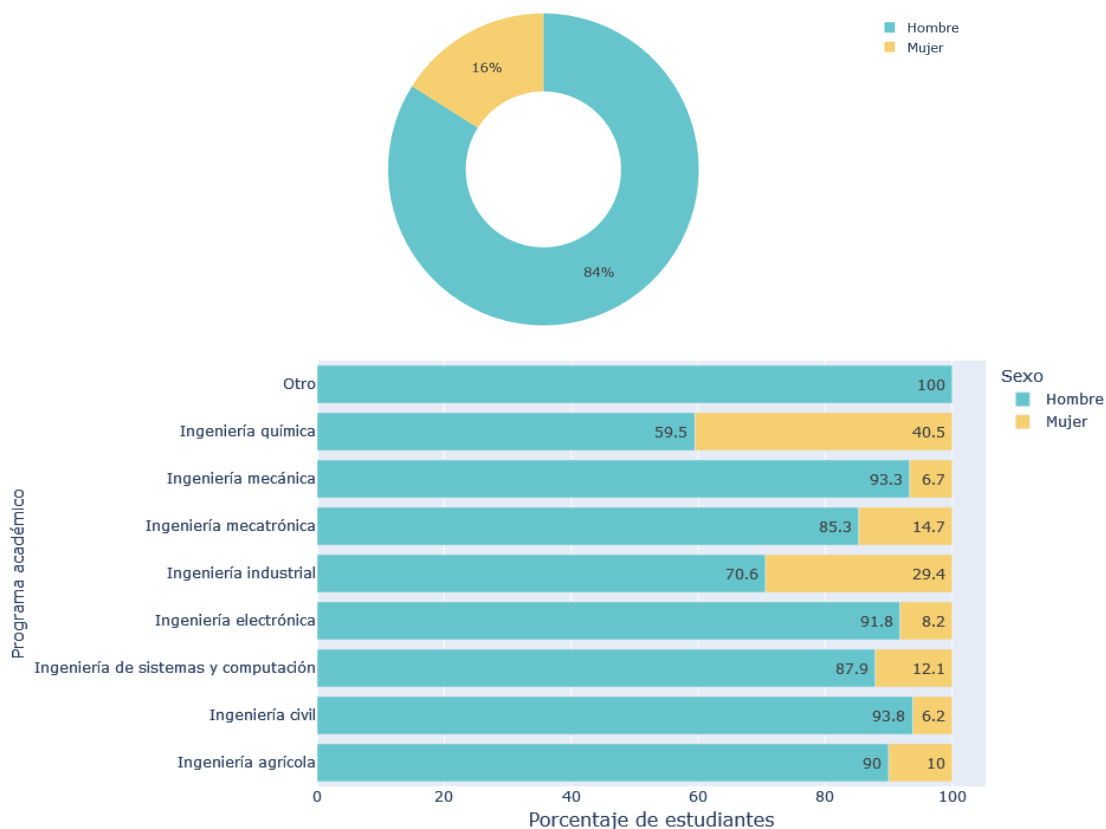


Figura 4-13: 1) Arriba: Porcentaje de estudiantes por sexo, 2) Abajo: Porcentaje de estudiantes por sexo discriminados por programa académico.

dustrial y mecatrónica con el mismo porcentaje de estudiantes de 9.74 %. Por último, los programas académicos con porcentajes menores al 5 % corresponden a ingeniería civil, mecánica, agrícola y la opción otro. Tiene sentido que estas tres ingenierías correspondan a los menores porcentajes, puesto que son carreras donde las aplicaciones de programación de computadores se realizan con software y lenguajes de dominio específico.

La Figura 4-15 resume la distribución de edades de los alumnos considerados en el conjunto de datos de las encuestas realizadas. El histograma general de edades (arriba Figura 4-15) evidencia que las edades de los estudiantes se concentran en el intervalo entre 17 y 19 años. Adicionalmente, el gráfico muestra que el rango total de edades se encuentra entre 16 y 42.

En el histograma discriminado por sexo (abajo Figura 4-15) se evidencia una concentración de datos en ambas categorías, similar a la del histograma general. Sin embargo, el rango de edades entre los hombres y las mujeres evidencia diferencias significativas. En el caso de las mujeres el rango es menor, dado que inicia en 16 y finaliza en 22. En contraste, las edades de los hombres inician también en 16

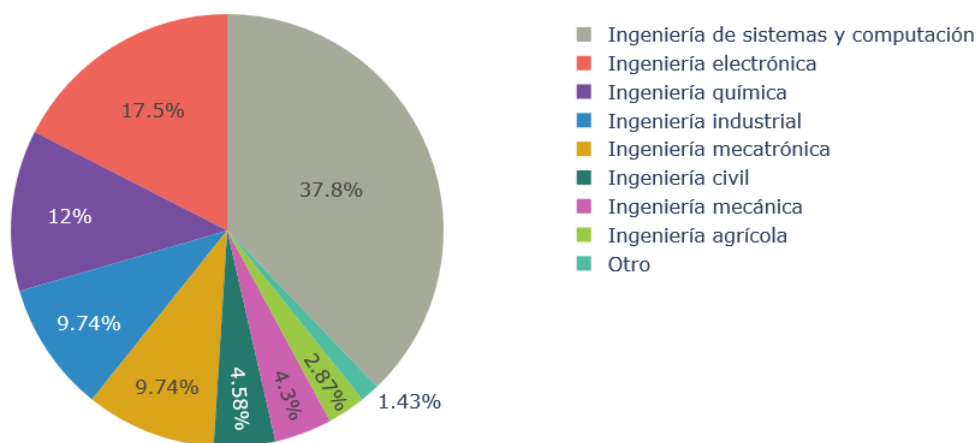


Figura 4-14: Porcentaje de estudiantes por carrera

pero tienen su valor máximo en 42. Esta diferencia en el rango de edades pueden estar relacionadas con el bajo porcentaje de mujeres consideradas en el conjunto de datos (4-13), y no necesariamente significa una diferencia significativa de edades entre mujeres y hombres. Adicionalmente, tanto el histograma de hombres como el de mujeres evidencia una disminución del número de estudiantes en las edades mayores a 20. Estos resultados indican que la muestra de estudiantes considerados son jóvenes, lo cual posiblemente se debe a que la asignatura de *Programación de Computadores* es un curso dirigido a estudiantes de pregrado.

La Figura 4-16 muestra la distribución de edades discriminada por curso. Acorde con la gráfica, en los cursos *pc-group15-2019-1* y *pc-group9-2020-1* se encuentran los estudiantes con mayores edades (42 y 40 años, respectivamente). Estos valores son interesantes, puesto que son más del doble del promedio de los datos. Estos dos estudiantes son considerados excepciones, dado que la asignatura *Programación de Computadores* es considerada como un curso introductorio, es decir, intenta ser la primera aproximación de los alumnos al uso de lenguajes de programación. Adicionalmente, la edad mínima en los datos es de 16, el promedio es de 18.6 con una desviación estándar de 2.5. En este sentido, el 50 % de los datos centrales se concentran en un rango intercuartílico de 1.6, el cual se encuentra entre 17.4 (1° cuartil) y 19 (3° cuartil). Estos resultados demuestran que la muestra de estudiantes tiende a ser joven, y posiblemente la razón es que generalmente la asignatura es cursada en los primeros semestres de universidad.

Finalmente, se hace un análisis de las medidas generadas de las preguntas cerradas de las encuestas de percepción. La Figura 4-17 resume los porcentajes de respuestas de las primeras tres preguntas cerradas de la encuesta. La distribución de porcentajes de las respuestas a la pregunta “UNCode fue útil en su proceso de aprendizaje de programación de computadores” (arriba Figura 4-17), evidencia que más del 90 % de los estudiantes está de acuerdo con la sentencia, incluyendo los tres niveles: algo de acuerdo, de acuerdo y totalmente de acuerdo. En este sentido, el restante 10 % dice estar en desacuerdo con

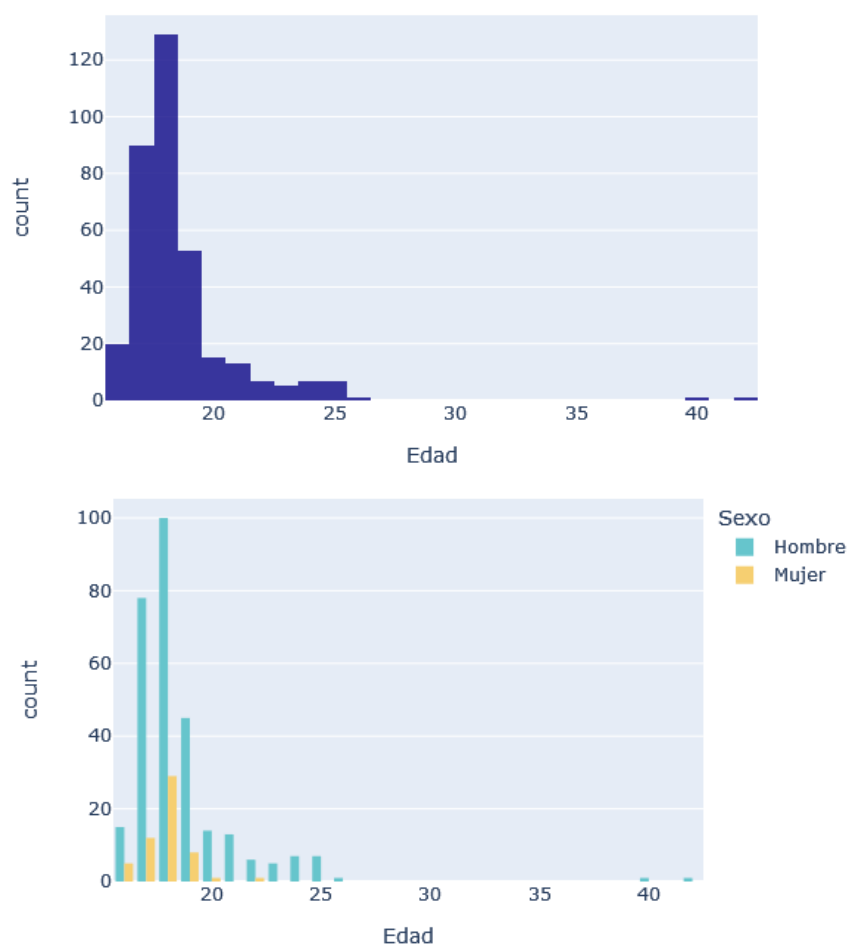


Figura 4-15: 1) Arriba: Histograma general de edades, 2) Abajo: Histogramas de edad discriminados por sexo

la utilidad de UNCode en el proceso de aprendizaje del curso. La gráfica de la pregunta “UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura” (centro Figura 4-17) muestra que más del 92 % está de acuerdo con la sentencia, y menos del 8 % considera que la plataforma no es útil para la calificación automática de los programas enviados.

Por último, la gráfica con las respuestas a la pregunta “La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas” (abajo Figura 4-17) muestra que más del 85 % está de acuerdo con la premisa. Y en contraste, menos del 15 % no considera UNCode una plataforma útil para obtener retroalimentación que sirva de ayuda para la corrección de errores. En general, los resultados evidencian que la mayoría de los alumnos consideran que el uso de UNCode logra promover el proceso de aprendizaje en el curso. En específico, los estudiantes utilizan la calificación y retroalimentación automática, puesto que las consideran útiles.

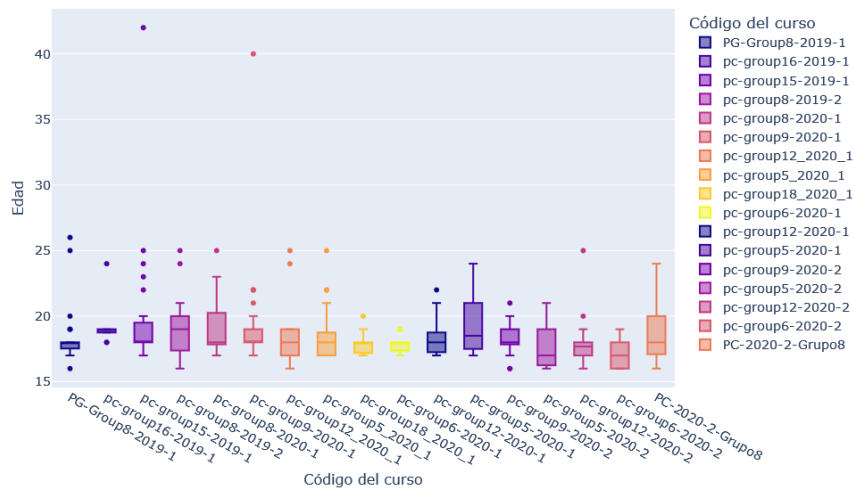


Figura 4-16: Distribución de edades por curso

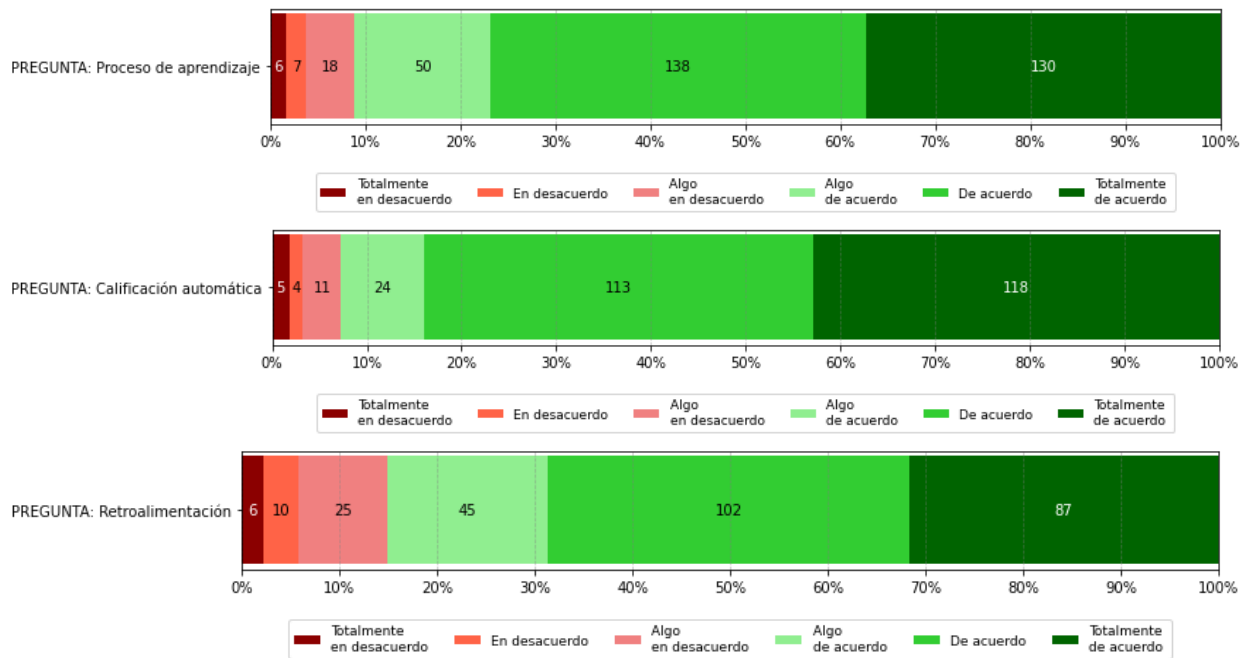


Figura 4-17: Porcentaje de respuestas en escala de Likert de las siguientes preguntas: 1) Arriba: “UN-Code fue útil en su proceso de aprendizaje de programación de computadores”, 2) Centro: “UN-Code fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura” y 3) Abajo: “La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas”.

Acorde con los porcentajes de respuesta de los estudiantes, la funcionalidad que la mayoría de es-

tudiantes reconoce en la plataforma (92 %) es la calificación automática de programas diseñados. En contraste, la funcionalidad con mayor tasa de desacuerdo (15 %) es retroalimentación automática para corregir errores en la solución construida. Estos resultados pueden indicar que existen algunos vacíos en el uso e interpretación de los veredictos de retroalimentación automática generados en la plataforma.

Por otra parte, la Figura 4-18 muestra el porcentaje de respuestas para la pregunta “Las siguientes actividades fueron útiles para su aprendizaje en esta asignatura”, donde cada columna representa la actividad calificada por el estudiante. La gráfica evidencia que más de la mitad de los estudiantes encuestados optan por la respuesta *No aplica* para las herramientas *Custom input* (54.5 %), *Python tutor* (52.4 %) y estadísticas de usuario (41.5 %); indicando posiblemente que no utilizaron estas herramientas durante todo el semestre académico o no tenían conocimiento su existencia o funcionamiento.

Acorde con la gráfica más del 30 % de los estudiantes está de acuerdo con la afirmación de que el uso de *Custom input* fue útil en el proceso de aprendizaje y el 13.8 % no está de acuerdo con la afirmación. Por otra parte, más del 38 % de los estudiantes consideran que la herramienta *Linter* es de utilidad para afianzar los aprendizajes del curso. Sin embargo, el 40.4 % de alumnos que no evidencian el beneficio del uso de esta herramienta. Sumado a esto, la gráfica evidencia que un porcentaje mayor al 34 % de alumnos dice estar de acuerdo con el valor de *Python tutor* en el proceso educativo, en comparación con el 13.1 % que dice no estar de acuerdo. Y por último, se evidencia gráficamente que el 38 % está de acuerdo en que utilizar las estadísticas de usuario es provechoso en los procesos de aprendizaje de la asignatura, en contraste con el 20 % que no evidencia el beneficio de acceder a estas estadísticas.

Dentro del conjunto de estudiantes que no responden *No aplica*, es decir, los que utilizan activamente las herramientas de la plataforma, la mayoría está de acuerdo con la utilidad de *User Statistics* en el proceso de aprendizaje. Este resultado indica que posiblemente la mayoría de los estudiantes que acceden a las estadísticas de usuario son capaces de realizar procesos de auto-reflexión y autorregulación, lo que tiene un impacto positivo en sus procesos de aprendizaje. En este sentido, acorde con los porcentajes de la Figura 4-18 es posible afirmar que la segunda herramienta que los estudiantes consideran más beneficiosa es *Python tutor*, lo que indica que la visualización paso a paso del programa construido posiblemente permite a la mayoría de los alumnos que la utilizan identificar errores dentro del código, promoviendo el proceso de aprendizaje del estudiante. La tercera herramienta con tasa alta de respuestas positivas es *Custom input*, este resultado indica que la mayoría de los alumnos que hacen uso de pruebas personalizadas en la plataforma, evidencian el beneficio de probar los programas diseñados por medio de entradas provistas por el usuario para afianzar los conocimientos de la asignatura. En contraste, la herramienta que tiene mayor tasa de desaprobación es *Linter*, es posible que este resultado se deba a que la información de la herramienta se concentra en asegurar las buenas prácticas de programación, las cuales en algunos casos entorpecen el aprendizaje de conocimientos básicos de programación.

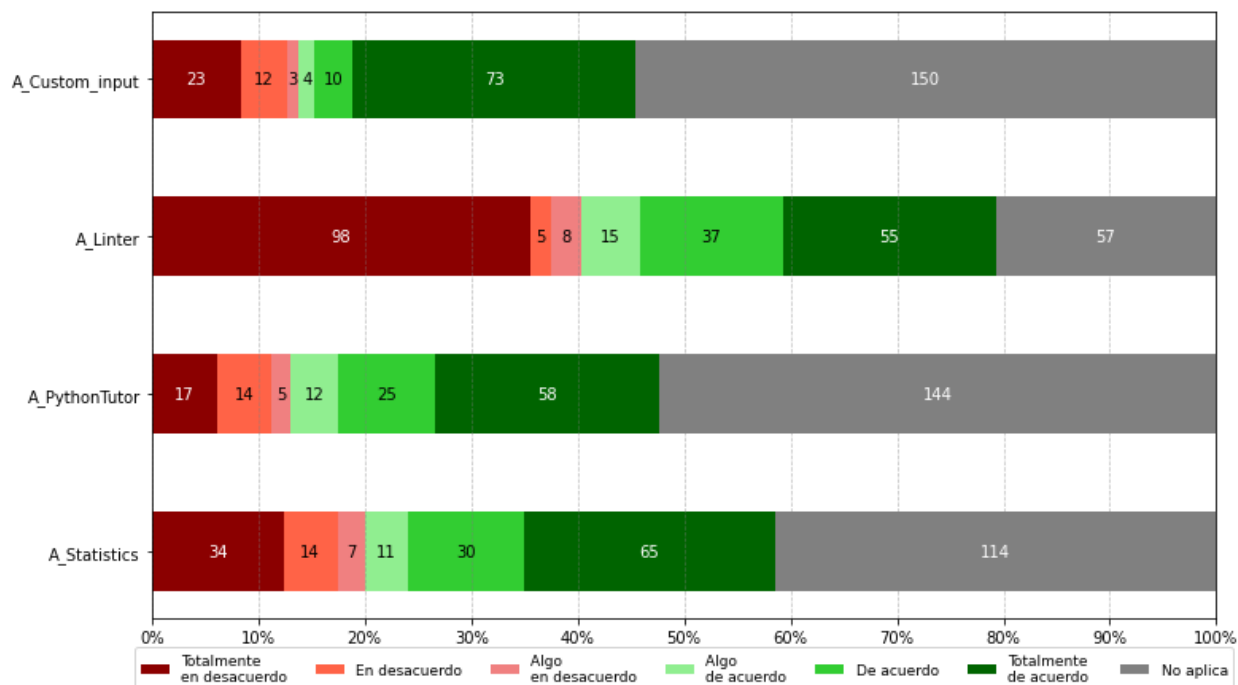


Figura 4-18: Porcentaje de respuestas en escala de Likert a la pregunta “Las siguientes actividades fueron útiles para su aprendizaje en esta asignatura”

4.8. Aplicación de técnicas de análisis y modelación de datos

En este último paso, el objetivo es identificar cuáles de las medidas o métricas están relacionadas significativamente con el desempeño académico de los estudiantes de la asignatura Programación de Computadores. En este sentido, se hace un análisis de correlaciones estableciendo como variable dependiente el rendimiento académico del estudiante (*uncode_grade*).

En primer lugar, se hace una prueba de normalidad de los datos, con el objetivo de seleccionar la prueba estadística apropiada para el cálculo de las correlaciones. La Figura 4-19 muestra la distribución de los desempeños académicos de los estudiantes considerados en el conjunto de datos. La gráfica evidencia una concentración de datos en las calificaciones mayores a 4.0. Adicionalmente, se realiza el test de *Shapiro-Wilks* el cual tiene como hipótesis nula que la muestra de datos proviene de una distribución normal y la hipótesis alterna permite afirmar que la distribución no es normal. Al realizar el test con los datos del desempeño académico de los estudiantes se obtiene un *p-valor* ≤ 1.00 , lo cual indica que no podemos asumir normalidad en los datos. En este sentido, se emplea el coeficiente de correlación de *Spearman*, dado que este coeficiente no requiere que las muestras estén normalmente distribuidas. Este coeficiente se utiliza para cuantificar las correlaciones entre las variables no cate-

góricas. En este sentido, para este análisis no se consideran las variables *Sexo* y *Programa académico*.

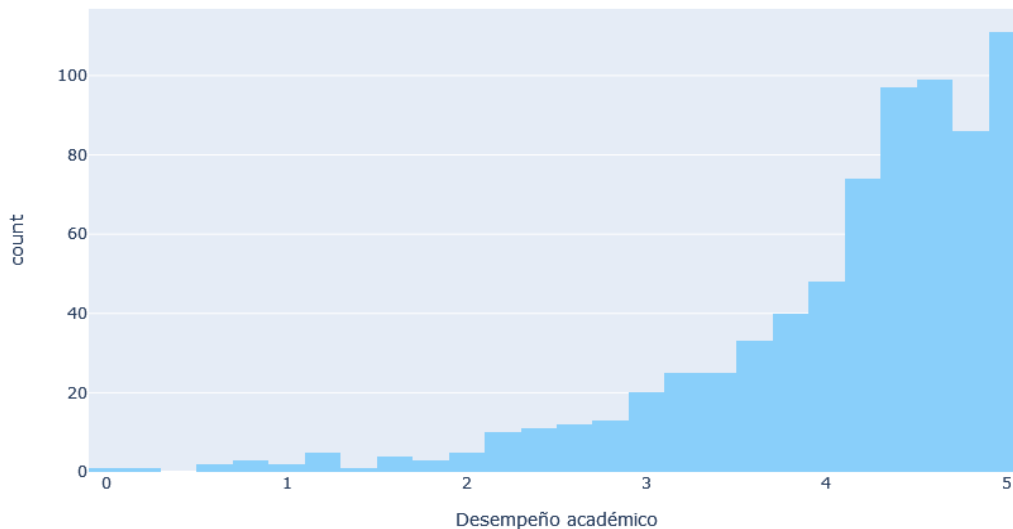


Figura 4-19: Histograma de las calificaciones finales de los estudiantes de la asignatura

La Figura 4-20 ilustra las 29 medidas y métricas que tienen correlación estadísticamente significativa ($p\text{-valor} \leq 0.05$) con el desempeño académico. En la parte izquierda de la figura se encuentran las variables con coeficiente de correlación positivo y en la parte derecha las de correlación negativa. Adicionalmente, la gráfica evidencia que hay una mayor cantidad de medidas y métricas con correlación positiva (19) en comparación con las de correlación negativa (10). Adicionalmente, la categoría de medidas y métricas con mayor número de variables correlacionadas positivamente con el desempeño académico son las basadas en los veredictos obtenidos en los intentos de solución (5 veredictos y 4 tasas). Solo hay cuatro respuestas de las preguntas de las encuestas, tres métricas de software y una tasa de uso de herramientas presente en este grupo de variables. En contraste, la categoría de medidas y métricas con mayor número de variables correlacionadas negativamente con el desempeño académico son las relacionadas con el uso de herramientas (4 números de accesos y 2 tasa de uso). Solo hay 3 tasas de veredictos y una métrica de software con coeficiente de correlación negativo.

La variable con la mayor correlación positiva es el número de respuestas correctas (*Accepted*) con coeficiente de 0.41. Este es un resultado esperado, puesto que un estudiante con una alta cantidad de respuestas correctas representa un alumno que tiene éxito en solucionar las actividades de clase. En orden de magnitud le siguen la PREGUNTA: Proceso de aprendizaje y PREGUNTA: Calificación automática con coeficientes de 0.26 y 0.25, respectivamente. Estas correlaciones tienen sentido, dado que los alumnos que están de acuerdo con la utilidad de la plataforma en el proceso de aprendizaje y los beneficios de obtener calificación automática, posiblemente son estudiantes que aprovechan las herramientas de la plataforma para probar de manera efectiva y mejorar los programas construidos.

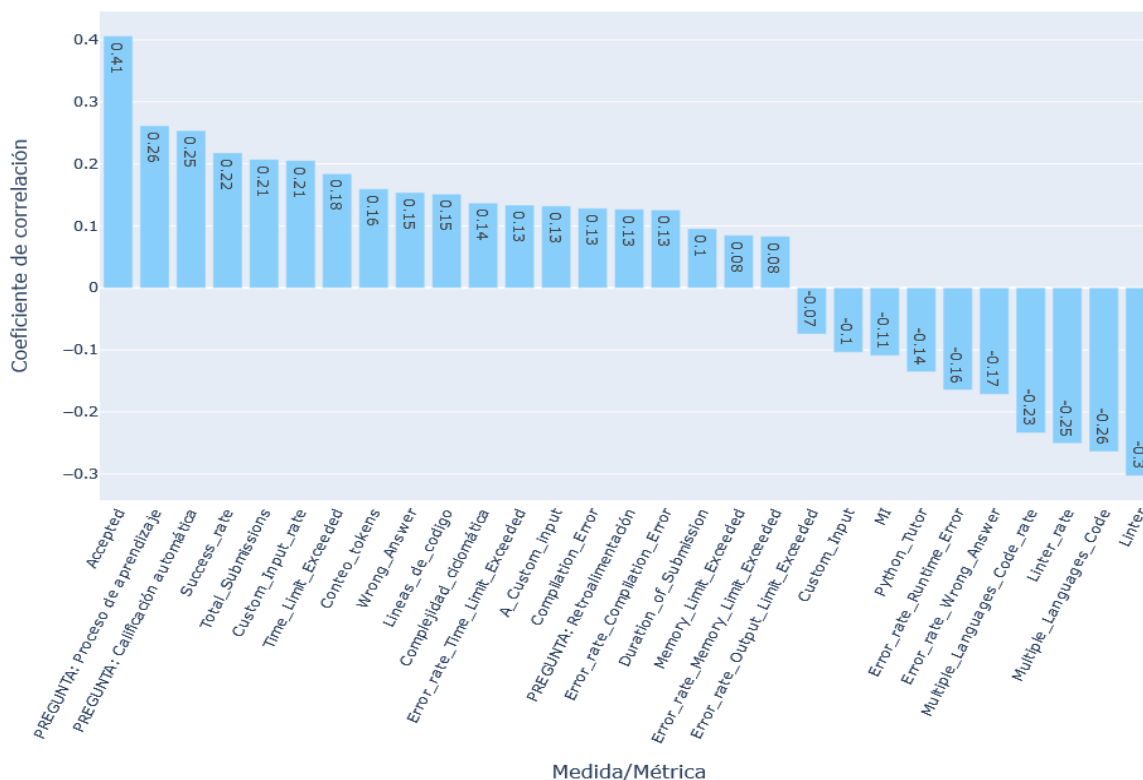


Figura 4-20: Variables con correlaciones significativas con respecto al desempeño académico

Por otra parte, está la tasa de respuestas correctas (*Success_rate*) con una correlación de 0.22. Este es un resultado esperado, dado que un estudiante con alto porcentaje de respuestas correctas, demuestra tener habilidades para la solución de ejercicios de programación. Luego, están el total de intentos realizados (*Total_Submissions*) y la tasa de uso de Custom Input (*Custom_Input_rate*) ambas con un coeficiente de correlación de 0.21. Estas correlaciones positivas son resultados esperados, dado que los estudiantes que tienen la capacidad de enviar una alta cantidad de soluciones y habilidad de realizar pruebas personalizadas del código construido, es probable hayan afianzado los suficientes conocimientos para la construcción y corrección de programas, lo que puede promover buenas calificaciones en las actividades del curso. Posteriormente, están las variables con correlaciones entre 0.20 y 0.15, las cuales corresponden a la cantidad de intentos con límite de tiempo excedido (*Time_Limit_Exceeded*), el conteo de tokens (*Conteo_tokens*), el número de respuestas incorrectas (*Wrong_Answer*) y la cantidad de líneas de código (NLOC) del programa construido por el estudiante. La cantidad de tokens y líneas de código son métricas que se esperaba tuvieran correlación positiva, puesto que a medida que el alumno diseña programas de alto contenido demuestra conocimientos en programación y posiblemente genera mejores desempeños académicos. En contraste, no se esperaba que la cantidad de errores de límite de tiempo excedido y respuestas incorrectas estuvieran en este grupo de variables, dado que estos veredictos indican falencias en la solución propuesta por el estudiante. Sin embargo,

una alta cantidad de envíos con estos tipos de error también puede indicar múltiples intentos incorrectos de solución hasta que el estudiante obtiene la solución correcta.

En la gráfica hay un grupo de medidas y métricas con correlaciones positivas menores a 0.15 y mayores a 0.10 que son la complejidad ciclomática (G), la tasa $Error_rate_Time_Limit_Exceeded$, las respuestas sobre la pregunta de Custom input (A_Custom_input), el número de errores de compilación ($Compilation_Error$), la PREGUNTA: Retroalimentación y la tasa de $Error_rate_Compilation_Error$. La correlación positiva de la complejidad ciclomática es un resultado no esperado, puesto que la construcción de un programa de alta complejidad ciclomática indica probablemente que el estudiante no logra utilizar las estructuras lógicas de manera óptima y eficiente en el código escrito. Adicionalmente, las correlaciones de las respuestas a la PREGUNTA: Retroalimentación y sobre Custom input indican que posiblemente los estudiantes con altos desempeños son los que logran utilizar la retroalimentación que ofrece la plataforma y el uso de pruebas personalizadas para el mejoramiento de las soluciones diseñadas. Las tasas de límite de tiempo excedido, la cantidad de veredictos y tasa de error de compilación son variables que no se esperaba tuvieran correlación positiva, puesto que estos veredictos indican que el programa del estudiante debe ser refinado. A pesar de esto, la correlación positiva de estas tasas de error con el rendimiento académico puede estar asociada con la retroalimentación formativa que ofrecen los veredictos; por el hecho de que los estudiantes con estos tipos de errores posiblemente obtienen información orientativa por medio de los veredictos, lo que les permite realizar correcciones hasta dar resolución al problema de programación.

Posteriormente, está el tiempo promedio entre entregas ($Duration_of_Submission$) con correlación de 0.1, lo que puede indicar que posiblemente los estudiantes que realizan cambios grandes en sus programas, que se traduce en mayor tiempo entre envíos, demuestran mayor aprendizaje para lograr solucionar efectivamente los ejercicios de clase. Las variables con menor correlación positiva son el número de veredictos y tasa de obtención de límite de memoria excedido ($Memory_Limit_Exceeded$ y $Error_rate_Memory_Limit_Exceeded$) ambos con coeficientes de 0.08. La cantidad y tasa de límite de memoria excedido son variables que no se esperaba tuvieran correlación positiva, puesto que estos veredictos indican que el programa del estudiante debe ser refinado para dar resolución a la actividad de programación. Adicionalmente, las correlaciones de estas variables son muy bajas (menores a 0.1), lo que indica que la intensidad de la asociación con el desempeño académico es muy débil y por esta razón no resulta útil generar una hipótesis sobre los resultados obtenidos.

Por otra parte, las variable con mayor correlación negativa son el número de accesos a *Linter* y a *Multiple_Languages_Codes* con valores de -0.3 y -0.26 , respectivamente. Posteriormente, están las tasas de uso de las anteriores dos herramientas (*Linter_rate*) y *Multiple_Languages_Codes_rate*) con -0.25 y -0.23 . Obtener correlaciones negativas para estas medidas y métricas no son resultados esperados, considerando que ambas son herramientas presentes en la plataforma con el objetivo de apoyar el proceso de aprendizaje de los alumnos.

Posteriormente, están las variables con correlaciones entre -0.20 y -0.10 que corresponden a la tasa *Error_rate_Wrong_Answer*, la tasa de obtención de errores de ejecución (*Error_rate_Runtime_Error*), total de accesos a *Python tutor*, el índice de mantenibilidad promedio (MI) y la cantidad de accesos a Custom Input (*Custom_Input*). Finalmente, la métrica con menor correlación negativa es la tasa de errores de límite de resultado excedido con un valor de -0.07 . En estos resultados llama la atención que la cantidad de accesos a Python tutor y a custom input tengan correlaciones negativas con el desempeño académico, puesto que estas herramientas están diseñadas para guiar a los estudiantes durante el proceso de construcción de sus soluciones a las actividades de la asignatura. El índice de mantenibilidad y las tasas de respuestas incorrectas, error de ejecución y límite de resultado excedido si se esperaba tuvieran correlación negativa, dado que un programa con índice de mantenibilidad alto indica un nivel bajo de habilidades de programación y un estudiante con una alta tasa de respuestas incorrectas, errores de ejecución y límite de resultado excedido significa que la mayoría de los intentos realizados no fueron exitosos.

Finalmente, los valores de desempeños académicos fueron clasificados en estudiantes aprobados, es decir, con calificaciones finales iguales o mayores a 3.0 y los alumnos que no aprobaron la asignatura. Lo anterior, con el objetivo de segmentar el conjunto de datos e identificar si las correlaciones encontradas cambian entre estudiantes con desempeños altos en comparación con los alumnos de desempeños bajos. La Figura 4-21 muestra las medidas y métricas con correlaciones significativas ($p\text{-valor} \leq 0.05$) diferenciadas en las dos categorías: *Aprobado* y *Reprobado*. En la parte superior de la gráfica se encuentran las correlaciones positivas y en el área inferior de la figura se encuentran las variables con correlaciones negativas. A partir de la gráfica se puede evidenciar que la cantidad de variables con correlaciones positivas es mayor para el grupo de estudiantes que aprueba la asignatura (12) en comparación con los estudiantes que no aprueban (6). El mismo comportamiento se observa para las medidas y métricas con correlación negativa, donde el grupo de alumnos aprobados tiene más variables con correlaciones significativas (10) con respecto a los no aprobados (3).

Al concentrarnos en las variables que resultan ser significativas en ambas categorías de estudiantes se pueden observar 6 medidas y métricas con coeficientes de correlación positivos. Las medidas y métricas corresponden a la cantidad de respuestas correctas (*Accepted*), la tasa de respuestas correctas (*Success_rate*), número total de veredictos de tiempo límite excedido (*Time_Limit_Exceeded*), los intentos totales realizados (*Total_Submissions*), la cantidad de respuestas incorrectas (*Wrong_Answer*) y la tasa de límite de tiempo excedido (*Error_rate_Time_Limit_Exceeded*). En todos los casos, las correlaciones positivas disminuyen en el grupo de los estudiantes que aprueban la asignatura. Se esperaba que la correlación de la cantidad de respuestas incorrectas, la cantidad de veredictos y la tasa de límite de tiempo excedido fuera mayor en el grupo de estudiantes reprobados, puesto que estos veredictos indican dificultades en el proceso de solución de actividades de programación. Sumado a esto, el aumento en la magnitud de la correlación de los intentos totales realizados tiene sentido, teniendo en cuenta que los estudiantes que tienen dudas sobre los temas tienden a realizar más intentos que los alumnos que tienen claro el tema. Por otra parte, no se esperaba que la cantidad de veredictos



Figura 4-21: Variables con correlaciones significativas con respecto al desempeño académico de los estudiantes discriminadas por desempeño académico

y tasa de respuestas correctas tuvieron una correlación de mayor magnitud en los estudiantes con desempeños bajos.

Por otra parte, la variable con correlación negativa que comparten los dos grupos de estudiantes es la tasa *Error_rate_Runtime_Error*, la cual también aumenta su magnitud en el caso de los estudiantes reprobados. Una correlación negativa mayor para la tasa de error de ejecución en los estudiantes de bajos desempeños tiene sentido, dado que esto indica una mayor cantidad de intentos que fallan en la ejecución. Adicionalmente, al observar exclusivamente las variables significativas solo para el grupo de alumnos reprobados corresponden a las respuestas sobre la utilidad de Linter (*A_Linter*) y el índice de mantenibilidad. El resultado de *A_Linter* es relevante por las siguientes dos razones: esta correlación es significativa únicamente para el grupo de estudiantes reprobados y se esperaba obte-

ner un correlación positiva, debido a que un estudiante que considera provechosa la herramienta de verificación de buenas prácticas de programación podría resultar en un mayor desempeño académico. En segundo lugar, está el índice de mantenibilidad que evidencia una la correlación más fuerte en comparación con el coeficiente del conjunto de datos sin clasificar (-0.26). Este resultado tiene sentido, porque las soluciones con índice de mantenibilidad altos indican vacíos en los conocimientos y habilidades de programación.

5 Datos cualitativos

Este capítulo especifica las actividades realizadas en la fase de datos cualitativos. Las tres primeras secciones del capítulo corresponden a la etapa de preparación de los datos. Esta etapa inicia con la recolección de datos de tipo cualitativo, la cual es detallada en la Sección 5.1. Posteriormente, la Sección 5.2 explica el proceso de consolidación de los datos cualitativos. Seguido de esto, la Sección 5.3 especifica los procesos de depuración realizados en el conjunto de datos consolidado. Las siguientes tres secciones del capítulo corresponden a la etapa de transformación, la cual está enfocada en el método de análisis de contenido. En este sentido, en la Sección 5.4 se detalla el proceso de exploración de temas generales presentes en los datos de tipo cualitativo. Posteriormente, la Sección 5.5 y 5.6 describen los procesos de codificación abierta y axial, respectivamente. Finalmente, la última sección del capítulo corresponde a la etapa de análisis. En este orden de ideas, la Sección 5.7 expone la etapa de relativización de los datos, donde se entienden los resultados obtenidos en el contexto de la investigación. Esta última etapa también es conocida como la fase de codificación selectiva.

5.1. Recolección de datos cualitativos

En primer lugar, se define el diseño de investigación de la fase de datos cualitativos como un estudio fenomenológico, puesto que el objetivo es describir y entender los fenómenos desde la perspectiva de los participantes (Hernández-Sampieri *et al.*, 2014). La muestra de participantes corresponde a un subgrupo de los estudiantes de los cursos del conjunto de datos considerados en la fase de datos cuantitativos. Este subconjunto considera 17 grupos de los 22 cursos de Programación de Computadores. La muestra corresponde a estudiantes del curso programación de computadores entre el segundo semestre del 2019 al segundo periodo académico de 2020. La fuente de datos utilizada son encuestas de percepción de los estudiantes sobre el uso de la plataforma UNCode en la asignatura. Las encuestas fueron aplicadas por medio de *Google Forms* y las respuestas se almacenaron automáticamente en una hoja de cálculo por cada semestre donde fue aplicada la encuesta. En esta fase se tienen en cuenta solo las respuestas de las preguntas abiertas de la encuesta, puesto que los datos recolectados son de tipo cualitativo. Las preguntas abiertas de la encuesta complementan las preguntas cerradas consideradas en la fase de datos cuantitativos. En este sentido, todas las preguntas abiertas corresponden a un *¿Por qué?*, a partir del cual los estudiantes responden las razones por las cuales están de acuerdo o en desacuerdo con las afirmaciones de las preguntas cerradas. Las preguntas realizadas en cada semestre especificando los grupos donde se realizó la encuesta y el periodo académico son detalladas en la Tabla 5-1.

Tabla 5-1: Preguntas realizadas en las encuestas de percepción de los estudiantes con respecto a UNCode discriminadas por semestre

Semestre	Grupos	Afirmaciones preguntas cerradas	Nombre archivo
2019-1	pc-group8-2019-1 pc-group16-2019-1 pc-group15-2019-1	<ul style="list-style-type: none"> ▪ Considera que su conocimiento en el área de programación de computadores ha mejorado gracias a esta asignatura ▪ En general, UNCode fue útil en su proceso de aprendizaje 	Cuestionario UNCode 2019-1
2019-2	pc-group8-2019-2	<ul style="list-style-type: none"> ▪ Su conocimiento en el área de programación de computadores ha mejorado gracias a esta asignatura ▪ UNCode fue útil en su proceso de aprendizaje 	Cuestionario UNCode 2019-2
2020-1	pc-group12-2020-1 pc-group5-2020-1 pc-group6-2020-1 pc-group5-2020-1 pc-group8-2020-1	<ul style="list-style-type: none"> ▪ UNCode fue útil para aprender a programar computadores ▪ UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura ▪ La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas 	Encuesta UNCode 2020-1
2020-1	pc-group9-2020-1 pc-group12-2020-1 pc-group18-2020-1	<ul style="list-style-type: none"> ▪ UNCode fue útil para aprender a programar computadores ▪ UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura ▪ La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas 	Encuesta UNCode 2020-1_2
2020-2	pc-group6-2020-2 pc-group5-2020-2 pc-group8-2020-2 pc-group9-2020-2 pc-group12-2020-2	<ul style="list-style-type: none"> ▪ UNCode fue útil para aprender a programar computadores ▪ UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura ▪ La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas 	Encuesta UNCode 2020-2

5.2. Consolidación del conjunto de datos cualitativos

En primer lugar, se determina la ubicación de almacenamiento de la información obtenida en el proceso de recolección de datos cualitativos. De manera similar a la fase de datos cuantitativos, los datos cualitativos son almacenados en unidades compartidas de Google Drive. Las unidades compartidas son utilizadas, debido a que permiten manipular la información manera colaborativa con el equipo de trabajo y admite el acceso desde diferentes ubicaciones. En este orden de ideas, los cinco archivos con las respuestas de las encuestas de percepción especificados en la última columna de la Tabla 5-1, son almacenados en la carpeta Learning Analytics 1/Recolección/Encuestas. Teniendo en cuenta que los archivos almacenados se encuentran desagregados por semestre, se realiza una agrupación de las

respuestas recolectadas.

El proceso de agrupación se realiza por medio de un notebook de Google Colaboratory, puesto que esta herramienta facilita la exportación, manipulación e importación de datos dentro de unidades compartidas de Google Drive. En este sentido, el Open-Ended Questions - Data consolidation tiene como objetivo generar el consolidado con la información de los cinco archivos especificados en la Tabla 5-1. En primer lugar, en el notebook se eliminan los datos no relevantes para la fase cualitativa. Posteriormente, la información considerada en esta fase se limita a las columnas correspondientes al nombre de usuario del estudiante (*username*), el código del curso (*course_id*) y las respuestas a las preguntas abiertas. Adicionalmente, dentro del notebook se identifican las preguntas abiertas, especificadas en la tercera columna de la Tabla 5-1, que tienen la misma temática pero se diferencian ligeramente en la redacción del interrogante. Las preguntas agrupadas son codificadas acorde con la temática, con el fin de homogeneizar el conjunto de datos. Los grupos de preguntas con sus respectivos códigos asignados, son detallados en la Tabla 5-2. Finalmente, en el notebook se concatenan las respuestas de todos los semestres en un DataFrame, el cual tiene como columnas el nombre de usuario, el código del curso y los cuatro códigos de preguntas definidos en la primera columna de la Tabla 5-2. El DataFrame final es transformado y exportado como un archivo con extensión .csv, el cual es almacenado en la carpeta Learning Analytics 2/Consolidado/Encuestas.

5.3. Limpieza del conjunto de datos cualitativos

En este proceso de limpieza se seleccionan las preguntas del conjunto de datos consolidados que son relevantes para la investigación. El grupo de preguntas correspondientes al código RESPUESTA: Conocimiento asignatura no fue considerado, ya que los interrogantes tienen el objetivo de entender cómo la metodología de la asignatura aporta al conocimiento de los estudiantes y no está relacionada con el uso de UNCode directamente. Teniendo en cuenta que la metodología del proyecto es de tipo mixto, es necesario integrar el conjunto de datos cualitativos con la información utilizada en la fase cuantitativa. En este caso, la integración fue posible a los nombres de usuario de los estudiantes, los cuales deben corresponder en ambos conjuntos de datos con el fin de asegurar que la información utilizada corresponde a los mismos individuos.

La verificación es realizada por medio del notebook Open-Ended Questions - Data filtering, donde se identifican 39 errores de digitación en los nombres de usuario del listado de estudiantes en el conjunto de datos cualitativos, los cuales fueron corregidos. En contraste, las respuestas escritas de los alumnos no son depuradas de errores sintácticos ni semánticos, puesto que realizar estas correcciones puede modificar significativamente la interpretación de los datos en el análisis de contenido que se realizará en el siguiente proceso de esta fase. Sumado a esto, al DataFrame corregido se le agregan las calificaciones de los estudiantes en la asignatura, las cuales fueron calculadas en la fase de datos cuantitativos. Posteriormente, las calificaciones son categorizadas acorde con la nota mínima

Tabla 5-2: Códigos asignados a las preguntas consideradas en la fase de datos cualitativos

Código	Archivo	Pregunta cerrada	Pregunta abierta
RESPUESTA: Conocimiento asignatura	Cuestionario UNCode 2019-1	• Considera que su conocimiento en el área de programación de computadores ha mejorado gracias a esta asignatura.	¿Por qué?
	Cuestionario UNCode 2019-2	• Su conocimiento en el área de programación de computadores ha mejorado gracias a esta asignatura.	¿Por qué?
RESPUESTA: Proceso de aprendizaje	Cuestionario UNCode 2019-1	• En general, UNCode fue útil en su proceso de aprendizaje.	¿Por qué?
	Cuestionario UNCode 2019-2	• UNCode fue útil en su proceso de aprendizaje.	¿Por qué?
	Encuesta UNCode 2020-1	• UNCode fue útil para aprender a programar computadores.	¿Por qué?
	Encuesta UNCode 2020-1_2	• UNCode fue útil para aprender a programar computadores.	¿Por qué?
	Encuesta UNCode 2020-2	• UNCode fue útil para aprender a programar computadores.	¿Por qué?
RESPUESTA: Calificación automática	Encuesta UNCode 2020-1	• UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura.	¿Por qué?
	Encuesta UNCode 2020-1_2	• UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura.	¿Por qué?
	Encuesta UNCode 2020-2	• UNCode fue útil para obtener la calificación automática de los programas que usted desarrolló en esta asignatura.	¿Por qué?
RESPUESTA: Retroalimentación	Encuesta UNCode 2020-1	• La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas.	¿Por qué?
	Encuesta UNCode 2020-1_2	• La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas.	¿Por qué?
	Encuesta UNCode 2020-2	• La realimentación automática ofrecida por UNCode fue útil para saber cómo corregir los errores que presentaban mis programas.	¿Por qué?

aprobatoria de 3.0, con las etiquetas APROBADO o REPROBADO. La inclusión de las calificaciones a la codificación se realiza con el objetivo de buscar diferencias en el análisis de los datos entre el grupo de estudiantes que aprueban la asignatura y los que no aprueban. Finalmente, el DataFrame final con las columnas de nombre de usuario, código del curso, etiqueta de la calificación obtenida y las tres preguntas abiertas relacionadas con UNCode, es exportado como un archivo *csv* llamado *Qualitative_dataset*. El archivo final es almacenado en la carpeta compartida Learning Analytics 2/Filtrado/Encuestas.

5.4. Exploración de temas

A partir del conjunto de datos consolidado y depurado, se inicia con el análisis de contenido. En este caso, el análisis se realiza con el apoyo de la herramienta computacional NVivo, la cual facilita el proceso de organización, almacenamiento, codificación y análisis de los datos (Dhakal, 2022). Adicionalmente, el software NVivo permite la importación del archivo en formato de Excel del conjunto de datos cualitativos. El archivo contiene en total 348 respuestas realizadas por los estudiantes y considera preguntas abiertas relacionadas con los siguientes temas:

- Utilidad de la plataforma UNCode para generar aprendizaje en programación de computadores, nombrada RESPUESTA: Proceso de aprendizaje.
- Utilidad de la plataforma para obtener calificación automática de los programas construidos por los alumnos, nombrada RESPUESTA: Calificación automática.
- Utilidad de la retroalimentación automática ofrecida por UNCode para entender cómo corregir errores presentes en los programas desarrollados por los estudiantes, nombrada RESPUESTA: Retroalimentación.

Antes de realizar el análisis de los datos se determina la unidad de análisis básica, la cual corresponde a las respuestas textuales generadas por los estudiantes en cada una de las tres preguntas abiertas de la encuesta de percepción de UNCode. Posteriormente, se realiza una lectura repetitiva de las respuestas con el objetivo de identificar temas pre-concebidos y emergentes.

En primer lugar, con respecto a la pregunta RESPUESTA: Proceso de aprendizaje, algunas respuestas tratan sobre el conocimiento de errores y ejercicios estimulantes, las cuales pueden estar relacionadas con temas de procesos metacognitivos fomentados con el uso de UNCode. Adicionalmente, las respuestas a esta pregunta evidencian una relación cercana entre la mención de cumplimiento de objetivos y la referencia a la herramienta Python Tutor. También se observa en las respuestas el tema de habilidades de programación, el cual puede ser desglosado en diferentes tipos de habilidades que pueden corresponder a diferentes categorías. Por otra parte, algunos estudiantes hacen referencia a la práctica constante, lo que da indicios de procesos de consolidación de conocimientos por medio de la plataforma. Sumado a esto, es mencionada la inflexibilidad de las comprobaciones disponibles en la plataforma, lo que posiblemente puede guiar hacia una duda de la objetividad de la plataforma e incluso puede indicar temáticas de frustración o molestia por parte de los estudiantes.

En segunda instancia, con respecto a la pregunta RESPUESTA: Calificación automática, se evidencian respuestas que resaltan la inmediatez de la plataforma como un aspecto positivo. Adicionalmente, algunos estudiantes relacionan el aspecto automático de la plataforma con el concepto de objetividad. En este orden de ideas, las menciones de inmediatez y objetividad en la calificación pueden indicar una temática general de características positivas que optimizan la evaluación sumativa de las soluciones construidas por los estudiantes. Por otra parte, una temática emergente es la relacionada

con las pruebas disponibles para comprobar la calidad de los programas desarrollados; donde existe una dicotomía en la que un grupo de respuestas resaltan la inflexibilidad en la comprobación de los programas, pero otros estudiantes mencionan la rigurosidad en los planteamientos como una característica positiva.

Finalmente, en relación con la pregunta RESPUESTA: Retroalimentación, se hace referencia en algunas respuestas a una carencia en la claridad tanto de las instrucciones iniciales de los ejercicios como de las descripciones de los errores. Sumado a esto, las respuestas afirman que conocer la ubicación del error dentro del código construido debe ser complementada con información sobre estrategias de corrección. También se evidencia que las respuestas sobre la utilidad del señalamiento de errores están condicionadas por el grado de familiaridad del estudiante, no solo con las herramientas de la plataforma sino también con el significado de los veredictos. Sin embargo, es posible que la utilidad del señalamiento de errores dependa también de las habilidades de los alumnos con respecto a búsqueda activa de información complementaria u otras aptitudes analíticas. En conclusión, la mención de falta de claridad y vaguedad en la información suministrada por la plataforma, puede indicar presencia de temáticas relacionadas con la insuficiencia de la retroalimentación.

5.5. Codificación abierta

Partiendo del bosquejo de temas resultado del examen de las unidades de análisis, se inicia el primer nivel de codificación. En este sentido, a cada unidad es asignado uno o varios códigos y categorías. Teniendo en cuenta que el proceso de codificación abierta y axial no es un proceso estrictamente secuencial, la generación de códigos y categorías se puede traslapar con la identificación de temas generales. En este caso, la carencia de detalle en algunas unidades de análisis no permiten asignar un código o categoría específica, en consecuencia estas unidades son ubicadas en temas generales, sin categoría ni código asignado. En este sentido, para cada una de las tres preguntas consideradas de la encuesta de percepción, son listadas y descritas las categorías y códigos identificados.

5.5.1. Pregunta proceso de aprendizaje

La Tabla 5-3 especifica la cantidad de unidades de análisis agrupadas por código, categoría y temática general, las cuales están clasificadas por el desempeño académico de los estudiantes. Cabe resaltar que, las respuestas de los estudiantes pueden ser etiquetadas con más de un código, categoría o tema, es decir, que la codificación pueden contener unidades de análisis comunes; por esta razón, la suma de referencias en la tabla no corresponde con el número de respuestas consideradas. En el caso de la pregunta relacionada con la utilidad de la plataforma en el proceso de aprendizaje de los estudiantes, se generaron en total 15 códigos, 21 categorías y 6 temas. A continuación, se presenta la definición y ejemplificación de cada una de las categorías identificadas junto con los códigos asignados correspondientes.

Tabla 5-3: Cantidad de unidades de análisis agrupadas por códigos, categorías y temas

Tema	Categoría	Código	Unidades de análisis		
			Alumnos aprobados	Alumnos reprobados	Total
Ambiente de la plataforma			13	1	14
			6	0	6
	Casos de prueba		9	1	10
Bondades de la plataforma	Consejos formativos		21	0	21
		Orienta las mejoras implementables	13	0	13
	Conocer los errores		80	2	82
	Disponibilidad en línea		8	1	9
	Espacio de trabajo		4	0	4
	Facilidad de uso		32	0	32
		Facilidad para escribir y corregir código	13	0	13
	Lenguajes de programación		5	0	5
	Práctica constante		17	3	20
Google Colaboratory			9	1	10
			33	0	33
Herramientas de UNCode	Custom input		30	1	31
	Linter		29	0	29
	Python Tutor		69	4	73
			1	0	1
Logros pedagógicos	Aprendizaje autónomo		13	0	13
	Ejercicios estimulantes		39	1	40
			10	0	10
	Evaluación optimizada	Calificación inmediata	22	0	22
		Calificación objetiva	15	0	15
		Planteamiento de problemas y ejercicios	15	0	15
		Presentación de actividades académicas	26	5	31
	Habilidades de programación		30	2	32
	Resolución de problemas		66	1	67
			45	0	45
Mejorable para aprender a programar		Calificación ineficiente	4	0	4
		Falla casos de prueba	5	0	5
	Fallas generales UNCode	Herramientas no disponibles	6	1	7
		Incompatibilidad	5	0	5
		Pérdida de información	3	0	3
		Plataforma caída	14	2	16
		Registro	0	1	1
		Velocidad de procesamiento	5	2	7
		Visualización	5	1	6
	Inflexibilidad en las comprobaciones		13	2	15
	Falla en el objetivo educativo		6	5	11
	Retroalimentación insuficiente		10	1	11
	Herramienta sustituible		1	1	2

Casos de prueba: Dentro de esta categoría se encuentran las respuestas de los estudiantes que hacen referencia a los casos de prueba de ejemplo que se encuentran integrados en la plataforma. En general, los alumnos resaltan que estos casos de prueba ayudan a obtener retroalimentación sobre el programa enviado e identificar errores. La Tabla 5-4 detalla algunas de las unidades de análisis que representan esta categoría.

Tabla 5-4: Ejemplos de unidades de análisis asignadas a la categoría casos de prueba

Unidad de análisis
“UNCode fue durante el semestre una herramienta de gran apoyo donde si visualizacion y modo de mostrar cada caso de prueba ayudaba a ver cuales eran los errores y por lo tanto que se debia corregir ...”
“Porque es una plataforma interesante que además permite conocer los errores en casos de prueba y califica de forma justa”
“... y también permite ver algunos casos de prueba para con las entradas y las salidas para encontrar errores ”

Consejos formativos: Categoría asignada a las unidades donde se menciona la utilidad de la retroalimentación formativa ofrecida por la plataforma, la cual está basada en sugerencias sobre el proceso de construcción del código en términos de sintaxis, semántica, eficiencia y aspectos de mantenibilidad. Las referencias en esta categoría resaltan las sugerencias de buenas prácticas de programación, los cuales en algunos casos orientan sobre el procedimiento a seguir para mejorar los programas diseñados por el estudiante. La Tabla 5-5 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-5: Ejemplos de unidades de análisis asignadas a la categoría consejos formativos

Unidad de análisis
“... me permitía ver en poco tiempo si mi código estaba mal permitiendo corregirlo y a su vez dando consejos de como escribir adecuadamente un código ”
“... permite ver cuales son los casos en que mi codigo no funciona, y aporta sugerencias para abordar el problema . De esta manera es posible corregir los ejercicios y aprender la forma correcta de solucionarlos ...”
“Porque permitía una retroalimentación inmediata y daba ciertas sugerencias que ayudaban a encontrar los bugs en el programa . Además que, en muchos programas, lo obligaba a uno a hacerlos de la manera más óptima posible, permitiéndonos pensar en una solución que tomará menos tiempo”
“Es una gran herramienta que permite ver paso a paso el programa en ejecución, además de dar sugerencias para mejorar las buenas prácticas de programación ”

Conocer los errores: Categoría asignada a las unidades que mencionan la retroalimentación de la plataforma que permite al estudiante identificar y analizar errores específicos en los programas construidos. Adicionalmente, se evidencia que muchos estudiantes utilizan este recurso como punto de partida para el proceso de corrección y refinamiento de la solución diseñada, fomentando la continuación del proceso de aprendizaje. La Tabla 5-6 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-6: Ejemplos de unidades de análisis asignadas a la categoría conocer los errores

Unidad de análisis
“... pues permite darse cuenta de errores que pudieron pasar desapercibidos”
“... en algunas ocasiones pude evidenciar mis errores viendo los ejercicios en los que fallaba ...”
“Es una herramienta que ofrece herramientas muy útiles a la hora de programar, permite ver los errores que cometió al estudiante además de ofrecer alternativas para solucionar los mismos ...”
“La plataforma de uncode me pareció muy buena ya que te informa claramente de los errores que cometes y donde lo cometiste, cosa que es muy útil para saber donde corregir el código y que es lo que no funciona ...”

Disponibilidad en línea: Categoría asignada a las unidades donde se resalta el beneficio que tiene para los estudiantes el hecho de que la plataforma funcione mediante un navegador web, en contraste con programas de ejecución en entornos locales. Las percepciones en esta categoría dan la sensación de que UNCode amplía los alcances de las sesiones del curso, ofreciendo un auténtico entorno de aprendizaje en el que se pueden desplegar y poner a prueba los conocimientos adquiridos. La Tabla 5-7 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-7: Ejemplos de unidades de análisis asignadas a la categoría disponibilidad en línea

Unidad de análisis
“Excelente debido a que se trabaja online . Y creo que esto es fundamental debido a que la comunidad universitaria la gran mayoría no cuenta con PC de última generación”
“Como plataforma virtual ayuda mucho si no tienes un editor de código en el PC”
“Por que nos brindaba una herramienta para programar completamente gratuita, evitando que tuviéramos que descargar programas para programar ...”
“por que es una plataforma que permite usar python sin necesidad de descargarlo , facil de usar, y al permitir guardar todo en el drive es muy conveniente”

Espacio de trabajo: Esta categoría engloba las respuestas de los estudiantes donde se identifica como aspecto positivo de la plataforma el hecho de que ofrece un espacio específico para la organización de actividades, trabajos y desarrollos realizados en el curso. La Tabla 5-8 detalla algunas de las unidades de análisis que representan esta categoría.

Tabla 5-8: Ejemplos de unidades de análisis asignadas a la categoría espacio de trabajo

Unidad de análisis
“Por que facilita al estudiante tener un lugar de trabajo académico especial, además las herramientas de compilado ...”
“Podíamos llevar nuestros avances, testear nuestros codigos, mejor organización para las actividades ”
“Es una herramienta que se adapta a las necesidades de los estudiantes... ”

Facilidad de uso: Categoría asignada a las unidades donde se resalta como atributo de la plataforma su versatilidad, expresado en términos de que la interacción del estudiante con la herramienta no representa mayores dificultades. Dentro de esta categoría se identifica un grupo de referencias que se enfocan en resaltar la facilidad para realizar actividades específicas de la construcción y modificación del programa directamente desde la plataforma, las cuales son codificadas como *Facilidad para escribir y corregir código*. Las Tablas 5-9 y 5-10 muestran algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-9: Ejemplos de unidades de análisis asignadas a la categoría facilidad de uso

Unidad de análisis
“... es muy útil, por sus herramientas y la facilidad de usarlo , para realizar las distintas actividades propuestas para la clase.”
“Porque es online y es fácil de usar , además gracias al juez virtual y los casos de prueba es más fácil entender cómo funciona cada ejercicio.”
“Es una plataforma muy fácil de usar , con muchas herramientas que hacen fácil el aprendizaje.”
“Es una herramienta fácil de usar , sencilla y útil a la hora de practicar, interactiva también”
“Menú de navegación simple y fácil de entender ”

Tabla 5-10: Ejemplos de unidades de análisis asignadas a facilidad para escribir y corregir código

Unidad de análisis
“Tener una herramienta que facilite escribir código y recibir una retroalimentación del mismo permite entender lo que uno esta haciendo y como lo esta haciendo ...”
“Respecto a otros IDE que toca iniciar de cero para probar el codigo, UNCode facilita esto de modo que no toque escribir las entradas cada vez que corre el código ...”
“... el contar con la herramienta visualize your code es de mucha ayuda ya que permite saber como se está comportando el código que se esta escribiendo, y corregir más fácilmente los errores.”
“Fue útil para realizar los talleres designados por el profesor, escribir el código en la página directamente lo hace más fácil ... ”
“Es una buena forma de calificar ya que da una respuesta rápida y permite facilidad al escribir y corregir código ”

Lenguajes de programación: Categoría asignada a las unidades donde se menciona la funcionalidad de UNCode que permite seleccionar diversos lenguajes de programación como Python, C++ y Java. La Tabla 5-11 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Práctica constante: Categoría asignada a las unidades que reportan que la plataforma permite practicar y realizar ejercicios frecuentemente, lo que permite fortalecer y consolidar habilidades de programación adquiridas en las clases. La Tabla 5-12 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-11: Ejemplos de unidades de análisis asignadas a la categoría lenguajes de programación

 Unidad de análisis

“Es una herramienta muy útil para practicar los **distintos lenguajes de programación ...**”

“ya que proporcionaba todas las herramientas necesarias para practicar y mejorar las habilidades básicas (y esenciales) de programación **tanto en c++ como en python 3**”

“Mediante UNCode, pude realizar muchos ejercicios que me permitieron profundizar y poner a prueba mis conocimientos de **los lenguajes que estaba aprendiendo ...**”

Tabla 5-12: Ejemplos de unidades de análisis asignadas a la categoría práctica constante

 Unidad de análisis

“Debido a que es una buena herramienta para **practicar y realizar ejercicios ...**”

“UNCode fue muy útil para aprender a programar computadores porque los **ejercicios** que allí se tienen sirven para **practicar** y afianzar los conocimientos que se imparten en la clase ...”

“... Durante todo el semestre, lo usé no solo para los talleres y quizzes, sino también para **probar y ejecutar códigos** para otros usos ...”

“A través de los **ejercicios propuestos** en la herramienta UNCode se **pone a prueba** lo aprendido en las clases de programación, ayudando a validar los temas vistos con la **práctica que conlleva realizar ejercicios**”

Custom input: Categoría asignada a las unidades donde se hace referencia a la herramienta custom input, la cual permite a los estudiantes realizar pruebas personalizadas del código programado. La Tabla 5-13 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-13: Ejemplos de unidades de análisis asignadas a la categoría Custom input

 Unidad de análisis

“Porque permite **probar el código**, hacer el seguimiento de su ejecución y aconseja sobre buenas prácticas de programación o errores que detecta antes de la ejecución del mismo ...”

“En primer lugar, la plataforma permite **testear el código** antes de ser enviado para su evaluación, lo que permite corregir los fallos ...”

“El hecho de **probar** y visualizar **el código** en la plataforma permite una reflexión muy efectiva acerca de los errores que en algún momento cada persona pueda tener y aprender de ellos ...”

“... También permite **probar el código** y colocar de antemano lo que queremos colocar, para no tener que estar repitiendo el tedioso proceso de poner las entradas.”

Linter: Categoría asignada a las unidades que mencionan la herramienta de resaltado de errores sintácticos y de estilo en código fuente. La Tabla 5-14 muestra algunos ejemplos.

Python Tutor: Categoría asignada a las unidades que hacen referencia a la herramienta integrada en la plataforma que permite la visualización paso por paso de la ejecución del programa construido. La Tabla 5-15 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-14: Ejemplos de unidades de análisis asignadas a la categoría Linter

Unidad de análisis
“... y también en ocasiones muestra las líneas de código que contienen error , sería bueno que la plataforma mostrara el procedimiento que realiza el código ...”
“Es una herramienta que ofrece herramientas muy útiles a la hora de programar, permite ver los errores que cometió al estudiante además de ofrecer alternativas para solucionar los mismos ...”
“... y daba ciertas sugerencias que ayudaban a encontrar los bugs en el programa. Además que, en muchos programas, lo obligaba a uno a hacerlos de la manera más óptima posible, permitiéndonos pensar en una solución que tomará menos tiempo.”

Tabla 5-15: Ejemplos de unidades de análisis asignadas a la categoría Python Tutor

Unidad de análisis
“Tenía la opción de visualizar la ejecución de mi código paso a paso , lo cual me ayudaba encontrar las debilidades de este.”
“porque UNcode al tener tantas herramientas a disposición, como Python Tutor , me permitieron ver como funcionan mis programas y a su vez detectar los errores que estaba cometiendo para corregirlos.”
“... además de una herramienta que nos permite ver el proceso del código paso por paso ”
“... y probar el código paso a paso , que son necesarias cuando no se entiende del todo alguna parte desarrollada”

Aprendizaje autónomo: Categoría asignada a las unidades donde se hace referencia a condiciones de la plataforma que promueven procesos de aprendizaje donde la intervención del docente o el monitor es mínima o no existe. En este sentido, este grupo de referencias indican que el uso de UNCode permite que el aprendizaje se extienda más allá de los espacios compartidos en las clases. La Tabla 5-16 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-16: Ejemplos de unidades de análisis asignadas a la categoría aprendizaje autónomo

Unidad de análisis
“Al utilizar esta plataforma evaluativa el estudiante podía conocer los errores en los procesos que se estaban efectuando, asimismo cómo se estaba ejecutando cada programa. Por lo cual se podía hacer una autocorrección y de la misma forma mejorar ”
“Uncode es una herramienta muy completa que motiva a los estudiantes a seguir programando incluso cuando se tienen bastantes falencias sobre un tema en específico.”
“Me permitía ver los errores en mis códigos sin necesidad de tener un profesor presente ”
“Plantea ejercicios interesantes y permite visualizar los códigos, además que le permite al estudiante el manejo de sus tiempos ”

Ejercicios estimulantes: Categoría asignada a las unidades que hacen referencia los ejercicios propuestos en UNCode como problemas de programación lo suficientemente exigentes como para poner a prueba y afianzar las habilidades y conocimientos adquiridos, sin desmotivar a los estudiantes proponiendo un nivel de dificultad que supera el alcance del curso. La Tabla 5-17 muestra algunos ejemplos.

Tabla 5-17: Ejemplos de unidades de análisis asignadas a la categoría ejercicios estimulantes

Unidad de análisis
“A través de los ejercicios propuestos en la herramienta UNCode se pone a prueba lo aprendido en las clases de programación, ayudando a validar los temas vistos con la práctica que conlleva realizar ejercicios”
“Mediante UNCode, pude realizar muchos ejercicios que me permitieron profundizar y poner a prueba mis conocimientos de los lenguajes que estaba aprendiendo ...”
“... Sin embargo, los ejercicios eran útiles para ahondar en los temas. ”
“Algunos de los casos “especiales” que evalúa UNcode me llevo a ir mas allá de simplemente construir mi programa , pensar alguna solución específica para esos casos e interpretar de mejor forma los ejercicios.”

Evaluación optimizada: Categoría asignada a las unidades donde se afirma que el uso de la plataforma hace mucho más sencilla, rápida y objetiva la evaluación de los programas diseñados y enviados por los estudiantes. En esta categoría encontramos cuatro grupos de referencias. En primer lugar, los estudiantes que resaltan la importancia de la inmediatez en la calificación de los intentos de solución enviados de las actividades realizadas en la plataforma, los cuales se les asigna el código *Calificación inmediata*. En segunda instancia, otro grupo de respuestas se enfocan en mencionar la objetividad de la calificación obtenida en la plataforma, puesto que se evita la subjetividad de un calificador manual. Este grupo de unidades es codificado como *Calificación objetiva*. El tercer conjunto de unidades corresponde al código *Planteamiento de problemas y ejercicios*, donde se hace referencia a la optimización del proceso de construcción de ejercicios en la plataforma, lo que hace más fácil para el estudiante comprender el contexto e instrucciones de los problemas de programación planteados. El último conjunto de referencias es agrupado con el código *Presentación de actividades académicas*, en el cual la optimización de la evaluación se evidencia en términos de la simplificación del proceso de carga y envío de soluciones de las actividades académicas como talleres o exámenes. A partir de la Tabla 5-18 hasta la Tabla 5-21 se muestran algunos ejemplos de unidades de análisis asignadas a cada código.

Tabla 5-18: Ejemplos de unidades de análisis asignadas al código calificación inmediata

Unidad de análisis
“Es muy conveniente el poder recibir una calificación casi inmediata , pues permite darse cuenta de errores que pudieron pasar desapercibidos”
“Es una herramienta muy útil ... y al dar las calificaciones inmediatamente , da una retroalimentación más efectiva.”
“Porque permitía una retroalimentación inmediata y daba ciertas sugerencias que ayudaban a encontrar los bugs en el programa.”
“Ya que se nos entregan resultados inmediatos , recalando en los errores ...”

Habilidades de programación: Categoría asignada a las unidades donde los estudiantes identifican que el uso de la plataforma favorece la adquisición de importantes habilidades de programación; en este sentido, UNCode se configura como un entorno significativo en el proceso de aprendizaje. Las habilidades mencionadas por los alumnos, trascienden el manejo puramente técnico de las herramientas o lenguajes de programación hacia aptitudes auténticamente profesionales que en el

Tabla 5-19: Ejemplos de unidades de análisis asignadas al código calificación objetiva

Unidad de análisis
“... me parece que el factor del juez agiliza mucho más las cosas, la evaluación es mucho más objetiva y rápida, en lugar de que el docente o monitor deba revisar todos los códigos ...”
“Porque es una plataforma interesante que además permite conocer los errores en casos de prueba y califica de forma justa ”
“Es positivo el hecho de tener una calificación tan rigurosa de los ejercicios”

Tabla 5-20: Ejemplos de unidades de análisis asignadas al código facilita el planteamiento de problemas y ejercicios

Unidad de análisis
“El planteamiento de los ejercicios es claro y el debugger es una herramienta muy util para encontrar errores”
“Por la facilidad en plantear problemas y ejercicios por parte de los profesores”
“Es una herramienta bastante útil para reforzar los conocimientos y poner a prueba el estudiante ante ejercicios bastante específicos y con muchos casos de pruebas”
“La explicación de los enunciados, las entradas y las salidas esperadas , me ayudo mucho a lo largo del semestre para entender los distintos ejercicios ...”

mediano o largo plazo pueden convertirse en diferenciales de la formación recibida. Dentro de estas habilidades de programación destacan el pensamiento algorítmico, y el entendimiento de la lógica de la programación como un proceso secuencial y sistemático. Estas habilidades exigen un mayor grado de planificación y visualización entre un objetivo particular y las herramientas disponibles. La Tabla 5-22 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Resolución de problemas: Categoría asignada a las unidades que resaltan la manera en que la plataforma facilita la comprobación de programas construidos como soluciones a los ejercicios o actividades académicas. Las verificaciones se realizan mediante la ejecución de múltiples pruebas o comparación con resultados esperados, los que permite entender el funcionamiento del programa diseñado y alimenta las habilidades para replicarlo y aplicarlo en otros casos con condiciones distintas. La Tabla 5-23 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-21: Ejemplos de unidades de análisis asignadas a presentación de actividades académicas

Unidad de análisis
“ Facilito el envío de talleres , además corrige los códigos de forma inmediata”
“Es una interesante modalidad para la realización de las actividades y tareas , de las cuales son comprobadas por varios casos de prueba para identificar si el programa cumple con los requisitos.”
“fue una herramienta útil para hacer la entrega de las actividades ”
“En realidad cuando estudio o similares no programo en uncode, solo lo uso para envíar las tareas ... ”

Tabla 5-22: Ejemplos de unidades de análisis asignadas a la categoría habilidades de programación

Unidad de análisis
“... así mismo tener varios casos de prueba nos ayuda a pensar el algoritmo de una manera funcional y optimizada. ”
“... y también nos permite identificar de forma clara el algoritmo de nuestros programas. ”
“Por que gracias a los ejercicios puestos en la plataforma pude desarrollar un pensamiento lógico y resolver problemas de acuerdo a los temas tratados en clase.”
“La explicación de los enunciados, las entradas y las salidas esperadas, me ayudo mucho a lo largo del semestre para entender los distintos ejercicios, tanto a nivel de lengua materna como a nivel de la lógica de programación ”

Tabla 5-23: Ejemplos de unidades de análisis asignadas a la categoría resolución de problemas

Unidad de análisis
“Porque me permitió comprobar si el código escrito para solucionar un problema determinado y su ejecución eran las indicadas para el problema. ”
“ facilita la solucion de problemas con su modulo de test code y python tutor”
“... También fue muy útil para identificar problemas y plantear diferentes soluciones según fuesen requeridas”
“Fue útil porque al poder obtener una retroalimentación inmediata del funcionamiento del código con base en los resultados obtenidos y los esperados permite corregir errores de manera progresiva y sobre la marcha para avanzar hacia la solución correcta. ”

Fallas generales UNCode: Categoría asignada a las unidades donde se reportan fallas y errores en el funcionamiento general de la plataforma. En otras palabras, las respuestas consideradas en esta categoría resaltan aspectos particulares de la plataforma que son identificados como problemáticos, los cuales merecen ser analizados puesto que es posible que estén interfiriendo con el proceso de aprendizaje de los estudiantes. La Tabla 5-24 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-24: Ejemplos de unidades de análisis asignadas a la categoría fallas generales UNCode

Unidad de análisis
“... es una plataforma que a pesar de los problemas que aún tiene , es muy útil, por sus herramientas y la facilidad de usarlo ...”
“Puesto que, a pesar de tener pequeños errores , fue una gran guía para hacer y corregir códigos según los problemas ...”
“la plataforma cumple en cuanto a sus servicios pero no de la forma mas adecuada ”
“Siento que la herramienta tiene fallas , personalmente solo lo utilice para subir el código y comprobar si pasaba o no”

Dentro de esta categoría se identifican once códigos con respecto a las fallas señaladas en las unidades de análisis. En primer lugar, algunos estudiantes reportan errores o incoherencias en las calificaciones obtenidas, puesto que la retroalimentación sumativa no corresponde con la calidad del programa construido. Este grupo de respuestas es codificado como *Calificación ineficiente*. Por otra parte, las unidades de análisis codificadas como *Fallos en casos de prueba* se concentran en señalar la ejecución incorrecta de los casos de prueba de los ejercicios planteados en la plataforma, lo que no permite que los estudiantes obtengan retroalimentación formativa de manera efectiva. El siguiente código asignado es *Herramientas no disponibles*, donde se resaltan los fallos en el funcionamiento de

herramientas específicas de la plataforma, lo que genera obstáculos en el proceso de construcción de programas y corrección de código. Otro código definido corresponde a *Incompatibilidad*, donde son ubicadas las respuestas donde se señala la incompatibilidad de los programas desarrollados en UNCode con otras plataformas de verificación de código y viceversa. Adicionalmente, algunos individuos reportan la pérdida ocasional de información al realizar ciertas actividades dentro de la plataforma, este conjunto de respuestas es clasificado como *Perdida de información*. Un conjunto considerable de participantes reporta como principal inconveniente el constante bloqueo del acceso a la plataforma debido al fallo de los servidores, los cuales son etiquetados con el código *Plataforma caída*. En contraste, se identifica una unidad de análisis relacionada con la falta de claridad e inconvenientes en el proceso de registro en UNCode, la cual es codificada como *Registro*. Sumando a esto, en las unidades de análisis se identifican reportes de demoras significativas en el procesamiento de los archivos cargados en la plataforma como solución a las actividades del curso, las cuales son identificadas con el código *Velocidad de procesamiento*. Finalmente, el código *Visualización* corresponde a las respuestas que mencionan fallas en la interfaz de visualización tanto del programa ejecutado como de los casos de prueba, lo que no permite la adquisición de información relevante para el proceso de aprendizaje de los estudiantes. A partir de la Tabla 5-25 hasta la Tabla 5-33 se muestran algunos ejemplos de unidades de análisis asignadas a cada código.

Tabla 5-25: Ejemplos de unidades de análisis asignadas al código calificación ineficiente

Unidad de análisis
“Al principio Uncode presentaba bastantes fallos y otras veces era muy ineficiente a la hora de calificar ”
“Porque en ocasiones marcaba 100 % funcional algunos programas y cuando los usaba para otra actividad estos no se desempeñaban de la manera correcta ”
“La herramienta muchas veces da errores aún cuando la salida requerida es exactamente a como la pide ... ”
“porque muchas veces el código que pedía, lo hacía y este cumplía el objetivo pero el uncode siempre me aparecía como error ”
“La evaluación de cada uno de los programas resultaba un poco estresante, ya que muchas veces el error no era del código en sí, sino de la manera de calificar en el interior de la plataforma. ”

Tabla 5-26: Ejemplos de unidades de análisis asignadas al código casos de prueba

Unidad de análisis
“... es bastante útil conocer los errores en el programa con los casos de prueba, pero no siempre fueron dados ... ”
“Al principio Uncode presentaba bastantes fallos, a veces no ejecutaba bien los casos de prueba ... ”
“... Sin embargo, la plataforma no siempre me decía el tipo de error que había cometido ...”
“Aunque hay problemas en algunos casos con las salidas mostradas por la prueba y el visualizador ...”

Inflexibilidad de las comprobaciones: Categoría asignada a las unidades que resaltan el rigurosidad excesiva y carencia de flexibilidad de la plataforma al momento de validar las soluciones

Tabla 5-27: Ejemplos de unidades de análisis asignadas al código herramientas no disponibles

Unidad de análisis

“En ocasiones fallaba y durante medio semestre **no permitía usar las herramientas** ni observar los errores del código al comparar respuestas.”

“... en algunas ocasiones pude evidenciar mis errores viendo los ejercicios en los que fallaba, mas no estoy totalmente de acuerdo porque **al principio no se podía acceder a esa opción**, y se me dificultó más evidenciar la equivocación.”

“Porque nos permitía conocer nuestros errores y verificar el código, pero muchas veces **las herramientas como Python Tutor o Test Code no funcionaban correctamente**, por lo que tocaba usar otro entorno para ejecutar y probar el código”

“Aunque posee herramientas de apoyo es difícil asegurar que ayude en el proceso de aprendizaje dada la **presencia de problemas** en la pagina y **de los apoyos que posee esta**. Problemas como caída de la pagina y la **no funcionalidad de algunos servicios con el probador de ejercicios y el Python tutor** .”

Tabla 5-28: Ejemplos de unidades de análisis asignadas al código perdida de información

Unidad de análisis

“el semestre pasado tuve problemas con uncode, **se me borro todo el progreso académico** y el profesor no habia guardado notas”

“Es una muy buena herramienta sin embargo en ocasiones se cae y **se pierden algunos trabajos**”

“... eso si personalmente de vez en cuando no registra el submit y **los datos se borran al cerrar la pestaña**.”

Tabla 5-29: Ejemplos de unidades de análisis asignadas al código registro

Unidad de análisis

“... sin embargo, a la hora de **registrarse para el curso puede llegar a ser un poco enredado** por el hecho de no saber si hay una cuenta creada o si hay que crear una y de como se tiene que hacer el registro”

Tabla 5-30: Ejemplos de unidades de análisis asignadas al código plataforma caída

Unidad de análisis

“... Sin embargo, hablando no de los jueces sino de UNCode específicamente, debe mejorar bastante, a veces los estudiantes no podíamos trabajar en los talleres porque **la plataforma estaba caída** o tenía errores que resultaban parecer obstáculos para nosotros.”

“Es una muy buena herramienta sin embargo **en ocasiones se cae** y se pierden algunos trabajos”

“... Sin embargo, la plataforma no siempre me decía el tipo de error que había cometido, **se “caía con frecuencia** y había situaciones donde no era acorde a otras plataformas que tenían el mismo objetivo que UNCode, por lo que podía confundirme aún más de lo que me ayudaría”

“... El único inconveniente es que **la página se cae constantemente**.”

construidas por los estudiantes. En específico, se hace referencia a casos donde la plataforma calificaba como incorrectos algunos programas que cumplen con el objetivo del ejercicio o problema, pero poseen errores menores de forma. La Tabla 5-34 muestra algunos ejemplos de esto.

Falla en el objetivo educativo: Categoría donde se agrupan las unidades de análisis donde los estudiantes afirman que la plataforma no representa una herramienta significativa en el proceso de

Tabla 5-31: Ejemplos de unidades de análisis asignadas al código replicar códigos

Unidad de análisis
“En realidad cuando estudio o similares no programo en uncode, solo lo uso para envíar las tareas y si se programa en Colab, al copiar y pegar de Colab a Uncode se indenta mal y perdí un quiz por eso”
“... había situaciones donde no era acorde a otras plataformas que tenían el mismo objetivo que UNCode , por lo que podía confundirme aún más de lo que me ayudaría”

Tabla 5-32: Ejemplos de unidades de análisis asignadas al código velocidad de procesamiento

Unidad de análisis
“... Lo único molesto era la velocidad para procesar las respuestas. ”
“Es muy lento al momento de hacer las entregas ”
“Si bien la aplicación cumple con su propósito, presenta varias fallas (Como las varias caídas de la pagina, o la lentitud en ciertas ocasiones) no permiten que el usuario tenga una experiencia amistosa con UNCode”
“Aunque es cierto que mostraba los casos de prueba, el tiempo era demasiado grande para su obtención.”

Tabla 5-33: Ejemplos de unidades de análisis asignadas al código visualización

Unidad de análisis
“Tiene bastantes fallas , tales como al visualizar la situación o el programa siendo ejecutado, la interfaz no permite ver el desarrollo para programas largos. O la clásica demora para testear o dar submit en el programa”
“Aunque hay problemas en algunos casos con las salidas mostradas por la prueba y el visualizador , es práctico forma de calificación automática de la plataforma”
“Porque en muchas ocasiones tenia errores en los enunciados, para estos momentos no esta corriendo bien y no brinda las herramientas de visualización como antes.”

Tabla 5-34: Ejemplos de unidades de análisis asignadas a la categoría inflexibilidad de las comprobaciones

Unidad de análisis
“Debido a su corto campo de aceptación de programas , evitando crear múltiples opciones para solucionar un ejercicio.”
“Tiende a fallar mucho en tiempos de ejecución y es una herramienta poco flexible. ”
“Es un buen programa, lo único es que es algo complicado de usarlo, puesto que las salidas de los códigos, deben seguir una estructura y orden específicos. ”
“... tiene un pequeño contra que es lo reducido de las salidas , aunque el código haga lo que se pide en el problema propuesto por cualquier carácter adicional hace que todo el test que no coincida este del todo mal”

aprender a programar computadores, ni favorece la adquisición de habilidades de programación. La Tabla 5-35 muestra algunos ejemplos.

Retroalimentación insuficiente: Categoría asignada a las unidades donde se señala que la retroalimentación de la plataforma no es lo suficientemente puntual, objetiva y/o detallada como para favorecer el aprendizaje de habilidades de programación de computadores. En este sentido, muchos

Tabla 5-35: Ejemplos de unidades de análisis asignadas a la categoría falla en el objetivo educativo

Unidad de análisis
“UNCode fue una herramienta bastante útil a la hora de presentar los talleres y quices, pero en sí, esta no fue útil para aprender acerca de la programación ... ”
“En general me parece una herramienta que es util para evaluar ejercicios de programacion, al momento de aprender no hace mucha diferencia ”
“Es un buen mecanismo para practicar y comprender mejor de programación, sin embargo no siento que su propósito sea garantizar el aprendizaje , porque no es una plataforma que explique errores o explique temas”
“Aunque posee herramientas de apoyo es difícil asegurar que ayude en el proceso de aprendizaje dada la presencia de problemas en la pagina y de los apoyos que posee esta ...”

estudiantes mencionan tener una perspectiva de la plataforma como una herramienta confusa y poco confiable. La Tabla 5-36 muestra algunos ejemplos.

Tabla 5-36: Ejemplos de unidades de análisis asignadas a la categoría insuficiencia de la retroalimentación

Unidad de análisis
“Para tener la capacidad de construir el conocimiento se debe obtener un Feedback puntual y objetivo , cosa que no ofrece la plataforma. ”
“La retroalimentación que da cuando hay algo incorrecto, resulta muy útil al momento de corregir el código, aunque a veces (como en el caso del taller 10) no era tan obvia la comparación entre lo que esta bien, y lo que estaba mal ”
“... Sin embargo, la plataforma no siempre me decía el tipo de error que había cometido , se “caía” con frecuencia y había situaciones donde no era acorde a otras plataformas que tenían el mismo objetivo que UNCode...”
“UNcode en algunos caso ayudó con el desarrollo pero creería que la retroalimentacion por parte de la página debería ser mayor ”

Herramienta sustituible: En esta categoría se resalta la posibilidad de reemplazar las funciones de UNCode con otras herramientas o programas disponibles, los cuales inclusive pueden cumplir con los objetivos de la plataforma de manera más efectiva. La Tabla 5-37 muestra la única unidad de análisis asignada a esta categoría.

Tabla 5-37: Ejemplos de la unidad de análisis asignada a la categoría herramienta sustituible

Unidad de análisis
“Aunque fue útil, es fácilmente suplantable por herramientas dadas en el curso como collab o la pagina oficial de python visualice. PDTA: esta mejor el visualice de uncode porque te deja meter varios inputs de una sola ves”

5.5.2. Pregunta calificación automática

La Tabla 5-38 especifica la cantidad de unidades de análisis agrupadas por código, categoría y temática general, las cuales están clasificadas por desempeño académico del estudiante. En el caso de

la pregunta relacionada con la utilidad de la plataforma para obtener calificación automática de las soluciones diseñadas por los estudiantes, se generaron en total 7 códigos, 8 categorías y 4 temas. A continuación, se hace la definición y ejemplificación de las categorías identificadas juntos con los códigos asignados correspondientes.

Tabla 5-38: Cantidad de unidades de análisis agrupadas por códigos, categorías y temas

Tema	Categoría	Código	Unidades de análisis			
			Alumnos aprobados	Alumnos reprobados	Total	
Dificultades de implementación			3	0	3	
	Participación docente		3	0	3	
	Poco uso de la plataforma		2	0	2	
Facilidad de uso			5	0	5	
Logros pedagógicos	Aprendizaje autónomo	Evaluación de capacidades	7	0	7	
		Retroalimentación formativa	63	1	64	
		Seguimiento de calificaciones	7	0	7	
	Evaluación optimizada			19	0	19
		Calificación inmediata		73	2	75
		Calificación objetiva		71	6	77
		Retroalimentación inmediata		22	1	23
Mejorable en calificación automática	Fallas en validaciones		20	0	20	
		Inflexibilidad en las comprobaciones	20	3	23	
	Fallas generales		12	0	12	
	Incomprensible		3	0	3	
	Retroalimentación insuficiente		2	1	3	

Participación docente: Categoría asignada a las unidades donde los estudiantes recomiendan que durante el uso de la plataforma se incluya al docente, específicamente en el proceso de evaluación del desempeño académico. Por ejemplo, en actividades que requieren un criterio subjetivo, como especificar los porcentajes de las actividades en la nota final del curso. Algunas de las unidades de análisis que corresponden a esta categoría son especificadas en la Tabla 5-39.

Poco uso de la plataforma: Categoría asignada a las unidades que reportan insuficiencia en la experiencia con la plataforma, puesto que es posible que en algunos cursos hayan sido pocas las actividades desarrolladas con UNCode. En este sentido, algunos estudiantes perciben que sus opiniones pueden no tener el suficiente respaldo de conocimiento de la herramienta. La Tabla 5-40 muestra un ejemplo.

Tabla 5-39: Ejemplos de unidades de análisis asignadas a la categoría participación docente

Unidad de análisis

“Ya que casi nunca tenía errores en la calificación, y **en caso de presentarse algún inconveniente el docente siempre estaba atento** a nuestros correos y corregía el problema rápidamente ...”

“El programa funciona muy bien, la verdad estoy en primer semestre y siento que esta plataforma **ayudó considerablemente a los profesores para que estos ya no tuvieran que visualizar código por código, estudiante por estudiante**. El método no pareció en nada inadecuado, impreciso; tal vez hubo algunas **actividades específicas con las que el profesor nos recomendaba añadir algo** para que la plataforma lo tomara como correcto pero nada grave. ”

“El **profesor jamás califico, solamente se basaba en el porcentaje**, sin embargo considero muy importante, que tengan en cuenta un modo de calificación en porcentajes distintos. Uno para tareas, otro para talleres, parciales, y uno para quiz, de manera que sea más sencillo saber como lleva el semestre tanto para alumno y maestro. ”

Tabla 5-40: Ejemplo de la unidad de análisis asignada a la categoría poco uso de la plataforma

Unidad de análisis

“**Utilizamos UNCode únicamente un par de ocasiones** por lo que realmente no fue automatizada la calificación de los laboratorios que desarrollamos. A la fecha, son pocas las notas que conocemos de esta asignatura. ”

Aprendizaje autónomo: Categoría asignada a las unidades que resaltan la capacidad de la plataforma para fomentar el aprendizaje de los estudiantes con una intervención mínima del docente o monitor. La definición de esta categoría es homóloga a la establecida en la pregunta relacionada con el proceso de aprendizaje. Sin embargo, se evidencian diferencias con respecto a los códigos identificados, lo que permite identificar un mayor número de componentes que contribuyen al aprendizaje autónomo.

En primer lugar, se identifica un grupo de referencias donde se afirma que el uso de la plataforma permite a los estudiantes auto evaluar el nivel de conocimiento y habilidades que tienen en la asignatura, los cuales son agrupados con el código *Evaluación de capacidades*. En segunda instancia, está la *Retroatimentación formativa* que se refiere al suministro de información de calidad para identificar errores, mejorar y evaluar la calidad de los programas desarrollados por los alumnos. Por último, está el *seguimiento de calificaciones* donde los estudiantes resaltan herramientas de UNCode que permiten llevar un control estadístico de las notas obtenidas durante el semestre en las actividades realizadas. La Tabla 5-41 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría. Adicionalmente, desde la Tabla 5-42 hasta la Tabla 5-44 se muestran algunos ejemplos de unidades de análisis asignadas a cada código.

Evaluación optimizada: Categoría donde se agrupan las referencias relacionadas con las características y aspectos de la plataforma que permiten obtener una evaluación eficaz y apropiada de las soluciones propuestas por los estudiantes. La definición de esta categoría es homóloga a la establecida en la pregunta relacionada con el proceso de aprendizaje. Sumado a esto, los códigos generados dentro de la categoría también coinciden con los identificados en la pregunta anterior; los cuales co-

Tabla 5-41: Ejemplos de unidades de análisis asignadas a la categoría aprendizaje autónomo

Unidad de análisis
“Porque solo con subir el código bastaba para recibir la calificación del ejercicio a partir de los casos de prueba, lo cual es muy útil y permite aprender mejor que si uno mandara el código al profesor para que lo pruebe sin que uno pueda saber los resultados o el formato en el que espera la respuesta.”
“Me permite estar al tanto de mis calificaciones sin tener que recurrir a los profesores ”
“Ahorraba bastante tiempo ya que no debía esperar a que el docente revisara el código de cada estudiante, y podía intentarlo varias veces con el fin de mejorar el código poco a poco”

Tabla 5-42: Ejemplos de unidades de análisis asignadas al código evaluación de capacidades

Unidad de análisis
“Da valoraciones y correcciones que permiten generar una calificación de acuerdo al desempeño. ”
“Brinda respuestas claras y casos de prueba muy bien estructurados que ayudan a revisar la eficiencia del programa ”
“ Mide la capacidad de resolver problemas del que hace los ejercicios”

Tabla 5-43: Ejemplos de unidades de análisis asignadas al código retroalimentación formativa

Unidad de análisis
“Porque la calificación de casos era automática y permitía tener una especie de retroalimentación ”
“Porque según los casos de entrada daba una retroalimentación precisa de los casos correctos y los que no ”
“Porque a partir de la calificación automática podía saber si tenía que relacionar asuntos de memoria o los casos específicos a que mi código no cubra todos los casos de entrada.”

Tabla 5-44: Ejemplos de unidades de análisis asignadas al código seguimiento de calificaciones

Unidad de análisis
“Porque me permitía conocer el avance que tenía con respecto a las notas de talleres y quizzes en tiempo real”
“... Además de esto permite saber las notas que uno va acumulando de manera inmediata.”
“ Al instante conocía mi calificación , por lo que me mantuve al tanto de mi estado académico ”

responden a *Calificación inmediata*, *Calificación objetiva* y *Retroalimentación inmediata*. El hecho de identificar códigos similares en ambas preguntas, indica la consolidación de los conceptos de inmediatez y objetividad como valores que respaldan la calidad de la plataforma. La Tabla 5-45 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría. A partir de la Tabla 5-46 hasta la Tabla 5-48 se muestran algunos ejemplos de unidades de análisis asignadas a cada código.

Fallas en validaciones: Categoría asignada a los comentarios que reportan errores en el proceso de validación de los programas construidos, porque a pesar de que cumplen con lo exigido en los ejercicios al calificarlos son catalogados como equivocados o erróneos (ver ejemplos en la Tabla 5-49). Por otra parte, muchos de los alumnos especifican que los criterios de calificación y evaluación que la plataforma aplica son demasiado estrictos e inflexibles, los cuales no toleran pequeños errores de

Tabla 5-45: Ejemplos de unidades de análisis asignadas a la categoría evaluación optimizada

Unidad de análisis
“Era muy eficiente a la hora de dar su calificación una vez fue evaluado el programa. ”
“Porque compara de una manera eficiente los resultados que debían salir y los resultados que nos salieron a nosotros. Aclarando qué cosas son las que nos quedaban mal o simplemente su todo estaba bien. ”
“Es una manera más eficiente de poder calificar a la hora de la entrega”

Tabla 5-46: Ejemplos de unidades de análisis asignadas al código calificación inmediata

Unidad de análisis
“Por que la calificación era muy rápida ”
“ Facilitó la rapidez con la que obteníamos las notas ”
“corregía y entregaba la nota de cada ejercicio de manera inmediata ”
“Como da un feedback instantaneo es fácil ver la calificación automática y deja que el desarrollo de actividades sea mas sencillo.”

Tabla 5-47: Ejemplos de unidades de análisis asignadas al código calificación objetiva

Unidad de análisis
“porque dependiendo del porcentaje de eficiencia se otorgaba una nota automáticamente sin necesidad de esperar que el maestro revisara uno por uno”
“Porque la calificación iba de acuerdo al porcentaje de aciertos del código , lo cual me parece correcto”
“ verifica el resultado del código, conforme lo exigido ”

Tabla 5-48: Ejemplos de unidades de análisis asignadas al código retroalimentación inmediata

Unidad de análisis
“Como da un feedback instantáneo es fácil ver la calificación automática y deja que el desarrollo de actividades sea mas sencillo.”
“Su retroalimentación inmediata daba pistas para resolver mejor el problema.”
“ Inmediatamente de enviar la solución me mostraba el porcentaje obtenido. ”

escritura e ignoran la calidad del código en su semántica y generan malas calificaciones; las referencias donde se hace está especificación se agrupan con el código *Inflexibilidad en las comprobaciones*. La Tabla 5-50 detalla algunas de las unidades de análisis que representan esta categoría.

Fallas generales: Categoría asignada al grupo de respuestas donde se resaltan las fallas y problemas que presenta la plataforma durante su uso. Cabe resaltar que, la definición de está categoría es similar a la categoría de fallas generales UNCode de la pregunta relacionada con el proceso de aprendizaje. Sin embargo, los errores reportados en esta pregunta tienden a ser menos específicos. Algunas de las unidades de análisis que corresponden a esta categoría son especificadas en la Tabla 5-51.

Tabla 5-49: Ejemplos de unidades de análisis asignadas a la categoría fallas en validaciones

 Unidad de análisis

“A veces los casos **no se aceptaban debido a detalles que no tenían que ver con la realización óptima del código**, es decir, el código a pesar de que cumplía con las tareas exigidas, este no se aceptaba por detalles que según los enunciados no eran requeridos.”

“Cuando el output era un string o una tabla diseñada por strings me complicaba el hecho de que **aunque la respuesta estuviera bien, la maquina lo interpretaba diferente** por ej: un '3' de un 3 o un espacio de un \hat{h} acian que la calificación fuera 0 %”

“Por que en algunos casos **el programa sí podía resolver el problema y algunos errores de impresión podían afectar negativamente**, afectando totalmente la nota sin tener en cuenta que se llegó a una solución pertinente para el problema.”

Tabla 5-50: Ejemplos de unidades de análisis asignadas a inflexibilidad en las comprobaciones

 Unidad de análisis

“Me parece que es **muy exigente calificando**, ya que si no es 100 % el código con los parámetros establecidos no se tiene en cuenta el código así tenga parte bien, para estar empezando a programar es algo duro calificando.”

“Debido a sus casos concretos, **exigía bastante una precisión en los códigos**”

“Si bien sí fue útil para obtener de manera rápida la calificación de los códigos, se tornaba **demasiado estricto a la hora de recibir los códigos**, obligando al usuario a encontrar la forma más óptima de escribir el código”

Tabla 5-51: Ejemplos de unidades de análisis asignadas a la categoría errores UNCode

 Unidad de análisis

“... **Tiene algunas pequeñas fallas** de vez en cuando pero no es mayor problema”

“Aunque **tuvo algunos fallos** en donde no estuvo disponible el uso de UNCode fue una forma muy óptima para obtener calificaciones”

“igual, algunos conceptos básicos los maneja con facilidad pero **cuando se prueban casos extremadamente grandes puede que tome demasiado tiempo** del que se querría ...”

Incomprensible: Categoría asignada a las unidades donde se señalan dificultades para comprender claramente el objetivo de la calificación automática, en específico su funcionalidad o la información que genera. La Tabla 5-52 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-52: Ejemplos de unidades de análisis asignadas a la categoría dificultades para entenderlo

 Unidad de análisis

“Debido a que en muchos programas, el enunciado y el código no van de la mano; la información respecto a **las condiciones de ejecución del código son ambiguas** con lagunas ...”

“Pues a veces era **difícil de entender**”

Retroalimentación insuficiente: Categoría asignada a las unidades donde se afirma que la información que se ofrece al estudiante cuando envía una solución carece de contenido y explicación, lo

que no permite el pleno entendimiento del programa diseñado ni favorece el proceso de construcción del mismo. La Tabla 5-53 muestra algunos ejemplos.

Tabla 5-53: Ejemplos de unidades de análisis asignadas a la categoría retroalimentación insuficiente

Unidad de análisis
“... Además, la evaluación automática de UNCode no da una retroalimentación sobre el estado del código , es decir, si este es está siendo escrito de la mejor manera o si, por el contrario, esta siendo escrito con malas prácticas de programación que el aprendiz difícilmente podrá identificar por su cuenta ...”
“es útil tener los porcentajes pero de nada sirve que uno tenga solo un visto bueno y no pueda revisar estos casos de prueba, además que siento que es necesario que una vez que uno complete con éxito los casos se libere el código fuente para una mayor comprensión caso adicional cuando el plazo se termina y algún estudiante no logro completar estos ejercicios, esto como un método adicional para el entendimiento del código y el proceso del mismo”

5.5.3. Pregunta retroalimentación inmediata

La Tabla 5-54 especifica la cantidad de unidades de análisis agrupadas por código, categoría y temática general, las cuales están clasificadas por desempeño académico del estudiante. En el caso de la pregunta relacionada con la utilidad de la plataforma para obtener retroalimentación automática de las soluciones diseñadas por los estudiantes, se generaron en total 3 códigos, 5 categorías y 4 temas. A continuación se presenta la definición y ejemplificación de las categorías identificadas juntos con los códigos asignados correspondientes.

Tabla 5-54: Cantidad de unidades de análisis agrupadas por códigos, categorías y temas

Tema	Categoría	Código	Unidades de análisis		
			Alumnos aprobados	Alumnos reprobados	Total
Condiciones iniciales claras			6	2	8
			2	0	2
Mejorable en retroalimentación	Casos de prueba ocultos		12	0	12
			35	0	35
	Orientación insuficiente	Detalles menores	7	2	9
		Falta de claridad	47	3	50
		Identificación autónoma de errores	12	0	12
			157	0	157
Resolución de problemas	Comparación con salidas esperadas		33	2	35
	Corregir errores		51	3	54
	Especificaciones en la retroalimentación		60	2	62
Señalar errores			89	3	92

Casos de prueba ocultos: Categoría asignada a las unidades que destacan ocasiones en las que la diferencia entre el resultado obtenido y el esperado no se visualiza al realizar un intento de solución. En este sentido, a los estudiantes se les dificulta obtener información relevante para la identificación de errores. Algunas de las unidades de análisis que corresponden a esta categoría son especificadas en la Tabla 5-55.

Tabla 5-55: Ejemplos de unidades de análisis asignadas a la categoría casos de prueba ocultos

Unidad de análisis
“Si había un error en un caso de prueba, UNCode permitía ver la salida esperada y la entrada del código en ciertos casos, a veces no permitía ver los casos de prueba y eso dificultaba un poco la depuración.”
“Para algunos problemas UNCode no entregaba el expected output vs el output generado por nosotros los estudiantes, en esos casos no era fácil saber en donde fallaba el código”
“... sin embargo, cuando el número de respuestas es muy alto, UNcode (a veces) no muestra la comparación , por lo que es un poco más difícil de conocer el error.”

Orientación insuficiente: Categoría asignada a las unidades donde se resalta que la retroalimentación obtenida no es suficiente, puesto que en algunos casos la información adquirida no permitía identificar específicamente el error o la manera de corregir dentro del programa. La Tabla 5-56 detalla algunas de las unidades de análisis que representan esta categoría.

Dentro de esta categoría se derivan causas y consecuencias de la insuficiencia en la retroalimentación. En primer lugar, está el código *detalles menores* que se refiere a la sensibilidad de la plataforma con respecto a aspectos errores menores en las salidas de los programas, como caracteres adicionales o sobrantes, lo que evita obtener cualquier tipo de retroalimentación para solucionarlos (ver Tabla 5-57). Por otra parte, un grupo de referencias codificadas como *falta de claridad*, afirman que las explicaciones de la plataforma carecen de claridad o detalle, y que también se puede evidenciar a nivel de las instrucciones iniciales o en los mensajes de la plataforma (ver Tabla 5-58). Por último, se identifican referencias codificadas como *identificación autónoma de errores*, las cuales afirman que debido a que la retroalimentación no permite la identificación clara de errores, se genera una obligación autónoma en el proceso de corrección de los programas desarrollados (ver Tabla 5-59).

Tabla 5-56: Ejemplos de unidades de análisis asignadas a la categoría orientación insuficiente

Unidad de análisis
“La parte del debugging se hace muy compleja en UNcode ya que los errores aunque son específicos. Muchas veces no da información importante para corregir”
“ Algunas veces la retroalimentación de UNCode no es muy específica en cuanto a tiempos de ejecución.”
“A veces era muy vaga la explicación del error , simplemente no era específico.”

Tabla 5-57: Ejemplos de unidades de análisis asignadas al código detalles menores

Unidad de análisis
“Es verdad que el código tiene que ser completamente exacto y eso lleva a fallos no tan graves como una mayúscula o una coma... ”
“Muchas veces no se retroalimentan los errores en los códigos, y en algunos casos sólo por espacios da todo el punto incorrecto ”
“... sin embargo muchas veces el error no era claro, como por ejemplo, en los casos de imprimir tablas, muchas veces mi programa imprimía espacios adicionales y era difícil encontrar el error. ”

Tabla 5-58: Ejemplos de unidades de análisis asignadas al código falta de claridad

Unidad de análisis
“... había errores que no era capaz de identificar, en parte porque no entendía el mensaje que la plataforma sacaba. Esto me confundía y entonces podía llegar a cometer errores donde no habían por este malentendido ...”
“Era difícil de entender dónde estaba el error y no daba descripción detallada ”
“Muchas veces los codigos de error son poco claros o directamente aparece un runtime en vez de marcar la línea del error ...”

Tabla 5-59: Ejemplos de unidades de análisis asignadas al código identificación autónoma de errores

Unidad de análisis
“En muy pocas ocasiones UNcode presentaba los posibles errores, mientras en la gran mayoría los errores por falta de un espacio o algo por el estilo debía descubrirlo por mi cuenta. ”
“En sí UNCode no daba las retroalimentaciones. Uno mismo debía pensar en los posibles errores. ”
“Aunque UNcode muestra los errores que presento el programa, son demasiado generales. Están clasificados en categorías, por lo que varias veces tuve que recurrir a otras herramientas para comprobar cual fue era el error (nombre del error).”

Comparación con salidas esperadas: Categoría asignada a las unidades que hacen referencia a la utilidad de hacer un paralelo entre las salidas que arroja el programa desarrollado por el estudiante y las salidas esperadas, lo que facilita la identificación de errores y estrategias de corrección. La Tabla 5-60 muestra algunos ejemplos de unidades de análisis asignadas en esta categoría.

Tabla 5-60: Ejemplos de unidades de análisis asignadas a la categoría comparación con salidas esperadas

Unidad de análisis
“Porque permitía ver qué salidas daba el programa frente a ciertas entradas en las cuales había un error, lo cual, con un poco de intuición y conocimiento del código se podía arreglar para que el mismo quedara sin errores algunos.”
“Ayuda a encontrar en que esta fallando el problema del programa, es buena herramienta la comparación de la salida esperada y la salida que esta dando el programa que uno ingresó.”
“Porque a partir de las entradas que UNcode da al programa y las respuestas que espera que salgan , permite intuir fácilmente los problemas que tiene el código”

Corregir errores: Categoría donde se agrupan las respuestas donde los estudiantes afirman que, la identificación clara de los errores en el desarrollo del código es lo que permite que entender el tipo de falla cometido y las estrategias de corrección más apropiadas. Algunas de las unidades de análisis que corresponden a esta categoría son especificadas en la Tabla 5-61.

Tabla 5-61: Ejemplos de unidades de análisis asignadas a la categoría corregir errores

Unidad de análisis
“Me pareció muy bueno el que la plataforma compare las salidas, ya que nos permite corregir errores de una manera ágil. ”
“es una forma facil de identificar los errores y encontrar una mas pronta solución ”
“Si, porque al dar el resultado automáticamente garantiza que no se exceda el tiempo límite para la entrega de los ejercicios y poder resolver los errores naturalmente ”

Especificaciones en la retroalimentación: Categoría asignada a las unidades donde se le atribuye a la retroalimentación generada en la plataforma, un alto grado de detalle y explicación. Adicionalmente, la especificidad sobre el tipo de errores presentes en el código, pueden contribuir a la resolución oportuna de problemas de programación. La Tabla 5-62 detalla algunas de las unidades de análisis que representan esta categoría.

Tabla 5-62: Ejemplos de unidades de análisis asignadas a la categoría especificaciones en la retroalimentación

Unidad de análisis
“Si es bastante ya que la misma plataforma tiene un apartado donde explica que significa cada mensaje que arroja al hacer funcionar el programa ...”
“Decía específicamente cual era el error y donde estaba ”
“ Explicaba exactamente cual era el error en el código además de permitir ver el momento exacto del error”

5.6. Codificación axial

Posterior a la identificación y definición de las categoría y códigos dentro del conjunto de datos, se realiza la codificación axial donde se agrupan las clasificaciones en tema generales. En este orden de ideas, para cada una de las tres preguntas consideradas de la encuesta de percepción, son listados y descritos los temas identificados. Como soporte visual se emplean mapas ramificados de cada tema, con los cuales se puede obtener una perspectiva general del tamaño ponderado de las categorías y códigos, respecto al conjunto total de referencias codificadas.

5.6.1. Pregunta proceso de aprendizaje

La Tabla 5-3 también especifica la cantidad de unidades de análisis agrupadas por tema con respecto a la pregunta relacionada con la utilidad de la plataforma en el proceso de aprendizaje. En este caso se generan 6 temas generales, los cuales son definidos, ejemplificados y representados por medio de mapas ramificados.

Ambiente de la plataforma: Tema donde se agrupan las unidades que hacen referencia a la estética, apariencia, organización y/o funcionalidad de la plataforma, en términos de su interfaz gráfica. Las referencias sobre este tema van desde el reconocimiento como aspectos facilitadores del aprendizaje, hasta el extremo opuesto de características que dificultan el proceso del estudiante. Dentro de este tema no se identifican categorías o códigos específicos. La Tabla 5-63 muestra algunos ejemplos de unidades de análisis asignadas en este tema.

Tabla 5-63: Ejemplos de unidades de análisis asignadas al tema ambiente de la plataforma

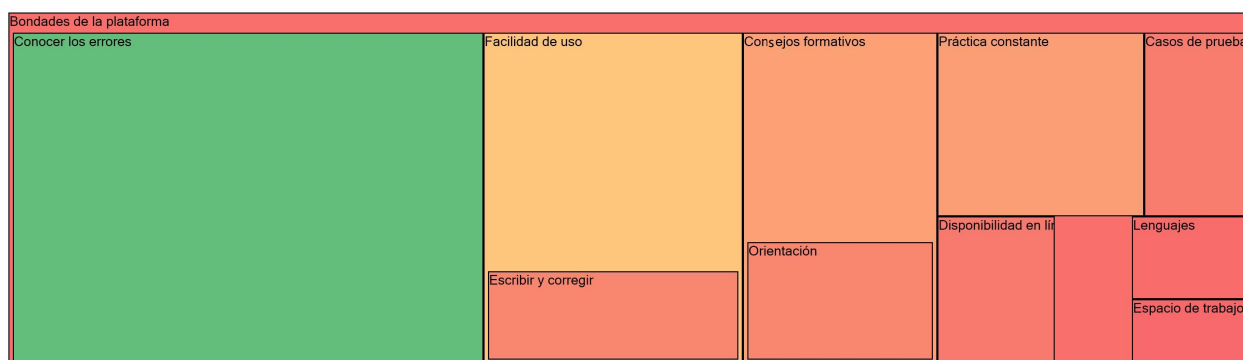
Unidad de análisis
“La interfaz no es muy amable con el usuario inicial, la escritura es mucho más estricta.”
“... Además tiene una plataforma estéticamente adaptada para el usuario.”
“... Recomiendo que la ventana de este sea un poco más grande.”

Bondades de la plataforma: Tema donde se reúnen las unidades que reconocen las fortalezas propias de la plataforma UNCode, la cuales acorde con los estudiantes contribuyen directamente al aprendizaje de valiosas habilidades para la programación de computadores. La Tabla 5-64 muestra algunos ejemplos de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-1, las categorías con mayor cantidad de registros son *Conocer los errores* y *Facilidad de uso*, lo que indica que la mayoría de estudiantes utilizan la plataforma para encontrar errores dentro de sus soluciones y la facilidad de manejo de la herramienta los motiva a emplearla en su proceso de aprendizaje. Posteriormente, están *Consejos formativos* y *Práctica constante*, indicando que algunos alumnos perciben UNCode como una plataforma que ofrece retroalimentación enriquecedora y que permite la aplicación práctica de los conocimientos de la asignatura. Finalmente, la menor cantidad de referencias se encuentra en las categorías *Disponibilidad en línea* y *Lenguajes de programación*, es decir, que un grupo pequeño de estudiantes valoran la plataforma por su disponibilidad de ejecución en línea, sin necesidad de descarga local, y su capacidad de soportar múltiples lenguajes de programación.

Google Colaboratory: Tema donde se agrupan las unidades que resaltan la utilidad del uso de notebooks de Google Colaboratory en el proceso de escritura y ejecución de código de Python desde un navegador disponible. En general, los estudiantes hacen comparaciones entre las funcionalidades de UNCode frente a Colab, identificando ventajas y desventajas entre las dos herramientas. Dentro de

Tabla 5-64: Ejemplos de unidades de análisis asignadas al tema bondades de la plataforma

Unidad de análisis
<p>“UNCode fue una herramienta bastante útil a la hora de presentar los talleres y quices, pero en sí, esta no fue útil para aprender acerca de la programación, aunque realmente, es bastante útil conocer los errores en el programa con los casos de prueba, pero no siempre fueron dados ...”</p> <p>“Porque me permitió realizar los ejercicios en un ambiente muy cómodo, además es fácil de manejar, lo que permite mejorar nuestras habilidades.”</p>

**Figura 5-1:** Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema bondades de la plataforma

este tema no se identifican categorías o códigos específicos. La Tabla 5-65 muestra algunos ejemplos de unidades de análisis asignadas en este tema.

Tabla 5-65: Ejemplos de unidades de análisis asignadas al tema Google Collaboratory

Unidad de análisis
<p>“La plataforma no es versátil con los resultados, es preferible usar Colab”</p> <p>“Si ya que UNcode permite ingresar la entrada del código de una manera mas rápida que plataformas como Colab lo cual facilitaba probar los ejercicios, aunque en ocasiones emitía mensajes de error al mismo tiempo que la respuesta ...”</p>

Herramientas de UNCode: Tema donde se encuentran las menciones directas, indirectas o genéricas a las herramientas presentes en la plataforma, las cuales fueron diseñadas para la generación de retroalimentación sumativa y formativa, facilitar del aprendizaje y apoyo durante el proceso de construcción de soluciones a problemas de programación de computadores. La Tabla 5-66 muestra algunos ejemplos de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-2, la categoría con mayor cantidad de registros es *Python Tutor*, lo que indica que la mayoría de estudiantes evidencian los beneficios de visualizar paso a paso la ejecución de los programas construidos. En orden de magnitud siguen *Custom input* y *Lint*, lo que significa que un

grupo más pequeño de alumnos, utiliza las pruebas personalizadas y el resaltado de buenas prácticas de programación para refinar las soluciones diseñadas.

Tabla 5-66: Ejemplos de unidades de análisis asignadas al tema herramientas de UNCode

Unidad de análisis
“Las herramientas de test y visualice son muy útiles a la hora de encontrar errores”
“Tenía las herramientas adecuadas , permitía probar diferentes aplicaciones y sobre todo tenía herramientas adicionales que permitían previsualizar el código paso por paso y revisar así cuáles eran los errores que se tenían”
“Porque cuenta con herramientas bastante útiles como un tutor de python y permite a los profesores limitar el tiempo de ejecución de un programa para que el estudiante deba optimizarlo lo que más pueda. Estas última herramienta hace que el estudiante tenga que pensar mejor el código, lo que conlleva un mayor aprendizaje. ”

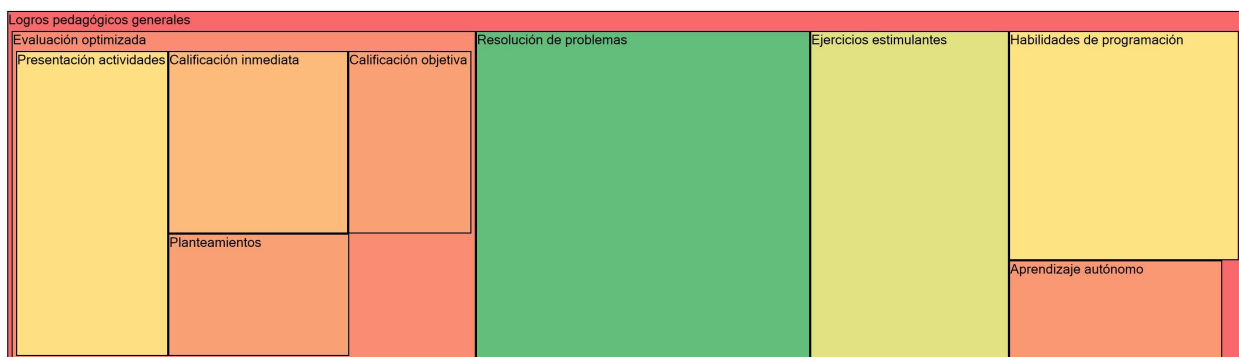


Figura 5-2: Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema herramientas de UNCode

Logros pedagógicos: Tema donde se agrupan las unidades que hacen referencia a los efectos en el proceso de aprendizaje que los estudiantes reportan obtuvieron debido al uso de UNCode. Dentro de este tema se percibe que los beneficios de la plataforma no se ubican exclusivamente en los alumnos, puesto que puede llegar a impactar favorablemente la labor docente. La Tabla 5-67 muestra algunos ejemplos de unidades de análisis asignadas aquí. En el mapa ramificado de la Figura 5-3 las categorías con mayor cantidad de registros son *Evaluación optimizada* y *Resolución de problemas*, lo que indica que la mayoría de estudiantes logra evaluar las soluciones construidas y obtener habilidades de resolución de problemas de programación de computadores mediante el uso de la plataforma. Posteriormente, están *Ejercicios estimulantes* y *Habilidades de programación*, es decir, que algunos alumnos perciben UNCode como una herramienta que permite plantear actividades académicas motivantes y el desarrollo de conocimientos prácticos de la asignatura. Finalmente, la menor cantidad de referencias se encuentra en la categoría *Aprendizaje autónomo*, indicando que para un porcentaje pequeño de estudiantes, la plataforma promueve procesos de aprendizaje sin necesidad de intervención de un docente o monitor.

Tabla 5-67: Ejemplos de unidades de análisis asignadas al tema logros pedagógicos

Unidad de análisis
“Es una herramienta que ofrece herramientas muy útiles a la hora de programar, permite ver los errores que cometió al estudiante además de ofrecer alternativas para solucionar los mismos ...”
“La plataforma fue un muy buen entorno para resolver ejercicios y con la implementación que se hizo casi al final hizo que casi no se necesitara de la tutoría del profesor para entender los errores que se estaban cometiendo ...”

**Figura 5-3:** Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema logros pedagógicos

Mejorable para aprender a programar: Tema donde se agrupan las referencias sobre las carencias, fallas y aspectos por mejorar de la plataforma, según la experiencia de uso de los estudiantes. La Tabla 5-68 muestra algunos ejemplos de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-4, las categorías con mayor cantidad de registros son *Fallas generales UNCode* y *Inflexibilidad en las comprobaciones*, lo que indica que la mayoría de los estudiantes reportan fallas en la plataforma con respecto a su funcionalidad y la rigurosidad de los casos de prueba planteados. Posteriormente, están *Falla en el objetivo educativo* y *Insuficiencia de la retroalimentación*, es decir, un grupo menor de alumnos reporta no evidenciar el beneficio de la plataforma en el proceso de aprendizaje y afirma que la retroalimentación obtenida debe ser refinada o complementada. Finalmente, la menor cantidad de referencias se encuentra en la categoría *Herramienta sustituible*, indicando que un número pequeño de estudiantes optan por utilizar herramientas diferentes a UNCode para la verificación y refinamiento sus programas.

Tabla 5-68: Ejemplos de unidades de análisis asignadas al tema mejorable para aprender a programar

Unidad de análisis
“... pero muchas veces las herramientas como Python Tutor o Test Code no funcionaban correctamente , por lo que tocaba usar otro entorno para ejecutar y probar el código”
“La herramienta muchas veces da errores aún cuando la salida requerida es exactamente a como la pide ... ”

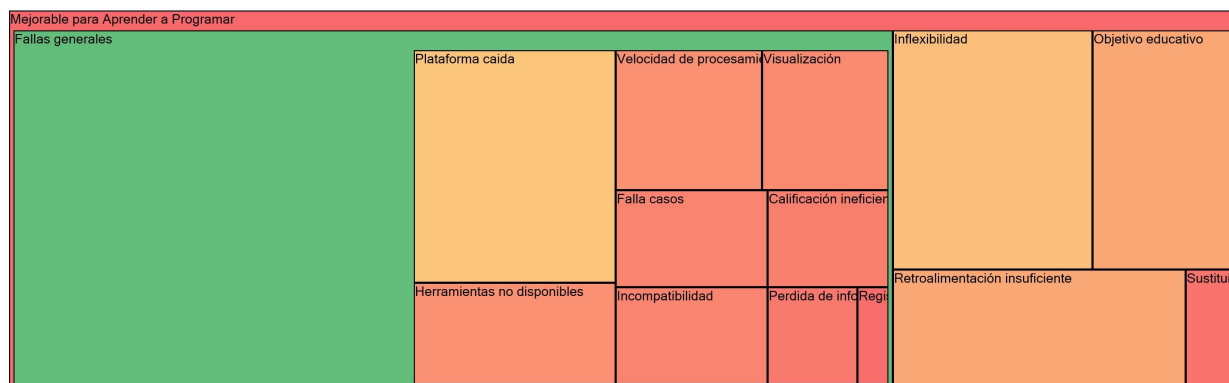


Figura 5-4: Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema mejorable para aprender a programar

5.6.2. Pregunta calificación automática

La Tabla 5-38 especifica la cantidad de unidades de análisis agrupadas por tema con respecto a la pregunta relacionada con la utilidad de la calificación automática. En este caso se generan 4 temas generales, los cuales son definidos, ejemplificados y representados por medio de mapas ramificados.

Dificultades de implementación: Tema donde se agrupan las unidades que destacan conductas y determinantes que no favorecen el proceso de implementación y uso de la herramienta. Cabe resaltar, que estas conductas no se derivan directamente del funcionamiento de la plataforma, sino que hace referencia al uso por parte de estudiantes y/o docentes. La Tabla 5-69 muestra algunos ejemplos de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-5, la categoría con mayor cantidad de registros es *Participación docente*, lo que indica que la mayoría de estudiantes afirman que el uso de la plataforma sería facilitado si el acompañamiento del profesor se incrementara. Posteriormente, la menor cantidad de referencias se encuentra en la categoría *Poco uso de la plataforma*, es decir, que un grupo pequeño de estudiantes consideran que su experiencia con UNCode no es suficiente para evidenciar sus beneficios.

Tabla 5-69: Ejemplos de unidades de análisis asignadas al tema dificultades de implementación

Unidad de análisis

“Utilizamos UNCode únicamente un par de ocasiones por lo que realmente no fue automatizada la calificación de los laboratorios que desarrollamos ...”

“El profesor jamás califico, solamente se basaba en el porcentaje, sin embargo **considero muy importante, que tengan en cuenta un modo de calificación en porcentajes distintos**. Uno para tareas, otro para talleres, parciales, y uno para quiz, de manera que sea más sencillo saber como lleva el semestre tanto para alumno y maestro.”

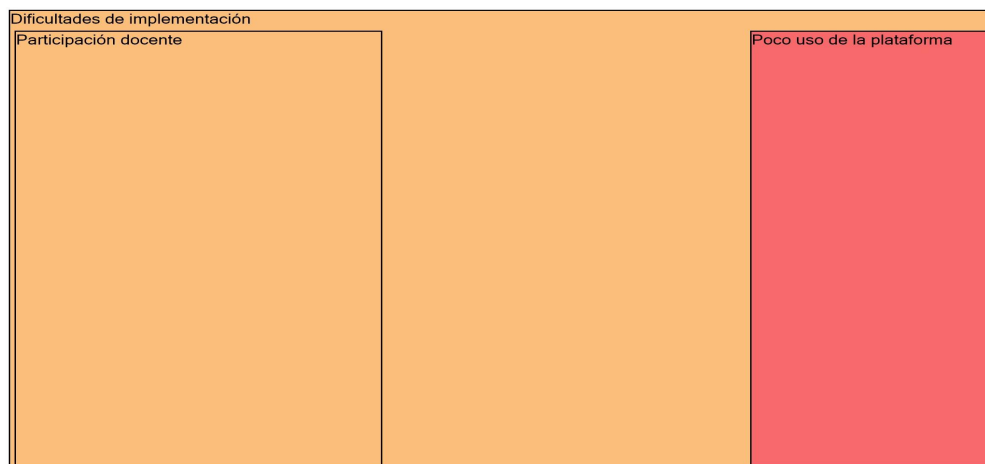


Figura 5-5: Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema dificultades de implementación

Facilidad de uso: Tema donde se agrupan las unidades donde se afirma que la plataforma es fácil y sencilla de entender. En específico, los estudiantes señalan que UNCode permite ingresar y poner a prueba los programas construidos de una forma rápida y sin dificultades. Dentro de este tema no se identifican categorías o códigos específicos. La Tabla 5-70 muestra algunos ejemplos de unidades de análisis asignadas en este tema.

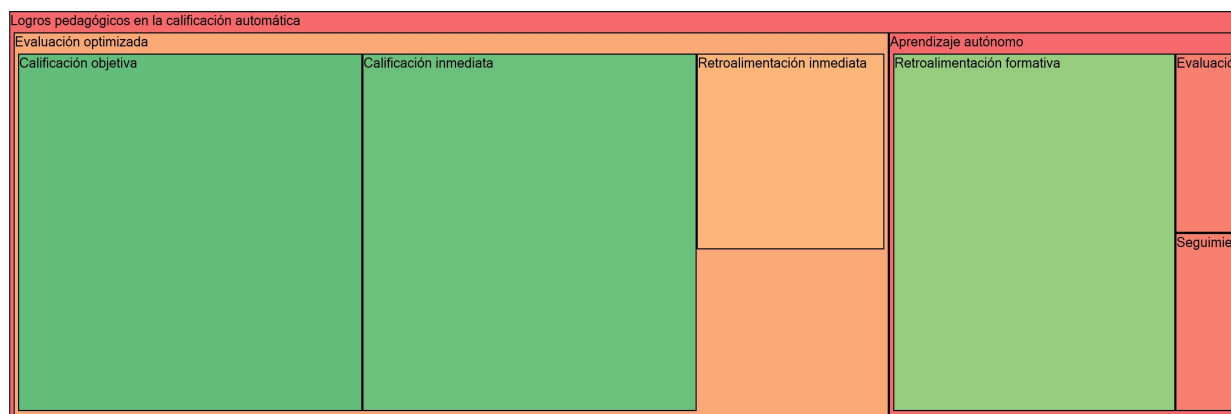
Tabla 5-70: Ejemplos de unidades de análisis asignadas al tema facilidad de uso

Unidad de análisis
“Bastante fácil y simple de entender, mas allá de 1 o 2 problemas en el semestre no paso nada importante.”
“Bastante simple, el programa daba las calificaciones automáticas con la respuesta correcta.”

Logros pedagógicos: Tema donde se agrupan las unidades que hacen referencia a los efectos pedagógicos que los estudiantes afirman que alcanzan debido al uso de la plataforma. La definición de este tema es homóloga al acuñado en la pregunta relacionada con la utilidad de la plataforma en el proceso de aprendizaje, lo que indica que en ambas preguntas se identifican percepciones similares que generan continuidad en los hallazgos. La Tabla 5-71 muestra algunos ejemplos de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-6, la categoría con mayor cantidad de registros es *Evaluación optimizada*. En contraste, la menor cantidad de referencias se encuentra en la categoría *Aprendizaje autónomo*. Estos resultados concuerdan con el mapa ramificado de la Figura 5-3, lo que indica que los estudiantes evidencian más el efecto que tiene la plataforma en la evaluación efectiva de las soluciones construidas, en comparación el apoyo a procesos de aprendizaje autónomos.

Tabla 5-71: Ejemplos de unidades de análisis asignadas al tema logros pedagógicos

Unidad de análisis
“Permite ser mas consiente de que avance se ha tenido en el curso , tanto por la calificación como por la reducción de intentos necesarios, quizá eso sea un buen indicador de mejora en el aprendizaje”
“Sí, pues la calificación se demoraba unos segundos y permitía ver no solo que se tenía mal, como se espera que este bien y que parte del código se encuentra el error , si no también las entradas que se usarían para evaluar el código. ”

**Figura 5-6:** Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema logros pedagógicos

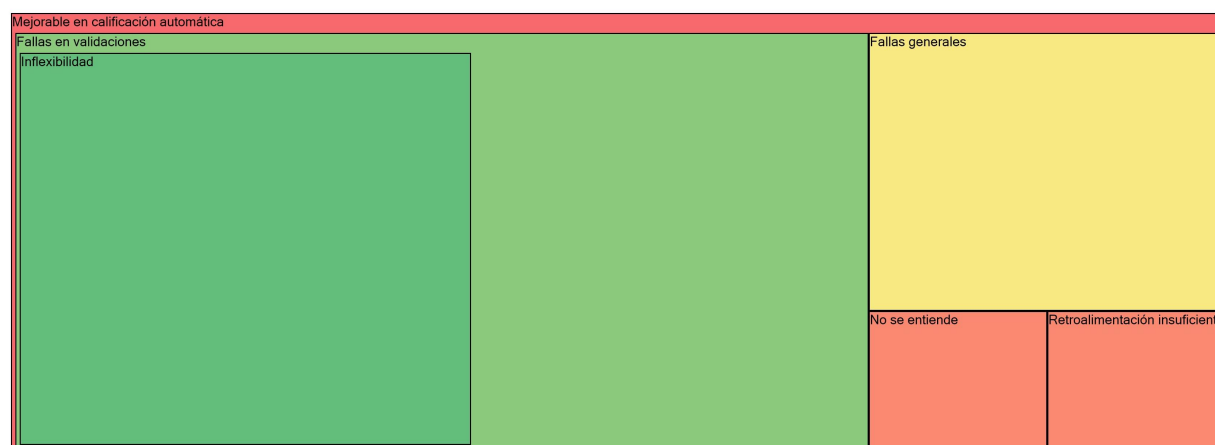
Mejorable en calificación automática: Tema donde se agrupan las carencias, fallas y aspectos por mejorar de la plataforma, según la experiencia de uso de los estudiantes durante el curso. La definición de este tema es homóloga al acuñado en la pregunta relacionada con la utilidad de la plataforma en el proceso de aprendizaje. Adicionalmente, la mayoría de las categorías de este tema concuerdan con las identificadas en la pregunta anterior, lo que indica continuidad en los hallazgos del análisis de contenido. La Tabla 5-72 muestra algunos ejemplos de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-7, la categoría con mayor cantidad de registros es *Fallas en validaciones*, lo que indica que la mayoría de estudiantes sugieren como mejora corregir errores o ajustar los casos de prueba de la plataforma. En orden de magnitud está la categoría *Errores UNCode*, es decir, que una cantidad menor de alumnos resaltan las fallas en la funcionalidad general de UNCode. Finalmente, la menor cantidad de referencias se encuentra en las categorías *Dificultades para entenderlo* y *Retroalimentación insuficiente*, indicando que pocos estudiantes recomiendan aumentar el nivel de detalle y complementar la retroalimentación ofrecida en la plataforma.

5.6.3. Pregunta retroalimentación inmediata

La Tabla 5-54 especifica la cantidad de unidades de análisis agrupadas por tema con respecto a la pregunta relacionada con la utilidad de la retroalimentación automática. En este caso se generan 4 temas

Tabla 5-72: Ejemplos de unidades de análisis asignadas al tema mejorable en calificación automática

Unidad de análisis
“No es completamente infalible y la salida puede tener validaciones erróneas bastante importantes”
“En algunos momentos la página presentaba inconvenientes que se salían de nuestras manos”
“Debido a que en muchos programas, el enunciado y el código no van de la mano; la información respecto a las condiciones de ejecución del código son ambiguas con con lagunas, lo que en muchos casos, aunque el código vaya a lugar, no lo hace la solución del problema y se descarta frívolamente.”

**Figura 5-7:** Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema mejorable en calificación automática

generales, los cuales son definidos, ejemplificados y representados por medio de mapas ramificados.

Condiciones iniciales claras: Tema donde se agrupan las unidades que señalan la necesidad de que las instrucciones o condiciones iniciales, descritas en cada ejercicio, tengan un mayor grado de claridad y detalle; ya que debido a esto, algunos estudiantes reportan dificultad en el entendimiento de la retroalimentación de la plataforma tras evaluar los programas desarrollados y enviados. Dentro de este tema no se identifican categorías o códigos específicos (ver Tabla 5-73).

Tabla 5-73: Ejemplos de unidades de análisis asignadas al tema condiciones iniciales claras

Unidad de análisis
“Realmente hay claridad dependiendo de las condiciones puestas por el profesor, a veces no fueron muy claras ”
“En casos si fue útil, pero en otros cuando no se entendía muy bien el ejercicio , tener las salidas no era muy útil.”

Mejorable: Tema donde se agrupan las unidades donde son señaladas las insuficiencias en la calidad o el tipo de retroalimentación ofrecida por la plataforma. La Tabla 5-74 muestra algunos ejemplos

de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-8, la categoría con mayor cantidad de registros es *Orientación insuficiente*. En contraste, la menor cantidad de referencias se encuentra en la categoría *Fallas de funcionamiento*. Estos resultados contrastan las conclusiones de la Figura 5-4 y 5-7, puesto que en este caso la mayoría de estudiantes se enfoca en resaltar como aspecto a mejorar las carencias en la retroalimentación obtenida, antes que los errores de funcionamiento de la plataforma.

Tabla 5-74: Ejemplos de unidades de análisis asignadas al tema mejorable

Unidad de análisis
<p>“Si bien permite ver el valor esperado de las respuestas, para casos específicos la información no se desglosaba para otros, dando una incertidumbre como resultado. Igualmente no había guianza ni acompañamiento para la resolución de los ejercicios más complejos.”</p> <p>“Si bien muestra un paso a paso del código, y presenta unas herramientas para verificar el sintaxis del código, estos a veces tardaban demasiado o no funcionaban”</p> <p>“En muy pocas ocasiones UNcode presentaba los posibles errores, mientras en la gran mayoría los errores por falta de un espacio o algo por el estilo debía descubrirlo por mi cuenta.”</p>



Figura 5-8: Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema mejorable

Resolución de problemas: Tema donde se agrupan las unidades que reconocen que el tipo y calidad de la retroalimentación generada de la plataforma, favorece significativamente el desarrollo de habilidades para la resolución efectiva de los problemas de programación propuestos en el curso. La Tabla 5-75 muestra algunos ejemplos de unidades de análisis asignadas en este tema. Acorde con el mapa ramificado representado por la Figura 5-9, la categoría con mayor cantidad de registros es *Especificaciones en la retroalimentación*, lo que indica que la mayoría de estudiantes considera que la información ofrecida por UNCode permite desarrollar habilidades para la solución de problemas de programación. En orden de magnitud, después está la categoría *Corregir errores*, indicando que para algunos alumnos la retroalimentación de la plataforma permite la identificación y corrección de

errores, lo que facilita la solución de las actividades del curso. Finalmente, la menor cantidad de referencias se encuentra en la categoría *Comparación con salidas esperadas*, es decir, que un grupo más pequeño de estudiantes logra dar solución a los ejercicios propuestos en la plataforma, por medio de la comparación del resultado esperado con la salida del programa construido.

Tabla 5-75: Ejemplos de unidades de análisis asignadas al tema resolución de problemas

Unidad de análisis
<p>“Como he dicho anteriormente, las herramientas que nos ofrece UNCode son una base esencial para nosotros los que aprendemos a programar, pues sin la ayuda de estas herramientas se nos dificultaría mucho mas resolver los problemas.”</p> <p>“En ocasiones aparecía un error e indicaba en dónde se encuentra ese error, o algunas salidas faltaban, y con la herramienta de visualizar código se entendía el funcionamiento general del código y se le podía realizar un seguimiento.”</p>

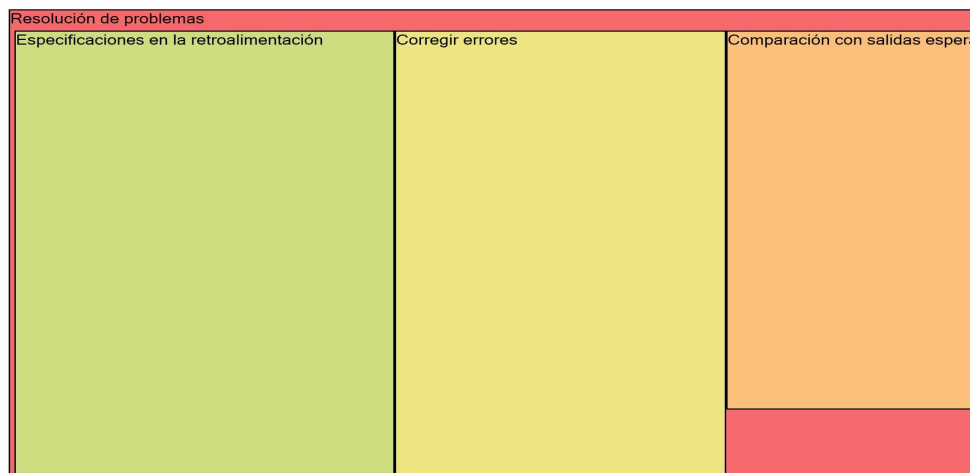


Figura 5-9: Mapa ramificado de acuerdo con el número de referencias codificadas de las categorías y códigos del tema resolución de problemas

Señalar errores: Tema donde se agrupan las unidades donde los estudiantes resaltan la utilidad de la retroalimentación automática en la identificación de los errores presentes en el desarrollo del programa. Las referencias en este tema evidencian visiones opuestas, puesto que algunos estudiantes afirman que el señalamiento de los errores es suficiente para corregir y refinar las soluciones construidas; en contraste, otro grupo de alumnos califican la información como insuficiente. Dentro de este tema no se identifican categorías o códigos específicos. La Tabla 5-76 muestra algunos ejemplos de unidades de análisis asignadas en este tema.

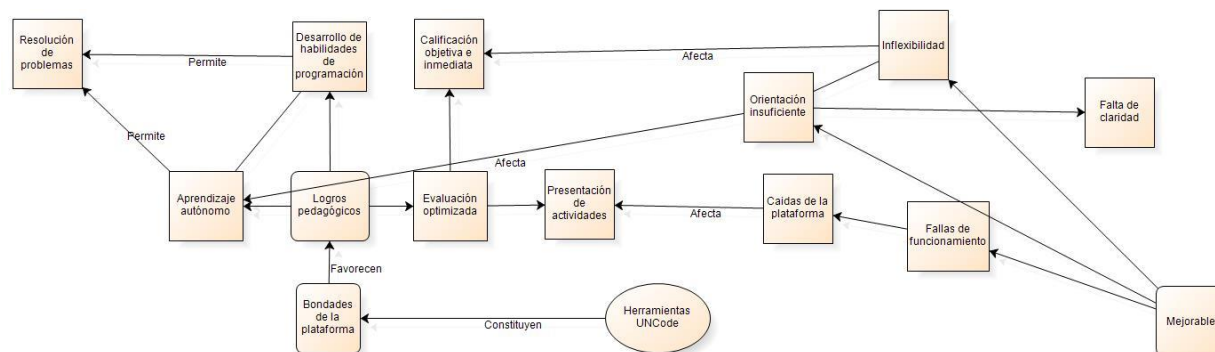
Tabla 5-76: Ejemplos de unidades de análisis asignadas al tema señalar errores

Unidad de análisis
“Permitía identificar la sección específica del programa para corregir. ”
“Sí, ya que mostraba la línea en la que fallaba en código ”
“Poder ver errores en los programas enviados facilita la identificación de problemas y la solución de las mismas”

5.7. Relativización de datos

A partir de los resultados de la codificación abierta y axial, se construye un modelo general, ilustrado en la Figura 5-10, que tiene como objetivo representar gráficamente los vínculos y relaciones entre temas, categorías y códigos. En primer lugar, se evidencia que el conjunto de herramientas embebidas en UNCode, especialmente python tutor, custom input y linter, son la base práctica de las bondades de la plataforma. En otras palabras, las fortalezas y posibilidades de UNCode que contribuyen al aprendizaje de programación de computadores se ven representadas por medio de las opciones propias de la plataforma. Adicionalmente, las bondades permiten a los estudiantes obtener logros pedagógicos que los estudiantes consideran se logran debido al uso de UNCode. Estos logros pedagógicos pueden dividirse en tres grupos:

1. Desarrollo de habilidades de programación.
2. Aprendizaje autónomo.
3. Evaluación optimizada.

**Figura 5-10:** Representación gráfica del modelo general producto del análisis de contenido

Entre los dos primeros logros existe una asociación recíproca, es decir, al promover el desarrollo de habilidades importantes de programación los estudiantes adquieren capacidades y conocimiento generando confianza en los alumnos permitiendo un proceso de aprendizaje con poca o nula intervención

de docentes y monitores. Sumando a esto, a través de la autonomía en el aprendizaje los estudiantes consolidan los conocimientos del curso, como pensamiento lógico y construcción de programas y cultivando habilidades que se derivan del uso de la plataforma. Adicionalmente, el desarrollo habilidades de programación capacita al estudiante para continuar con su proceso de aprendizaje al margen de recomendaciones o instrucciones del docente; lo que resulta en la formación de estudiantes capaces de resolver de manera eficiente problemas de programación.

Por otra parte, el tercer logro académico correspondiente a la optimización de la evaluación de los programas enviados por los estudiantes, se ve representado en dos aspectos: 1) la simplificación del proceso de presentación y envío de las diferentes actividades académicas del curso y 2) la objetividad y rapidez en la calificación de la solución enviada por el alumno. Cabe resaltar que, la calificación objetiva e inmediata se convierte en un rasgo distintivo de UNCode, dado que le aporta confiabilidad y eficacia al funcionamiento de la plataforma.

En contraste a los aspectos positivos evidenciados a partir de la experiencia de los estudiantes, aparecen también aspectos susceptibles de mejoras. Los puntos de mejora a los que se hace referencia pueden afectar y obstaculizar significativamente la experiencia del uso de la plataforma, lo que deteriora la percepción y valoración global de UNCode en general. Dentro de estos aspectos por mejorar destacan las fallas generales de funcionamiento, en particular, las eventuales caídas que sufre la plataforma cuando hay un alto volumen de usuarios conectados de manera simultánea. Los fallos en el funcionamiento dificultan directamente la presentación de las actividades académicas del curso, lo que afecta negativamente la calidad de evaluación optimizada evidenciada como bondad de la plataforma. En este sentido, a mediano plazo estas fallas pueden impactar la confiabilidad de UNCode, pues la percepción de inestabilidad en su funcionamiento puede hacer que los estudiantes y docentes terminen descartándola como herramienta de aprendizaje o recurso de enseñanza.

Continuando con los aspectos por mejorar, encontramos una percepción de insuficiencia en la orientación que ofrece UNCode. En este orden de ideas, la carencia de información o falta de claridad en la retroalimentación, puede potencialmente obstaculizar o ralentizar los procesos de aprendizaje autónomo. En algunos casos, se identificó que quienes reportan esta carencia también ven comprometida su capacidad de resolución de problemas de programación, pues no obtienen la suficiente instrucción que les permita identificar las mejores estrategias o prácticas para solucionar errores. Cabe resaltar, que en este aspecto pueden estar ocultas otras variables relevantes que permitirían ampliar los alcances explicativos del modelo. Un ejemplo son los estilos de aprendizaje de los usuarios, puesto que existen estudiantes que prefieren entornos de aprendizaje muy estructurados, con un alto grado de dirección e instrucción que deje poco espacio a la búsqueda activa de información. Sin embargo, otros alumnos consideran de mayor utilidad, o más motivante, un entorno abierto en el que se fomente la búsqueda autónoma de soluciones, por encima del seguimiento de instrucciones.

Finalmente, el tercer aspecto por mejorar es la inflexibilidad en el proceso de validación, la cual se

identificó en las tres preguntas analizadas. La falta de flexibilidad es un aspecto que merece atención por parte de los desarrolladores de la plataforma, dado puede tener un impacto directo sobre la calificación inmediata y objetiva; lo que puede deshacer lo que se ha consolidado por otras vías en cuanto a la confianza que genera la plataforma. La alta rigurosidad en la validación de las soluciones enviadas genera una percepción de falla en la plataforma al calificar como erróneos programas que cumplen con lo solicitado en los ejercicios pero poseen errores menores; a su vez esto afecta negativamente las calificaciones de los estudiantes, generando frustración en algunos y deteriorando la valoración positiva que se tiene sobre UNCode.

Por último, se realiza una comparación sistemática entre las respuestas de quienes aprobaron la asignatura y quienes no, la cual se hace con el objetivo de poner a prueba la hipótesis de que el haber aprobado o no el curso tiene un efecto en la experiencia de uso de UNCode, y por consiguiente, la valoración de esta. No obstante, en ninguna de los temas, categorías o códigos se encontraron diferencias atribuibles a la pertenencia a alguno de estos grupos. En un principio podría afirmarse que la diferencia entre grupos no es evidenciable, debido al desequilibrio en el número de integrantes de cada agrupación; pero se observa que a nivel discursivo tampoco hay diferencias sustanciales en función de la pertenencia al grupo. En otras palabras, al realizar un análisis de trama de sentido, en todos los temas, categorías y códigos fue posible encontrar enunciados con estructuras similares para ambos grupos. Un ejemplo es la categoría *falla en el objetivo educativo* de la primera pregunta, que se refiere a los reportes relacionados con que la plataforma no favorece el proceso de aprendizaje ni la adquisición de habilidades en programación de computadores. En específico, se encontraban respuestas de quienes aprobaron que señalaban “*En general me parece una herramienta que es útil para evaluar ejercicios de programación, al momento de aprender no hace mucha diferencia*” (sic.). De igual manera, del lado de quienes no aprobaron el curso se encontraban respuestas del tipo “*Para tener la capacidad de construir el conocimiento se debe obtener un Feedback puntual y objetivo, cosa que no ofrece la plataforma.*” (sic.). Estas referencias ilustran que los discursos de ambos grupos son esencialmente equivalentes, y por ende se concluye que el hecho de haber aprobado o reprobado la asignatura no constituye ninguna diferencia en la experiencia de uso de UNCode, ni en la valoración que se tenga de esta.

6 Discusión de resultados

En este capítulo se describen las actividades realizadas en la tercera fase de la metodología propuesta, que corresponde a la discusión de los resultados obtenidos. El capítulo inicia con la Sección 6.1, donde se hace la integración entre los resultados de la fase cuantitativa, descritos en el Capítulo 4 y los hallazgos de fase cualitativa, detallados en el Capítulo 5. La integración de resultados se realiza por medio de una comparación detallada de los resultados de cada fase, identificando vínculos entre medidas y conceptos. Posteriormente, en la Sección 6.2 se establece el significado que tienen las relaciones encontradas en la integración de resultados dentro del contexto de aprendizaje en programación de computadores. En específico, se aborda la pregunta de investigación planteada en al inicio de la investigación, la cual se intenta responder con base en los resultados obtenidos. El capítulo finaliza con la Sección 6.3, donde, a partir de la interpretación dada a los resultados, se genera una hipótesis que represente los fenómenos presentes en el contexto estudiado.

6.1. Integración de resultados cuantitativos y cualitativos

Los resultados de la fase cuantitativa son resumidos en la Figura 4-20, donde se evidencian las medidas y métricas con correlaciones significativas en relación con el desempeño académico de los estudiantes. Las medidas y métricas pueden ser clasificadas en cinco categorías, que corresponden a: veredictos obtenidos, preguntas cerradas en la encuesta de percepción, métricas de software, uso de herramientas e intentos de solución. Con respecto a la categoría correspondiente a los veredictos, siete evidencian una correlación positiva y tres una correlación negativa. Estos resultados se corroboran con los resultados de análisis de contenido de los datos cualitativos.

6.1.1. Resultados de veredictos de error

La dominancia del impacto positivo de la retroalimentación generada por los veredictos de error, es corroborada con los resultados de análisis de contenido. En específico, con respecto a la pregunta nombrada como RESPUESTA: Retroalimentación, la mayoría de estudiantes hacen referencia la utilidad que el uso de la plataforma en la adquisición de habilidades de resolución de problemas.

Entre los veredictos con correlación positiva están respuesta correcta, límite de tiempo excedido, respuesta incorrecta, error de compilación y límite de memoria excedido, adicionalmente se encuentra en este grupo la tasa de obtención de límite de tiempo excedido. Estas correlaciones positivas pueden ser vinculadas a algunas categorías identificadas en el análisis de contenido de la pregunta denominada RESPUESTA: Proceso de aprendizaje; en específico, los logros pedagógicos referentes a la resolución de problemas, calificación objetiva, evaluación optimizada y aprendizaje autónomo. En este orden de ideas, es posible que los veredictos que ofrece UNCode hagan que los estudiantes la perciban como una herramienta objetiva y eficiente para validar los programas construidos. Con respecto a la resolución de problemas y aprendizaje autónomo, es posible que los estudiantes obtengan información relevante por medio de los veredictos de error, lo que permite desarrollar habilidades de resolución de problemas de programación y fomentar el proceso de aprendizaje sin ayuda de la intervención de un docente.

Adicionalmente, las correlaciones positivas de los veredictos pueden estar vinculadas también a las bondades de la plataforma identificadas en la pregunta RESPUESTA: Proceso de aprendizaje; en específico, a las categorías de conocer los errores, consejos formativos, orientación y casos de prueba. En este sentido, los veredictos que ofrece la plataforma con cada intento de solución, permiten la identificación de los errores dentro del programa enviado. Sumado a esto, los veredictos ofrecen a los estudiantes consejos y dirección sobre los pasos a seguir para la corrección del programa construido. Por otra parte, la correlación positiva de los veredictos de respuesta correcta o incorrecta, se puede entender con las referencias de la categoría casos de prueba; puesto que estos veredictos generan retroalimentación sobre la similitud entre la respuesta obtenida por el estudiante y la respuesta establecida en los casos de prueba.

En el caso de la pregunta codificada como RESPUESTA: Calificación automática, los resultados que se pueden relacionar con los veredictos de correlación positiva son algunos logros pedagógicos identificados por los estudiantes como calificación objetiva, evaluación optimizada, retroalimentación inmediata, evaluación de capacidades y retroalimentación formativa. En el caso de la calificación objetiva y evaluación optimizada, es posible que los veredictos de error hagan que los estudiantes perciban el uso de la plataforma relevante, específicamente para evaluar de manera justa y óptima los programas diseñados. Con base a las referencias de retroalimentación inmediata y formativa, es posible afirmar que los veredictos generan una percepción positiva de la plataforma, para obtener una valoración sobre la calidad de la solución construida de manera automática y con información orientada a la corrección de errores. Sumado a esto, la categoría de evaluación de capacidades posiblemente se puede relacionar con los veredictos, dado que por medio de la retroalimentación obtenida los estudiantes pueden evaluar las capacidades y conocimientos que han adquirido en clase e identificar los aspectos que deben mejorar.

Por último, en la pregunta denominada RESPUESTA: Retroalimentación existen algunas categorías que se pueden asociar a los veredictos dentro del tema general de resolución de problemas. En es-

pecífico, la categoría de resolución de problemas, especificaciones en la retroalimentación, corregir errores y comparación con salidas esperadas. Con respecto a la primera categoría, similar que en la pregunta relacionada con el aprendizaje en programación de computadores, es posible que por medio de los veredictos los estudiantes obtengan por medio de los veredictos la información necesaria para cultivar habilidades de resolución de problemas de programación. Adicionalmente, los veredictos con correlación positiva se pueden asociar los estudiantes que hacen referencia a que las especificaciones dentro de la retroalimentación, que es ofrecida por la plataforma, promueven significativamente en el proceso de aprendizaje; también pueden estar asociadas a estudiantes que afirman que la retroalimentación de UNCode, facilita la corrección de los errores señalados. Sumado a esto, encontramos la categoría comparación de salidas esperadas, la cual se puede asociar directamente con los veredictos de respuesta correcta e incorrecta, dado que estos dos son los veredictos que surgen del símil entre la respuesta generada con el programa del estudiante y el resultado esperado en la prueba establecida por el docente. Por otra parte, dentro de esta pregunta surge el tema general señalar errores, el cual se puede vincular a los veredictos con correlación positiva, dado que cada tipo de veredicto ofrecido por la plataforma hace referencia a diferentes clases de errores que pueden estar presentes en el programa construido.

En contraste, las métricas en la categoría de veredictos con correlaciones negativas corresponden a la tasa de respuesta incorrecta, tasa de error de ejecución y tasa de límite de resultado excedido. Las tasas de veredicto con correlaciones negativas pueden ser relacionadas con los aspectos a mejorar identificados en la pregunta RESPUESTA: Proceso de aprendizaje, en específico las categorías de inflexibilidad, retroalimentación insuficiente, visualización y falla casos. En primera instancia, la inflexibilidad se puede asociar a la tasa de respuestas incorrectas, puesto que estos estudiantes afirman que la plataforma es demasiado estricta en la verificación de los programas, calificándolos como totalmente incorrectos por errores menores de forma, sin considerar la validez del contenido. La categoría de retroalimentación insuficiente se puede relacionar con las tasas de error de ejecución y de límite de resultados excedido, posiblemente indicando que estos veredictos no ofrecen suficiente información sobre el error del programa diseñado, haciendo que los estudiantes perciban estos veredictos como incompletos o no útiles. Finalmente, las categorías de errores en la visualización y fallos en los casos de prueba, se pueden asociar a las tres tasas de veredictos, haciendo referencia a que en ocasiones los veredictos no son correctamente visualizados o presentan fallas al ejecutar los casos de prueba, lo que no permite que el estudiante obtenga la retroalimentación de manera efectiva.

En el caso de la pregunta RESPUESTA: Calificación automática, algunas categorías dentro de los aspectos mejorables se pueden relacionar con las correlaciones negativas de las tasas de veredicto, específicamente las correspondientes a inflexibilidad, retroalimentación insuficiente, incomprendible y falla en validaciones. Las categorías en esta pregunta son muy similares a la primera pregunta, donde la categoría de inflexibilidad y falla en validaciones se puede relacionar de igual manera con la tasa de respuestas incorrectas, debido al nivel alto de rigidez al evaluar las soluciones enviadas como totalmente incorrectas. La categoría de retroalimentación insuficiente también se asocia a estas tasas

como fue mencionado antes, posiblemente porque la información sobre los errores que representan estos veredictos no es clara para los estudiantes, generando una percepción negativa. La categoría donde se hace referencia a que el proceso de retroalimentación es en algunas ocasiones incomprendible, se pueden relacionar con las correlaciones negativas, lo que puede indicar que estos veredictos requieren mejorar el detalle o explicación que se da sobre la valoración de los programas. Adicionalmente, en esta misma pregunta se pueden relacionar las tasas de veredicto con correlación negativa, con las referencias a dificultades en la implementación, en específico la falta de la participación docente; debido a que los estudiantes consideran que la información aportada por los veredictos debería ser complementada con la explicación del profesor, con el objetivo de asegurar que el estudiante tenga claro a que hace referencia el error señalado.

Finalmente, en la pregunta RESPUESTA: Retroalimentación, se pueden relacionar también algunos aspectos mejorables identificados en la fase cualitativa, con las tasas de veredicto con correlación negativa. En primer lugar, la categoría de detalles menores de sintaxis, que se asemeja a la categoría de inflexibilidad identificada en anteriores preguntas, tiene relación con la tasa de respuestas incorrectas, debido a que los estudiantes hacen referencia a ocasiones en las que debido a errores menores de forma en el código, la solución es valorada como incorrecta en su totalidad. También se puede encontrar un vínculo con las categorías de falta de claridad y orientación insuficiente, de manera similar con la categoría de retroalimentación insuficiente de anteriores preguntas, porque es posible que los estudiantes consideren que estos tipos de veredicto carecen de explicación o detalle, evitando que la información para corregir el programa sea transmitida. Otra categoría relacionada es fallas en el funcionamiento, dado que es posible que los errores de ejecución de la plataforma, sean un gran obstáculo para los estudiantes en la obtención de retroalimentación formativa. En cuanto al tema de aspectos mejorables también se puede asociar indirectamente la categoría identificación autónoma de errores, dado que la percepción de retroalimentación insuficiente, debido a los otros aspectos por mejorar como falta de claridad, fallas de funcionamiento e inflexibilidad, generan que los estudiantes se vean obligados a buscar maneras alternas para la identificación y corrección de errores fuera de la plataforma. Finalmente, dentro de esta pregunta también se puede asociar el tema general de condiciones iniciales, teniendo en cuenta que la falta de entendimiento de la retroalimentación de los veredictos puede estar relacionada con una carencia en las especificaciones de las instrucciones iniciales del problema.

6.1.2. Resultados de intentos realizados

Con respecto a la categoría de intentos, los datos cuantitativos evidencian dos métricas con correlaciones positivas. La primera métrica es el total de intentos de solución realizados por los estudiantes, lo que se puede relacionar con los resultados de la fase cualitativa donde se hace referencia a logros pedagógicos obtenidos por medio del uso de la plataforma en la pregunta RESPUESTA: Proceso de aprendizaje. En específico, las categorías ejercicios estimulantes y presentación de las actividades,

puesto que la facilidad que aporta la plataforma para planteamiento de variedad de actividades académicas, las cuales sean desafiantes para los estudiantes, puede influir positivamente en la cantidad de soluciones enviadas en cada actividad. También se pueden relacionar algunas categorías presentes en el tema de bondades de la plataforma, como facilidad de uso, práctica constante, disponibilidad en línea y espacio de trabajo. En el caso de facilidad de uso, la relación hace referencia a que los estudiantes que perciben UNCode como una herramienta de fácil funcionamiento, es posible tiendan a utilizar la plataforma activamente mediante el envío de una cantidad alta de soluciones. Con respecto a las demás categorías, se puede establecer una relación con el número de intentos, dado que la constante disponibilidad de la plataforma, genera un espacio de trabajo exclusivo para el estudiante, el cual permite la práctica constante de ejercicios inclusive fuera de clase, lo que es probable se vea reflejado en un alto número de intentos registrados. También, dentro de esta pregunta se puede relacionar el tema general ambiente, porque al ser la plataforma didáctica y estéticamente llamativa, es mayor la probabilidad de incentivar a los estudiantes a construir y enviar una mayor cantidad de intentos de solución. Por último, en la pregunta RESPUESTA: Calificación automática surge el tema general facilidad de uso, similar a la categoría con el mismo nombre de la pregunta anterior, la cual se puede vincular al número de intentos registrados, ya que cuando los alumnos se sienten cómodos con la plataforma es más probable que la utilicen, intentando dar solución a los ejercicios planteados.

En contraste, algunos resultados de la fase cualitativa también hacen referencia a aspectos que algunos estudiantes consideran deben ser mejorados dentro de la plataforma, los cuales pueden afectar negativamente la cantidad de soluciones enviadas por los estudiantes. En general los aspectos a mejorar identificados en la pregunta RESPUESTA: Proceso de aprendizaje, que se pueden vincular son fallas generales, plataforma caída, pérdida de información y registro, que hace referencia a problemas con el ingreso a UNCode, los cuales pueden ser interpretados como obstáculos para el envío de soluciones. Otras categorías que se pueden relacionar son incompatibilidad y sustituible, puesto que la falta de compatibilidad de la plataforma con programas desarrollados en otros entornos de desarrollo externos, hace que los alumnos opten por herramientas externas para el desarrollo, evaluación y corrección de programas, sustituyendo el uso de UNCode por otras plataformas y disminuyendo significativamente la cantidad de envíos realizados. Adicionalmente, otra de las categorías que se pueden relacionar con los envíos es calificación ineficiente, puesto que pueden generar una desmotivación en los estudiantes al realizar intentos de solución en la plataforma, dado que la calificación no va a ser acertada. Finalmente, dentro de esta misma pregunta se puede establecer un vínculo entre el número de intentos y las referencias del tema general Colab, dado que el uso de Google Collaboratory como en la herramienta para la construcción de programas, hace que los estudiantes se concentren en construir y evaluar sus soluciones fuera de UNCode y utilizar la plataforma solo para enviar el programa final, lo que puede limitar bastante la cantidad de intentos registrados en UNCode.

Con respecto a la pregunta RESPUESTA: Calificación automática, también se pueden relacionar la categoría fallas generales identificadas como aspectos mejorables en la calificación automática, similar a la categoría con el mismo nombre de la pregunta anterior, considerando que las fallas en el funciona-

miento de la plataforma pueden dificultar o incluso impedir el envío de soluciones en las actividades del curso. Sumado a esto, algunas de las referencias correspondientes a las categorías de dificultades de implementación y poco uso de la plataforma se pueden asociar a la cantidad de intentos registrados por estudiante, debido a que los obstáculos que afectan negativamente el uso de la herramienta durante el curso, necesariamente se traducen en el poco uso de UNCode para la presentación de actividades académicas, limitando la interacción de los alumnos con la plataforma y reduciendo considerablemente los envíos realizados.

La segunda métrica con correlación positiva en esta categoría es el tiempo entre intentos, la cual se puede relacionar con la categoría calificación inmediata que surge tanto en los logros pedagógicos identificados en la pregunta RESPUESTA: Proceso de aprendizaje como en RESPUESTA: Calificación automática, puesto que la valoración numérica automática de las soluciones enviadas, permite que los estudiantes enfoquen su tiempo en la corrección del programa construido antes que en su evaluación. El tiempo entre intentos también se puede vincular con una de las bondades de la plataforma identificada en la pregunta RESPUESTA: Proceso de aprendizaje, en específico la categoría escribir y corregir; indicando posiblemente que los estudiantes utilizan la plataforma tanto para construir como para corregir los programas enviados, haciendo que el promedio de tiempo entre intentos sea mayor. En contraste, en la misma pregunta algunos estudiantes hacen referencia a aspectos mejorables que contradicen la correlación positiva del tiempo entre envíos. En específico, un aspecto a mejorar es la categoría velocidad de procesamiento, puesto que las demoras de la plataforma en la revisión de los programas disminuyen el tiempo disponible para que los estudiantes construyan y modifiquen sus soluciones. Sin embargo, la cantidad de referencias en la categoría velocidad de procesamiento es menor a 1 % de los estudiantes encuestados, lo que puede indicar que esta percepción no es generalizada sino específica para unos pocos alumnos.

6.1.3. Resultados de uso de herramientas

Con respecto a la categoría de uso de herramientas, los resultados de la fase cuantitativa muestran que siete medidas tienen correlación significativa con el desempeño de los estudiantes. La tasa de uso de custom input es la única que tiene correlación positiva, la cual se puede relacionar con las referencias de herramientas de UNCode identificadas en la pregunta RESPUESTA: Proceso de aprendizaje. Este resultado se puede relacionar directamente con la categoría custom input, indicando que probablemente los estudiantes que identifican los beneficios de realizar pruebas personalizadas para la verificación de los programas construidos, tienden a utilizar más custom input que las demás herramientas.

Por el contrario, las medidas y métricas restantes del uso de herramientas tienen correlación negativa con el desempeño académico; las cuales corresponden al número de accesos a linter, multiple languages code, python tutor y custom input, junto con la tasa de uso de linter y multiple languages

code. Las correlaciones negativas se pueden vincular con la categoría herramientas no disponibles, la cual hace referencia a aspectos mejorables de la plataforma identificados en la pregunta RESPUESTA: Proceso de aprendizaje; ya que las fallas en el funcionamiento de las herramientas no permiten que los estudiantes evidencien los beneficios de su uso en el proceso de aprendizaje. No obstante, los resultados de la fase cualitativa, en la misma pregunta, también evidencian referencias positivas sobre el uso de las herramientas, entre las cuales están las categorías herramientas de UNCode, python tutor, custom input y linter; lo que indica que algunos estudiantes perciben estas herramientas como instrumentos positivos que ofrece la plataforma, pero posiblemente dificultades en su uso evitan que logren sus objetivos pedagógicos. Adicionalmente, dentro de las bondades de la plataforma identificadas en la misma pregunta, surge la categoría lenguajes la cual contrasta con la correlación negativa de multiple languages code y su tasa de uso. A pesar de la diferencia entre resultados, la cantidad de alumnos que hacen referencia a la posibilidad de utilizar diferentes lenguajes de programación es menor al 1 %, indicando que son pocos estudiantes los que evidencian un beneficio en el uso de esta herramienta. Finalmente, dentro los logros pedagógicos a los que se hace referencia en la pregunta RESPUESTA: Calificación automática surge la categoría seguimiento de calificaciones, la cual es de interés porque la correlación del uso de estadísticas de usuario no evidencia ser significativa en la fase cuantitativa. La oposición entre resultados, puede estar relacionada con que menos del 2 % de los estudiantes encuestados hacen referencia al seguimiento de calificaciones como logro pedagógico de la plataforma.

6.1.4. Resultados de preguntas percepción

La fase cuantitativa evidencia que las respuestas a las tres preguntas cerradas de la encuesta de percepción, correspondientes a PREGUNTA: Proceso de aprendizaje, PREGUNTA: Calificación automática y PREGUNTA: Retroalimentación, tienen correlaciones positivas significativas con el desempeño de los estudiantes. Las correlaciones positivas son corroborados con los resultados de la fase cualitativo, puesto que en la tres preguntas abiertas de la encuesta de percepción la mayoría de estudiantes hace referencia a los aspectos positivos de la plataforma, en comparación con los que resaltan los aspectos a mejorar. En específico, la pregunta relacionada con la utilidad del uso de UNCode en el aprendizaje de programación de computadores, evidencia 31.2 % de referencias clasificadas en el tema logros pedagógicos y 27.2 % en bondades de la plataforma, las cuales representan que más de la mitad de los alumnos encuestados identifican los beneficios del uso de la plataforma. En el caso de la pregunta sobre la utilidad de UNCode en la obtención de calificación automática, la temática con mayor número de referencias es logros pedagógicos con 78.6 % de los datos. Adicionalmente, en el caso de la pregunta relacionada con utilidad de la retroalimentación automática de UNCode, las respuestas de los estudiantes en su mayoría hacen referencia a como el uso de la plataforma promueve la resolución de problemas con el 41.6 % de los datos, en contraste con los alumnos que resaltan los aspectos a mejorar en la retroalimentación que corresponden al 31.8 %. Finalmente, los resultados cuantitativos evidencia que la respuesta a la pregunta: “la prueba de programas usando entradas pro-

vistas por el usuario (custom input) fue útil para su aprendizaje en esta asignatura” codificada como A_custom input, tiene correlación significativa positiva con el rendimiento de los estudiantes; lo que se puede vincular con el 3 % de las respuestas en la categoría custom input de las herramientas de UNCode identificadas en la pregunta RESPUESTA: Calificación automática.

6.1.5. Resultados de métricas de software

Finalmente, con respecto a la categoría métricas de software, en la fase cuantitativa se obtiene que tres métricas tienen correlación positiva y una correlación negativa con el desempeño académico. Entre las métricas con correlación positiva está el conteo de tokens, las líneas de código y la complejidad ciclomática, y la métrica con correlación negativa es el índice de mantenibilidad (MI). La correlación positiva del conteo de tokens y líneas de código se relaciona con el 5 % de los estudiantes que en la pregunta RESPUESTA: Proceso de aprendizaje, señalan como logro pedagógico la categoría de ejercicios estimulantes, ya que el planteamiento de actividades académicas desafiantes promueve la construcción de programas más largos, por medio del uso de palabras clave y sentencias de código. Por otra parte, la correlación positiva de la complejidad ciclomática y la correlación negativa del índice de mantenibilidad, pueden estar vinculadas con el 3 % de respuestas que identifican como logro pedagógico la categoría habilidades de programación; puesto que la destreza de los estudiantes para construir soluciones correctas, deben ser reflejadas en el uso de estructuras lógicas y en el diseño de programas capaces de adaptarse a modificaciones.

6.2. Interpretación de resultados integrados

La pregunta de investigación ¿cómo los métodos mixtos de investigación aplicados en la analítica del aprendizaje, pueden generar un mejor entendimiento de las relaciones existentes entre las variables generadas a lo largo del proceso de aprendizaje y el rendimiento académico de los estudiantes en la asignatura programación de computadores?, se puede responder por medio de los resultados integrados que son resumidos a continuación.

En primer lugar, las correlaciones positivas del número de respuestas correctas, tasa de éxito, la cantidad de errores de límite de memoria excedido, errores de compilación, junto con el número de veredictos y tasas de obtención de límite de tiempo excedido; están relacionadas con alumnos que perciben la plataforma como una fuente de retroalimentación formativa, la cual no se limita a identificar errores porque ofrece también recomendaciones orientativas para la corrección del programa construido, generando habilidades de resolución de problemas y aprendizaje autónomo. Lo anterior indica que, por medio de estos veredictos, los estudiantes posiblemente adquirieron conocimientos suficientes para resolver con éxito las actividades del curso; lo que se ve reflejado positivamente en el desempeño académico del alumno. Estos resultados corroboran los hallazgos del trabajo de Azcona *et al.*

(2019), el cual también evidencia que el porcentaje acumulado de ejercicios correctos tiene un coeficiente de correlación significativo de 0.67 con respecto al desempeño académico del estudiante. Adicionalmente, la correlación positiva del número de respuestas incorrectas, esta relacionada con las referencias de casos de prueba y comparación con salidas esperadas; lo que indica que el uso de pruebas estandarizadas para la evaluación automática de los programas es efectiva como retroalimentación formativa, beneficiando el rendimiento académico de los estudiantes.

Por otra parte, las tasas de error de ejecución y límite de resultado excedido tienen correlaciones negativas, las cuales están relacionadas con la percepción de retroalimentación insuficiente, falla en la visualización o en los casos de prueba, falta de claridad en las condiciones iniciales, carencia de participación docente e identificación autónoma de errores. Estos resultados posiblemente indican que algunos problemas de funcionamiento de la plataforma hacen que los estudiantes perciban la evaluación automática como un proceso confuso y poco confiable, considerando que el acompañamiento del docente es aun esencial en el proceso de evaluación y aclaración de dudas; lo que genera dificultades para obtener retroalimentación efectiva a partir de estos veredictos y impacta negativamente el rendimiento académico. Sumado a esto, la tasa de respuestas incorrectas también tiene una correlación negativa, lo cual está asociado con las referencias de inflexibilidad como aspecto a mejorar en la plataforma; lo que probablemente indica que debido a que la evaluación de los programas es demasiado estricta, valorando como incorrectas soluciones con errores menores de sintaxis, los estudiantes no obtienen orientación a partir de este veredicto, evitando que den solución a las actividades académicas y disminuyendo sus calificaciones. En este orden de ideas, el rendimiento académico del curso podría mejorar al enriquecer la manera en la que se comunica el objetivo del ejercicio, y asegurando que sea clara la información que ofrecen los veredictos de la plataforma.

En segundo lugar, la tasa de uso de custom input tiene una correlación positiva, la cual se relaciona con las referencias de custom input en el grupo de herramientas de UNCode; esto indica que los estudiantes que perciben como una herramienta útil la opción de evaluar los programas construidos con pruebas personalizadas, tienden a preferir utilizar esta herramienta, dado que tienen las habilidades para diseñar pruebas para depurar la solución propuesta y obtener buenos desempeños académicos.

En contraste, la cantidad de accesos a python tutor y custom input, junto con el número accesos y tasa de uso de linter y multiple languages code, tienen un impacto negativo en el rendimiento del estudiante, la cual se puede relacionar con las referencias de herramientas no disponibles. Esta asociación indica que las fallas en el funcionamiento de las herramientas, posiblemente desmotivan el uso de las mismas, privando a los estudiantes de instrumentos útiles para la resolución de dudas y orientación, lo que puede resultar en bajas calificaciones. No obstante, estas correlaciones negativas se ven refutadas por las referencias donde se identifican las herramientas python tutor, custom input y linter, como elementos significativos dentro de la plataforma, y como bondades de la plataforma la diversidad de lenguajes de programación. Estos resultados concuerdan con los hallazgos de los trabajos realizados por Restrepo-Calle *et al.* (2020) y Ramírez-Echeverry *et al.* (2022), donde también se

analizan las percepciones de los estudiantes sobre el uso de UNCode como una plataforma de apoyo en el proceso de aprendizaje; dentro de estas investigaciones se evidencia que los alumnos reconocen como valor agregado de la plataforma la herramienta para la visualización de la ejecución del código (*Python tutor*), la cual está asociada con la identificación y corrección de errores, adicionalmente los alumnos resaltan las herramientas de verificación de buenas prácticas de programación (*Linter*) y pruebas con entradas personalizadas (*Custom input*). Esta contradicción en los resultados, puede deberse a que los alumnos que tienen dudas fundamentales o dificultades en la comprensión de los temas buscan ayuda en estas herramientas, pero la utilidad de la información adquirida depende del conocimiento y las habilidades del estudiante para poner en práctica las funciones de las herramientas. En el caso específico de *Linter*, existe la posibilidad de que la correlación negativa se deba a que esta herramienta está diseñada para promover buenas prácticas de programación, las cuales en ocasiones no son las estrategias más efectivas para dar solución a ejercicios de programación. Adicionalmente, el uso de estadísticas de usuario tiene una correlación no significativa, lo que concuerda con las investigaciones realizadas por Zacharis (2015) y Macfadyen y Dawson (2010), donde la cantidad de accesos a la herramienta de calificaciones no evidencia una correlación significativa con la calificación. Este resultado se puede relacionar con un porcentaje pequeño de respuestas que señalan el seguimiento de calificaciones dentro los logros pedagógicos, lo que indica que probablemente para la mayoría de estudiantes la plataforma no evidencian el beneficio directo que tiene ser conscientes de sus logros académicos, por lo cual esta herramienta no tiene un impacto evidenciable en el proceso de aprendizaje.

En tercer lugar, el total de intentos realizados por estudiante tiene una correlación positiva con la calificación final, la cual se puede asociar con aspectos positivos percibidos por los estudiantes como ejercicios estimulantes, presentación de actividades, facilidad de uso, práctica constante, disponibilidad en línea, espacio de trabajo y ambiente. El vínculo entre resultados, indicar que posiblemente los estudiantes con alto desempeños utilizan la plataforma como fuente de retroalimentación para mejorar las soluciones construidas, realizando múltiples intentos en la misma actividad, puesto que la plataforma genera un espacio de trabajo que facilita la presentación de actividades académicas, la cual es amigable con el usuario, permite el planteamiento de ejercicios estimulantes y fomenta la práctica constante, dado que está disponible inclusive fuera del espacio de clase. Este resultado concuerda con la investigación de Zacharis (2015), donde se encuentran una correlación débil (0.2 a 0.39) entre la cantidad de actividades entregadas durante el curso y la calificación final de curso. En contraste, existen algunas referencias que indican aspectos de UNCode que deben ser mejorados, los cuales pueden impactar negativamente en la cantidad de intentos realizados por los alumnos, entre los cuales están las fallas generales de funcionamiento, la incompatibilidad con programas desarrollados fuera de la plataforma y el poco uso en la metodología de clase. No obstante, el porcentaje de estas referencias en estas categorías es pequeño, indicando que la mayoría de estudiantes no evidencian estas fallas y el impacto negativo no es significativo en el rendimiento de los estudiantes.

Por otra parte, la correlación positiva del tiempo entre intentos se puede asociar con el grupo de es-

tudiantes que resalta como característica positiva de UNCode la calificación inmediata y el aspecto escritura y corrección. Estos resultados concuerdan con los hallazgos de Andergassen *et al.* (2014), los cuales obtienen una correlación positiva de 0.18 entre el tiempo promedio de diferencia entre repeticiones (i.e. intentos) de ejercicios con la calificación del examen final. Estos resultados indican que debido a la rapidez del proceso de evaluación en la plataforma, posiblemente los alumnos con buenos rendimientos académicos invierten la mayoría de su tiempo en la construcción y corrección de la solución entre cada envío realizado. Adicionalmente, las referencias a aspectos que deben ser mejorados como la velocidad del procesamiento de la plataforma, los cuales pueden influir negativamente en la distribución del tiempo de los estudiantes, no tienen la cantidad suficiente de respuestas para generar un efecto significativo en el desempeño de los alumnos. Otro de los resultados obtenidos son las correlaciones positivas de las percepciones en las tres preguntas cerradas (PREGUNTA: Proceso de aprendizaje, PREGUNTA: Calificación automática y PREGUNTA: Retroalimentación), las cuales son corroboradas con las respuestas de las preguntas abiertas, donde la mayoría de estudiantes identifican logros pedagógicos y las bondades de la plataforma, esto probablemente indica que los usuarios que tienen una experiencia positiva con la plataforma, tienden a identificar y aprovechar los beneficios de la plataforma, logrando un buen desempeño académico.

Finalmente, las correlaciones positivas del conteo de tokens y las líneas de código se encuentra asociado con las referencias de la categoría ejercicios estimulantes, esto quiere decir que los estudiantes que perciben las actividades propuestas como desafiantes, son los alumnos que tienen conocimientos en el uso adecuado de palabras clave y sentencias de código, para la construcción de programas de alto contenido, lo que se traduce la resolución exitosa de problemas y por ende buenas calificaciones. Estos resultados pueden estar relacionados con el nivel de complejidad de los temas de la asignatura, puesto que el curso Programación de Computadores es de tipo introductorio, es decir, que la mayoría de estudiantes no tiene conocimientos previos de programación. En este sentido, los estudiantes que logran adquirir y aplicar los conocimientos del curso, logran diseñar programas de mayor complejidad y longitud que los estudiantes de bajos desempeños, los cuales al no tener conocimientos previos construyen soluciones sencillas y cortas. Este resultado es relevante, porque en cursos de temas avanzados de programación el comportamiento puede ser opuesto, ya que el objetivo es diseñar programas cada vez más eficientes. El resultado de la métrica de complejidad, contradice los descubrimientos del trabajo de Vahdat *et al.* (2015), donde se evidencia que la complejidad ciclomática de los programas construidos está correlacionada negativamente con el promedio de notas finales de los estudiantes. Adicionalmente, la correlación negativa de índice de mantenibilidad se puede asociar con las respuestas que señalan las habilidades de programación como logros académicos de UNCode. La relación entre resultados indica que las habilidades de programación adquiridas por los estudiantes debido al uso de UNCode, son representadas por la construcción de códigos de baja mantenibilidad, lo que probablemente favorece el éxito en la solución de problemas de programación y por ende favorece el rendimiento académico del estudiante.

En conclusión, la respuesta a la pregunta de investigación ¿cómo los métodos mixtos de investiga-

ción aplicados en la analítica del aprendizaje pueden generar un mejor entendimiento de las relaciones existentes entre las variables generadas a lo largo del proceso de aprendizaje y el rendimiento académico de los estudiantes en la asignatura programación de computadores? corresponde al hecho de que los resultados obtenidos evidencian que el uso de métodos mixtos permite complementar los hallazgos de la fase cuantitativa por medio de las observaciones de la fase cualitativa. En este sentido, en la mayoría de los casos los datos cualitativos, los cuales corresponden a las percepciones de los estudiantes, corroboran y amplían los resultados de los datos cuantitativos, es decir, las interacciones registradas de los estudiantes con la plataforma UNCode. La concordancia entre los resultados de ambas fases permite generar varias hipótesis sobre las razones subyacentes de los comportamientos observados y los procesos de aprendizaje de los alumnos, las cuales estén basadas tanto en resultados cuantitativos como cualitativos. En otros casos, el enfoque mixto evidencia contradicciones entre los resultados de ambas fases (p. ej. resultados del uso de herramientas), lo cual permite identificar temas de interés fuera del alcance de la investigación y generar preguntas que pueden ser abordadas en trabajos futuros.

La principal contribución de este trabajo es el marco metodológico propuesto para la aplicación de analítica del aprendizaje en cursos de programación de computadores, el cual está basado en métodos mixtos, donde se especifican las actividades desde la recolección de los datos, tanto de tipo cuantitativo como cualitativo, hasta la integración y discusión de resultados. Cabe resaltar que, las actividades de la metodología son descritas de manera general con el objetivo de ser utilizadas como referencia en investigaciones similares futuras. Adicionalmente, los resultados de esta investigación contribuyen al conocimiento de cómo la retroalimentación formativa automática puede ser favorable en el proceso de aprendizaje por sí sola, en el caso de algunos estudiantes. Sin embargo, para otro grupo de alumnos este tipo de retroalimentación necesita ser complementada con la orientación de un instructor, para cumplir su objetivo. Adicionalmente, se evidencia como la posibilidad de realizar pruebas personalizadas diseñadas por los estudiantes tiene beneficios en la construcción de soluciones correctas. Sumado a esto, se observa que la técnica de los alumnos con calificaciones altas es realizar la mayor cantidad de intentos de solución y utilizar la mayoría del tiempo en la corrección del programa diseñado. Por otra parte, se evidencia la importancia de asegurar que los estudiantes entiendan la utilidad del uso de la plataforma en la metodología de clase, puesto que esto aumenta la probabilidad de que los alumnos aprovechen la herramienta y aumenten su rendimiento académico. Finalmente, con base en los resultados se logra entender que las habilidades de programación se pueden generar, enfocando la enseñanza en la aplicación e importancia de las métricas de software en la construcción de programas de buena calidad.

6.3. Generación de hipótesis

A partir de la integración de resultados de ambas fases de la metodología mixta utilizada en la investigación y la interpretación de los mismos, se plantea una hipótesis que describa los comportamientos

evidenciados. En primer lugar, los estudiantes de la asignatura Programación de Computadores interactúan con la plataforma UNCode durante el desarrollo del curso, por medio: un espacio de trabajo que permite realizar múltiples intentos de solución a ejercicios de programación, la retroalimentación formativa ofrecida a través de veredictos de error y las herramientas disponibles dentro de la plataforma. En este sentido, muchos estudiantes perciben la plataforma como un espacio de trabajo disponible inclusive fuera de las actividades de clase. La disponibilidad en línea de la plataforma permite que los estudiantes puedan practicar constantemente sus conocimientos en programación de computadores, generando capacidades de aprendizaje autónomo y habilidades de programación. Las habilidades en la resolución de problemas de programación se ven representadas en las métricas de software de los programas construidos por los estudiantes. Adicionalmente, durante el proceso de calificación y retroalimentación automática que ofrece la plataforma, que se da mediante los veredictos de error, aportan información significativa para los estudiantes, dado que no se reducen a identificar tipos de error presentes en el programa, sino que genera orientación sobre como corregirlos. A pesar de que los estudiantes identifican los beneficios de las herramientas de la plataforma como el seguimiento paso a paso de la ejecución del código, realización de pruebas personalizadas y señalamiento de errores de buenas prácticas de programación, pero su uso es desincentivado debido a fallas en el funcionamiento de la plataforma. Finalmente, la percepción de los estudiantes con respecto a la utilidad de todos los aspectos que ofrece la plataforma, con respecto al proceso de aprendizaje en el curso, la calificación automática y la retroalimentación inmediata, influyen en el beneficio que los estudiantes obtienen de la herramienta.

A partir de los hallazgos de esta investigación surgen ideas de posibles trabajos futuros que logren ampliar los descubrimientos del presente trabajo. En primer lugar, se propone realizar una investigación con diseño experimental, a partir de la cual se intente evidenciar el impacto que tiene el uso de UNCode como herramienta del curso en el proceso de aprendizaje de los estudiantes. El estudio puede emplear pruebas estadísticas para verificar si existe una diferencia significativa entre el promedio de las calificaciones finales del grupo experimental, es decir, un curso de programación de computadores donde todas las actividades sean realizadas dentro de la plataforma y el grupo de control, que correspondería a un curso de la misma asignatura donde la plataforma no se utilice. Por otra parte, las correlaciones significativas evidenciadas pueden promover el diseño y ejecución de intervenciones educativas dentro del curso, lo que corresponde a la etapa final de la metodología cíclica de analítica del aprendizaje propuesta por Carter *et al.* (2019). El desarrollo de intervenciones consiste en tomar decisiones en el contexto educativo estudiado, donde se comparte con los estudiantes información, orientación o retroalimentación, con el objetivo de influenciar positivamente su comportamiento (Carter *et al.*, 2019). Por ejemplo, algunas de las intervenciones pueden ser, aumentar la visualización tanto de la descripción de cada veredicto de error como las orientaciones para la corrección de los mismos, con el objetivo de mejorar las percepciones de algunos alumnos de insuficiencia en la retroalimentación. Otra intervención educativa puede ser, la recomendación del uso de una herramienta específica, según su funcionalidad, al obtener un veredicto de error determinado (p. ej. sugerir utilizar *Python tutor* al obtener errores de compilación o ejecución), con el fin de promover el uso de

las herramientas de la plataforma y analizar el efecto que tienen estas sugerencias en las percepciones y/o desempeños académicos de los alumnos. El impacto de estas intervenciones también puede ser evaluado por medio de un diseño experimental, el cual busque diferencias estadísticas entre un grupo que implemente alguna de las intervenciones y un grupo de control.

7 Conclusiones y trabajos futuros

7.1. Conclusiones

Las aplicaciones de analítica del aprendizaje aprovechan nuevas formas de datos digitales educativos y facilitan la toma de decisiones que mejoran tanto las metodologías de enseñanza como los procesos de aprendizaje. Por esto, varios autores han planteado metodologías para la aplicación de la analítica del aprendizaje, iniciando con la recolección de los datos hasta la aplicación de intervenciones con impacto educativo positivo. Teniendo en cuenta todos los beneficios que tiene la analítica de aprendizaje, sus aplicaciones se han realizado en contextos académicos de diferentes áreas del conocimiento. Un grupo de investigaciones se ha concentrado en aplicar la analítica de aprendizaje en cursos de programación de computadores, en específico en la identificación de variables del proceso de aprendizaje relacionadas con el desempeño académico de los estudiantes.

Sin embargo, algunos investigadores han identificado como desafío en las aplicaciones de analítica de aprendizaje la carencia en el uso de datos cualitativos o metodologías mixtas. La principal causa de esta carencia, es una mayor disponibilidad y facilidad de recolección de datos cuantitativos en cursos de programación, lo que limita los hallazgos de las investigaciones que utilizan solo este tipo de datos. Con el objetivo de responder a este desafío, en esta tesis de maestría se propone un diseño metodológico mixto secuencial explicativo no experimental de analítica del aprendizaje, el cual está conformado por tres fases principales. La primera corresponde a la preparación, transformación y análisis de datos cuantitativos. La segunda fase constituye la recolección de datos cualitativos, junto con su análisis de contenido. Y en la fase final, se hace tanto la integración de los resultados de las fases anteriores como la discusión e interpretación de los hallazgos. En este trabajo, la metodología diseñada es utilizada para el estudio de datos cuantitativos históricos de dos (2) años, resultantes de la interacción de los estudiantes de la asignatura de Programación de Computadores en la Universidad Nacional de Colombia con una plataforma de evaluación y retroalimentación automática de ejercicios de programación utilizada en el curso; por otra parte, la fase cualitativa se desarrolla por medio de los datos recolectados de encuestas de percepción sobre el uso de la plataforma, realizadas a un subgrupo de la misma muestra poblacional de estudiantes.

Los resultados de la investigación evidencian algunas diferencias cuantitativas significativas en la percepción que tienen los estudiantes que aprueban y reprueban la asignatura con respecto a variables del conjunto de datos. Los resultados evidencian que el número total de intentos realizados, el tiempo promedio entre intentos, cinco veredictos (respuesta correcta e incorrecta, límite de tiempo excedido,

límite de memoria excedido y error de compilación), dos tasas de veredictos (tasa de éxito y límite de tiempo excedido), la tasa de uso de la herramienta custom input, cuatro preguntas cerradas de la encuesta de percepción (PREGUNTA: Aprender a programar, PREGUNTA: Calificación automática, PREGUNTA: Retroalimentación inmediata y A_custom input) y tres métricas de software (conteo de tokens, número de líneas de código y complejidad ciclomática) tienen una correlación positiva significativa con el desempeño académico de los estudiantes. Sin embargo, los datos cualitativos no evidencian esa diferencia de percepciones entre las categorías generadas acorde con el desempeño académico. La diferencia de los resultados de cada fase indican que probablemente existen algunos limitantes en el conjunto de datos proveniente de las encuestas de percepción, en específico, con respecto al reducido número de estudiantes de la categoría de alumnos que no aprueban la asignatura.

La correlación positiva del total de intentos realizados por estudiante coincide con las menciones de los estudiantes de beneficios de la plataforma como ejercicios estimulantes, presentación de actividades, facilidad de uso, práctica constante, disponibilidad en línea, espacio de trabajo y ambiente. Con respecto a la correlación positiva del tiempo promedio entre intentos realizados, los datos cualitativos evidencian resultados relacionados como respuestas sobre calificación inmediata y la utilidad de la plataforma para escribir y corregir el código diseñado. Las correlaciones positivas de los veredictos de error son corroboradas con las menciones de los estudiantes de la utilidad de la plataforma para la resolución de problemas, calificación objetiva y optimizada, aprendizaje autónomo, identificación y corrección de errores, orientación, retroalimentación específica, formativa e inmediata, evaluación de capacidades, casos de prueba y comparación con salidas esperadas. Las correlaciones positivas del uso de custom input y de la pregunta A_custom input, se ven corroboradas por las menciones de los estudiantes de esta herramienta como una propiedad relevante de la plataforma para promover el proceso de aprendizaje. Las correlaciones positivas de las preguntas abiertas coinciden con el hecho de que la mayoría de las menciones en las preguntas abiertas, se refieren a aspectos positivos de la plataforma, en específico, temas como logros pedagógicos que se obtienen con el uso de UNCode, las bondades de la plataforma y la utilidad para resolución de problemas. Finalmente, las correlaciones positivas de las métricas de software se ven reflejadas en las menciones de los estudiantes con respecto a bondades de la plataforma como ejercicios estimulantes y habilidades de programación.

La integración de los resultados de correlaciones positivas indican que posiblemente los estudiantes con altos desempeños logran utilizar los veredictos de error obtenidos como fuente de retroalimentación formativa, lo que promueve las habilidades de resolución de problemas y aprendizaje autónomo; y este mismo grupo de alumnos evidencia una tendencia hacia la construcción de programas de alto contenido de tokens, líneas de código y uso correcto de estructuras lógicas. Adicionalmente, los hallazgos indican que los estudiantes con buenos desempeños perciben la plataforma como un espacio de trabajo, el cual permite la práctica constante de ejercicios debido a su disponibilidad en línea y facilidad de uso, promoviendo la cantidad de soluciones enviadas por actividad; adicionalmente, el beneficio de la plataforma para generar calificación inmediata, permite a los alumnos concentrar su tiempo en la escritura y la corrección de los programas diseñados, lo que hace más probable que

logren resolver con éxito los ejercicios del curso. Sumado a esto, los resultados resaltan que el desempeño académico está ligado a la percepción de los estudiantes, es decir, que los alumnos que tienen una experiencia positiva con el uso de la plataforma, son los que obtienen calificaciones favorables. Finalmente, los resultados señalan que la herramienta custom input, es utilizada por estudiantes de buenos desempeños académicos, puesto que tienen las capacidades para diseñar pruebas personalizadas para la verificación de los programas construidos.

En contraste, las variables con correlación negativa con respecto al rendimiento académico, son tres tasas de veredictos (respuesta incorrecta, error de ejecución y límite de resultado excedido), cuatro accesos a herramientas (linter, multiple languages code, python tutor y custom input), dos tasas de uso de herramientas (linter y multiple languages code) y una métrica de software correspondiente al índice de mantenibilidad. Las correlaciones negativas de las tasas de veredictos se ven corroboradas por las respuestas donde se menciona la insuficiencia y falta de claridad en la retroalimentación, inflexibilidad en los casos de prueba y validaciones, detalles menores de sintaxis, fallos en la visualización y el funcionamiento de la retroalimentación, falta de participación docente y aspectos a mejorar en las condiciones iniciales. Las correlaciones negativas de las tasas de uso de herramientas se ven contradichas por las menciones positivas de la utilidad de python tutor, custom input y linter, junto con las menciones de lenguajes de programación como beneficios del uso de la plataforma. Por último, la correlación negativa del índice de mantenibilidad se puede ver reflejado en los datos cualitativos en las referencias de habilidades de programación.

Los hallazgos de las correlaciones negativas indican la necesidad de refuerzo en algunos aspectos de curso como la falta de claridad, inflexibilidad y fallas en la visualización de los veredictos de la plataforma, puesto que estas dificultades obstaculizan el proceso de aprendizaje de los estudiantes, lo que se ve reflejado en desempeños académicos deficientes. Adicionalmente, la contradicción entre las correlaciones negativas del uso de herramientas y las menciones positivas en las encuestas de percepción, indican que a pesar de que los alumnos perciban estas herramientas útiles dentro de la plataforma, aún existen carencias en la aplicabilidad de la información ofrecida por las mismas (por ejemplo Linter, la cual se enfoca en la verificación de buenas prácticas de programación, antes que en la resolución de problemas). Finalmente, los resultados relacionados con el índice de mantenibilidad, probablemente significan que los alumnos con buenos rendimientos académicos son los que logran desarrollar habilidades de programación con el uso de la plataforma, por ende tienen la capacidad de diseñar programas modulares de bajo índice de mantenimiento, lo cual es deseable. Sumado a esto, los resultados cuantitativos evidencian que no existe una correlación significativa entre el uso de la herramienta estadísticas de usuario y el desempeño del estudiante, lo cual concuerda con hallazgos de investigaciones identificadas en la revisión de literatura; estos resultados indican que posiblemente dentro del contexto educativo analizado, el seguimiento de calificaciones no es un comportamiento que tenga un impacto relevante en el rendimiento académico de los estudiantes.

La contribución principal de este trabajo es el planteamiento de una marco metodológico con enfo-

que mixto de analítica del aprendizaje aplicable en cursos de programación de computadores, el cual permite complementar, corroborar o refutar los resultados evidenciados cuantitativamente con datos cualitativos y generar hipótesis sobre posibles causas o explicaciones de los comportamientos de los estudiantes. Sumado a esto, los resultados permiten obtener un mejor entendimiento de los procesos de aprendizaje del curso de programación de computadores de la Universidad Nacional de Colombia. A partir de los resultados, es posible evidenciar que el uso de retroalimentación automática y formativa es beneficioso para algunos estudiantes, los cuales logran desarrollar habilidades de programación y aprendizaje autónomo, pero otro grupo de alumnos requieren aún el acompañamiento docente. Adicionalmente, se evidencia que el uso de la mayoría de las herramientas disponibles en la plataforma no está vinculado con estudiantes de desempeños académicos sobresalientes, posiblemente debido a carencias en la claridad del objetivo de cada herramienta, lo que hace que los alumnos que tienen dudas o dificultades no logren solucionarlas.

La principal limitación en esta investigación es el desequilibrio presente en el conjunto de datos utilizados tanto en la fase cuantitativa como cualitativa. En primer lugar, existe un desbalance con respecto a los desempeños académicos, en específico, la cantidad de estudiantes con calificaciones mayores a la nota mínima aprobatoria del curso es mucho mayor que la cantidad de alumnos que no aprueban la asignatura. Esta diferencia en cantidad de estudiantes por categoría se considera una amenaza a la validez, puesto que puede influir en la magnitud de las correlaciones obtenidas diferenciadas por categorías reprobados y aprobados. Adicionalmente este desequilibrio entre categorías probablemente es una restricción en los resultados del análisis cualitativo, donde no se encontró una diferencia notoria en las menciones diferenciadas por grupo. Sumado a esto, existe un desequilibrio en las características demográficas de la muestra de estudiantes, debido a que menos del 20 % de alumnos son mujeres, lo que genera un sesgo en los hallazgos de la investigación hacia los hombres, dejando de lado dinámicas de aprendizaje propias de las estudiantes mujeres.

Por otra parte, existe una limitación ligada con la alta dispersión de los datos de intentos totales realizados y tiempo promedio entre intentos, debido a que las actividades realizadas en el curso poseen una alta diversidad de tipología. En otras palabras, algunos docentes proponen más talleres prácticos, ejercicios de refuerzo o proyectos que son actividades con plazos de entrega más flexibles, y en contraste, otros profesores se concentran en evaluar los conocimientos de los estudiantes por medio de pruebas cortas y exámenes, las cuales comúnmente tienen un límite de tiempo corto. Esto implica que los comportamientos y estrategias implementadas por los estudiantes durante sus procesos de aprendizaje pueden cambiar significativamente en función del tipo de actividad a la que se ven enfrentados. Sin embargo, en la investigación realizada los procesos de aprendizaje son examinados a nivel general, específicamente en variables como el tiempo promedio entre intentos y el total de envíos realizados, lo que no permite observar los procesos individuales que se pueden dar en cada actividad de manera diferenciada. La última de las limitaciones de este estudio está relacionada con el alcance de las preguntas planteadas en las encuestas de percepción de UNCode; teniendo en cuenta que el diseño de estas encuestas se realizó fuera del contexto de la presente investigación, las preguntas

carecen de especificidad sobre las variables consideradas en este estudio. En este sentido, las variables como los tipos de veredicto obtenidos, las herramientas utilizadas y las métricas de software son mencionadas de manera indirecta en los datos cualitativos, generando una restricción en las relaciones que se pueden establecer entre los hallazgos cuantitativos y cualitativos.

7.2. Trabajos futuros

Los resultados y vínculos hallados en este estudio, junto con los limitantes identificados generan oportunidades para investigaciones a futuro. Las investigaciones a futuro se plantean con el objetivo de fortalecer los hallazgos obtenidos, complementar la metodología propuesta o dar solución a problemáticas encontradas. Entre las líneas de investigación que tienen el potencial de ser realizadas como continuación de este trabajo, se destacan las siguientes:

1. Trabajos que implementen la metodología propuesta de analítica de aprendizaje con enfoque mixto, en una muestra poblacional diferente a la utilizada en este estudio. En otras palabras, realizar investigaciones similares en contextos educativos diferentes al utilizado, los cuales pueden corresponder a grupos de estudiantes de otras sedes de la Universidad Nacional de Colombia y de otras universidades donde se ofrecen asignaturas de programación de computadores.
2. Estudios de analítica de aprendizaje donde se utilicen técnicas para la corrección de conjuntos de datos desbalanceados como algoritmos de refinación (*fine-tuning*), remuestreo, submuestreo o sobremuestreo aleatorio de datos. Estas técnicas se podrían aplicar con el objetivo de corregir la diferencia entre el grupo de estudiantes aprobados y reprobados, comparando los resultados obtenidos con el conjunto de datos original y el modificado.
3. Investigaciones donde se aplique un enfoque metodológico cuantitativo y las interacciones de los estudiantes con la plataforma UNCode sean analizadas teniendo en cuenta las diferencias entre las tipologías de actividades académicas (p.ej. exámenes, pruebas cortas, talleres, proyectos y tareas), es decir, las correlaciones se establezcan con respecto a las calificaciones por actividad, en vez de la nota final del curso.
4. Recolección de datos cualitativos adicionales que complementen los resultados de este trabajo. En específico, incluir preguntas abiertas en la encuesta de percepción específicas sobre el uso de las herramientas custom input, linter, python tutor y multiple languages code, lo que permitiría entender en detalle las interacciones de los estudiantes con las herramientas de la plataforma y verificar los resultados contradictorios encontrados en esta investigación.
5. Los comportamientos y percepciones de los estudiantes, sobre el uso de la plataforma UNCode en el desarrollo del curso, sirven como fuente de información para el diseño de intervenciones

que tengan impacto positivo en el desempeño académico de los estudiantes. En otras palabras, con base en las variables que resultan tener correlación significativa con la calificación final, se pueden tomar decisiones sobre el uso de la plataforma o la metodología utilizada en el curso, intentando influenciar positivamente el comportamiento de los alumnos. Las intervenciones pueden ser evaluadas por medio de un diseño experimental, evidenciando estadísticamente su impacto en el proceso de aprendizaje. Algunas de las intervenciones propuestas son, mejorar la visualización de las descripciones y orientaciones de cada veredicto de error y fomentar el uso de las herramientas de la plataforma, recomendándolas después de obtener veredictos de error.

6. Investigaciones con diseño de tipo experimental, en las cuales se determine el impacto del uso de la plataforma o sus herramientas en el rendimiento académico de los estudiantes. Estos trabajos pueden establecer como grupo experimental, un curso de programación de computadores que use UNCode durante las actividades de clase, el cual debe ser comparado con un grupo de control. En específico, se pueden utilizar pruebas estadísticas para verificar si existen diferencias significativas entre el promedio de las calificaciones finales de las dos muestras de alumnos.
7. El conjunto de datos construido tiene el potencial de ser utilizado como base para el diseño de un modelo predictivo. El modelo se puede construir por medio de técnicas avanzadas de aprendizaje de máquina, con el objetivo de predecir el desempeño académico de los estudiantes utilizando como factores las medidas y métricas cuantitativas del conjunto de datos.

7.3. Publicaciones

En el desarrollo de esta investigación se realizó la siguiente publicación:

(Chaparro *et al.*, 2021)

Chaparro, E., Restrepo-Calle, F. & Ramírez-Echeverry, J. J. (2021). *Learning analytics in computer programming courses* [October 19–21, 2021, Arequipa, Perú]. *Proceedings of the IV Latin American Conference on Learning Analytics*, 78-87. <http://ceur-ws.org/Vol-3059/paper8.pdf>

Bibliografía

- Aljohani, N. R., Daud, A., Abbasi, R. A., Alowibdi, J. S., Basher, M. & Aslam, M. A. (2019). An integrated framework for course adapted student learning analytics dashboard. *Computers in Human Behavior*, 92, 679-690.
- Andergassen, M., Mödritscher, F. & Neumann, G. (2014). Practice and Repetition during Exam Preparation in Blended Learning Courses: Correlations with Learning Results. *Journal of Learning Analytics*, 1(1), 48-74.
- Ang, K. L.-M., Ge, F. L. & Seng, K. P. (2020). Big Educational Data amp; Analytics: Survey, Architecture and Challenges. *IEEE Access*, 8, 116392-116414. <https://doi.org/10.1109/ACCESS.2020.2994561>
- Argyris, C. & Schon, D. A. (1974). *Theory in practice: Increasing professional effectiveness*. Jossey-Bass.
- Arnold, K. E. & Pistilli, M. D. (2012). Course Signals at Purdue: Using Learning Analytics to Increase Student Success. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, 267-270.
- Azcona, D., Hsiao, I.-H. & Smeaton, A. F. (2019). Detecting students-at-risk in computer programming classes with learning analytics from students' digital footprints. *User Modeling and User-Adapted Interaction*, 29, 759-788.
- Baker, R. S. & Inventado, P. S. (2014). Educational Data Mining and Learning Analytics. *Learning Analytics: From Research to Practice* (pp. 61-75). Springer New York.
- Barber, R. & Sharkey, M. (2012). Course Correction: Using Analytics to Predict Course Success. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, 259-262. <https://doi.org/10.1145/2330601.2330664>
- Berelson, B. (1952). *Content analysis in communication research*. Glencoe (Ill.) : Free Press.
- Berland, M., Davis, D. & Smith, C. P. (2015). AMOEBA: Designing for collaboration in computer science classrooms through live learning analytics. *International Journal of Computer-Supported Collaborative Learning*, 10, 425-4447.
- Blikstein, P. (2011). Using Learning Analytics to Assess Students' Behavior in Open-Ended Programming Tasks. *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 110-116.
- Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S. & Koller, D. (2014). Programming Pluralism: Using Learning Analytics to Detect Patterns in the Learning of Computer Programming. *Journal of the Learning Sciences*, 23(4), 561-599.
- Bryman, A. (2015). *Mixed methods research: combining quantitative and qualitative research*. Social Research Methods. Oxford University Press.

- Cao, L. (2017). Data Science: A Comprehensive Overview. *ACM Comput. Surv.*, 50(3).
- Carter, A., Hundhausen, C. & Olivares, D. (2019). *Leveraging the Integrated Development Environment for Learning Analytics*. University Press.
- Chaparro, E., Restrepo-Calle, F. & Ramírez-Echeverry, J. J. (2021). Learning analytics in computer programming courses [October 19–21, 2021, Arequipa, Perú]. *Proceedings of the IV Latin American Conference on Learning Analytics*, 78-87. <http://ceur-ws.org/Vol-3059/paper8.pdf>
- Charlton, P., Mavrikis, M. & Katsifli, D. (2013). The Potential of Learning Analytics and Big Data. *Ariadne*. <http://www.ariadne.ac.uk/issue/71/charlton-et-al/>
- Clow, D. (2012). The Learning Analytics Cycle: Closing the loop effectively. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge - LAK '12*, 134-138.
- Coffrin, C., Corrin, L., de Barba, P. & Kennedy, G. (2014). Visualizing Patterns of Student Engagement and Performance in MOOCs. *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, 83-92.
- Corbin, J. M. & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, 13(1), 3-21. <https://doi.org/10.1007/BF00988593>
- Creswell, J. W. (2014). Chapter 1: The Selection of a Research Approach. *Research Design Qualitative, Quantitative, And Mixed Methods Approaches* (4th). SAGE Publications.
- Dascalu, M.-I., Bodea, C.-N., Mogos, R. I., Purnus, A. & Tesila, B. (2018). A Survey on Social Learning Analytics: Applications, Challenges and Importance. En G. C. Silaghi, R. A. Buchmann y C. Boja (Eds.), *Informatics in Economy* (pp. 70-83). Springer International Publishing.
- Dhakal, K. (2022). NVivo. *Journal of the Medical Library Association : JMLA*, 110(2), 270-272.
- Dietze, S., Siemens, G., Taibi, D. & Drachsler, H. (2016). Editorial: Datasets for Learning Analytics. *Journal of Learning Analytics*, 3(2), 307-311.
- Drachsler, H. & Greller, W. (2016). Privacy and Analytics: It's a DELICATE Issue a Checklist for Trusted Learning Analytics. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, 89-98.
- Elia, G., Solazzo, G., Lorenzo, G. & Passiante, G. (2019). Assessing learners' satisfaction in collaborative online courses through a big data approach. *Computers in Human Behavior*, 92, 589-599.
- Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, 4(5/6), 304-317.
- Gasevic, D., Mirriahi, N., Long, P. & Dawson, S. (2014). Editorial: Inaugural Issue of the Journal of Learning Analytics. *Journal of Learning Analytics*, 1(1).
- Gašević, D., Mirriahi, N. & Dawson, S. (2014). Analytics of the Effects of Video Use and Instruction to Support Reflective Learning. *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, 123-132.
- Gergen, K. J., Josselson, R. & Freeman, M. (2015). The promises of qualitative inquiry. *American Psychologist*, 70(1), 1-9.
- Guo, P. J. (2013). Online Python Tutor: Embeddable Web-Based Program Visualization for Cs Education. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 579-584. <https://doi.org/10.1145/2445196.2445368>

- Hernández-Sampieri, R., Fernández-Collado, C. & Baptista-Lucio, P. (2014). Proceso de la investigación cualitativa. *Metodología de la Investigación*. McGraw-Hill.
- Hilliger, I., Ortiz-Rojas, M., Pesántez-Cabrera, P., Scheihing, E., Tsai, Y.-S., Muñoz-Merino, P. J., Broos, T., Whitelock-Wainwright, A. & Pérez-Sanagustín, M. (2020). Identifying needs for learning analytics adoption in Latin American universities: A mixed-methods approach. *The Internet and Higher Education*, 45, 100726.
- Hsiao, I.-H. & Lin, Y.-L. (2017). Enriching programming content semantics: An evaluation of visual analytics approach. *Computers in Human Behavior*, 72, 771-782.
- Hu, Q. & Huang, Y. (2018). A Framework for Analysis Learning Pattern Toward Online Forum in Programming Course. En L. Deng, W. W. K. Ma y C. W. R. Fong (Eds.), *New Media for Educational Change* (pp. 71-80). Springer Singapore.
- Ifenthaler, D. (2017). Are Higher Education Institutions Prepared for Learning Analytics? *TechTrends*, 61(4), 366-371.
- Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., Rubio, M. Á., Sheard, J., Skupas, B., Spacco, J., Szabo, C. & Toll, D. (2015). Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies. *ITICSE-WGR '15: Proceedings of the 2015 ITICSE on Working Group Reports.*, 41-63.
- Kizilcec, R. F., Pérez-Sanagustín, M. & Maldonado, J. J. (2017). Self-regulated learning strategies predict learner behavior and goal attainment in Massive Open Online Courses. *Computers and Education*, 104, 18-33.
- Klašnja-Milićević, A., Vesin, B., Ivanović, M. & Budimac, Z. (2011). E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56(3), 885-899.
- Kolb, D. A. (1984). *Experiential learning: experience as the source of learning and development*. Prentice Hall.
- Kop, R., Fournier, H. & Durand, G. (2017). A Critical Perspective on Learning Analytics and Educational Data Mining. En C. Lang, G. Siemens, A. F. Wise y D. Gašević (Eds.), *The Handbook of Learning Analytics* (1.^a ed., pp. 319-326). Society for Learning Analytics Research (SoLAR).
- Kumar, V. S., Kinshuk, Somasundaram, T. S., Boulanger, D., Seanosky, J. & Vilela, M. F. (2015). Big Data Learning Analytics: A New Perspective. *Ubiquitous Learning Environments and Technologies* (pp. 139-158). Springer Berlin Heidelberg.
- Kurilovas, E. (2019). Advanced machine learning approaches to personalise learning: learning analytics and decision making. *Behaviour and Information Technology*, 38(4), 410-421.
- Lagus, J., Longi, K., Klami, A. & Hellas, A. (2018). Transfer-Learning Methods in Programming Course Outcome Prediction. *ACM Trans. Comput. Educ.*, 18(4).
- Laurillard, D. (2002). *Rethinking University Teaching: A conversational framework for the effective use of learning technologies*. Routledge.
- Leony, D., Muñoz-Merino, P. J., Pardo, A. & Delgado Kloos, C. (2013). Provision of awareness of learners' emotions through visualizations in a computer interaction-based environment. *Expert Systems with Applications*, 40(13), 5093-5100.

- Lockyer, L. & Dawson, S. (2011). Learning Designs and Learning Analytics. *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, 153-156.
- Long, P. & Siemens, G. (2011). Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review*, 46(5), 31-40.
- Lonn, S., Aguilar, S. J. & Teasley, S. D. (2015). Investigating student motivation in the context of a learning analytics intervention during a summer bridge program. *Computers in Human Behavior*, 47, 90-97.
- Lu, O. H. T., Huang, J. C. H., Huang, A. Y. Q. & Yang, S. J. H. (2017). Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course. *Interactive Learning Environments*, 25(2), 220-234.
- Macfadyen, L. P. & Dawson, S. (2010). Mining LMS data to develop an “early warning system” for educators: A proof of concept. *Computers and Education*, 54(2), 588-599.
- Mangaroska, K. & Giannakos, M. (2017). Learning Analytics for Learning Design: Towards Evidence-Driven Decisions to Enhance Learning. En É. Lavoué, H. Drachsler, K. Verbert, J. Broisin y M. Pérez-Sanagustín (Eds.), *Data Driven Approaches in Digital Education* (pp. 428-433). Springer International Publishing.
- Martin, F. & Whitmer, J. C. (2016). Applying Learning Analytics to Investigate Timed Release in Online Learning. *Technology, Knowledge and Learning*, 21, 59-74.
- Monllaó Olivé, D., Huynh, D. Q., Reynolds, M., Dougiamas, M. & Wiese, D. (2020). A supervised learning framework: using assessment to identify students at risk of dropping out of a MOOC. *Journal of Computing in Higher Education*, 32, 9-26.
- Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R. & Muharemagc, E. (2016). Deep Learning Techniques in Big Data Analytics. *Big Data Technologies and Applications* (pp. 133-156). Springer International Publishing. https://doi.org/10.1007/978-3-319-44550-2_5
- Pardo, A. & Siemens, G. (2014). Ethical and privacy principles for learning analytics. *British Journal of Educational Technology*, 45(3), 438-450.
- Pistilli, M. D., Willis, J. E. & Campbell, J. P. (2014). Analytics Through an Institutional Lens: Definition, Theory, Design, and Impact. *Learning Analytics: From Research to Practice* (pp. 79-102). Springer New York.
- Ramírez-Echeverry, J. J., Restrepo-Calle, F. & González, F. A. (2022). A case study in technology-enhanced learning in an introductory computer programming course. *Global Journal of Engineering Education*, 24(1).
- Restrepo-Calle, F., Ramírez-Echeverry, J. & Gonzalez, F. (2018). UNCODE: INTERACTIVE SYSTEM FOR LEARNING AND AUTOMATIC EVALUATION OF COMPUTER PROGRAMMING SKILLS. *EDULEARN18 Proceedings*, 6888-6898. <https://doi.org/10.21125/edulearn.2018.1632>
- 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. & González, F. A. (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*, 22(3), 174-185.
- Rienties, B. & Toetenel, L. (2016). The Impact of 151 Learning Designs on Student Satisfaction and Performance: Social Learning (Analytics) Matters. *Proceedings of the Sixth International Conference on Learning Analytics and Knowledge*, 339-343.
- Robinson, C., Yeomans, M., Reich, J., Hulleman, C. & Gehlbach, H. (2016). Forecasting Student Achievement in MOOCs with Natural Language Processing. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, 383-387.
- Ruipérez-Valiente, J. A., Muñoz-Merino, P. J., Leony, D. & Kloos, C. D. (2015). ALAS-KA: A learning analytics extension for better understanding the learning process in the Khan Academy platform. *Computers in Human Behavior*, 47, 139-148.
- Scheffel, M., Niemann, K., Leony, D., Pardo, A., Schmitz, H.-C., Wolpers, M. & Delgado Kloos, C. (2012). Key Action Extraction for Learning Analytics. En A. Ravenscroft, S. Lindstaedt, C. D. Kloos y D. Hernández-Leo (Eds.), *21st Century Learning for 21st Century Skills* (pp. 320-333). Springer Berlin Heidelberg.
- Schmitz, M., van Limbeek, E., Greller, W., Sloep, P. & Drachsler, H. (2017). Opportunities and Challenges in Using Learning Analytics in Learning Design. En É. Lavoué, H. Drachsler, K. Verbert, J. Broisin y M. Pérez-Sanagustín (Eds.), *Data Driven Approaches in Digital Education* (pp. 209-223). Springer International Publishing.
- Schön, D. A. (1983). *The Reflective Practitioner: How professionals think in action*. Temple Smith.
- Schön, D. A. (1991). *The Reflective Turn: Case studies in and on educational practice*. Teachers College Press.
- Selwyn, N. (2020). Re-imagining 'Learning Analytics' ... a case for starting again? *The Internet and Higher Education*, 46, 100745.
- Seufert, S. & Meier, C. (2018). Big Data in Education: Supporting Learners in Their Role as Reflective Practitioners. *Frontiers of Cyberlearning: Emerging Technologies for Teaching and Learning* (pp. 103-123). Springer Singapore.
- Shen, H., Liang, L., Law, N., Hemberg, E. & O'Reilly, U.-M. (2020). Understanding Learner Behavior Through Learning Design Informed Learning Analytics. *Proceedings of the Seventh ACM Conference on Learning @ Scale*, 135-145.
- Siemens, G. (2013). Learning Analytics: The Emergence of a Discipline. *American Behavioral Scientist*, 57(10), 1380-1400.
- Society for Learning Analytics Research, S. (2022). *What is Learning Analytics?* <https://www.solaresearch.org/about/what-is-learning-analytic>
- Stemler, S. (2000). An overview of content analysis. *Practical assessment, research, and evaluation*, 7(1), 17.
- Tabuenca, B., Kalz, M., Drachsler, H. & Specht, M. (2015). Time will tell: The role of mobile learning analytics in self-regulated learning. *Computers and Education*, 89, 53-74.

- Tam, V., Lam, E. Y., Huang, Y., Liu, K., Tam, V. & Tse, P. (2016). Developing the Petal E-Learning Platform for Facial Analytics and Personalized Learning. *Learning, Design, and Technology: An International Compendium of Theory, Research, Practice, and Policy* (pp. 1-17). Springer International Publishing.
- Tempelaar, D. T., Rienties, B. & Giesbers, B. (2016). Verifying the Stability and Sensitivity of Learning Analytics Based Prediction Models: An Extended Case Study. En S. Zvacek, M. T. Restivo, J. Uhomobhi y M. Helfert (Eds.), *Computer Supported Education* (pp. 256-273). Springer International Publishing.
- Trætteberg, H., Mavroudi, A., Sharma, K. & Giannakos, M. (2017). Utilizing Real-Time Descriptive Learning Analytics to Enhance Learning Programming. *Learning, Design, and Technology: An International Compendium of Theory, Research, Practice, and Policy* (pp. 1-22). Springer International Publishing.
- Vahdat, M., Oneto, L., Anguita, D., Funk, M. & Rauterberg, M. (2015). A Learning Analytics Approach to Correlate the Academic Achievements of Students with Interaction Data from an Educational Simulator. En G. Conole, T. Klobučar, C. Rensing, J. Konert y E. Lavoué (Eds.), *Design for Teaching and Learning in a Networked World* (pp. 352-366). Springer International Publishing.
- Wong, B. T.-m. & Li, K. C. (2020). A review of learning analytics intervention in higher education (2011–2018). *Journal of Computers in Education*, 7, 7-28.
- Wu, Y. & Wu, W. (2018). A Learning Analytics System for Cognition Analysis in Online Learning Community. En L. H. U y H. Xie (Eds.), *Web and Big Data* (pp. 243-258). Springer International Publishing.
- Zacharis, N. Z. (2015). A multivariate approach to predicting student outcomes in web-enabled blended learning courses. *The Internet and Higher Education*, 27, 44-53.
- Zikopoulos, P., Eaton, C. & IBM. (2011). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data* (1st). McGraw-Hill Osborne Media.