

**RETOÑOSAPP: UNA PLATAFORMA DE ENSEÑANZA VIRTUAL PARA
APOYAR LA ENSEÑANZA DE LA PROGRAMACIÓN EN EL AULA**

**DAVID STEVEN SALAMANCA SÁNCHEZ
JUAN SEBASTIÁN RODRÍGUEZ OBREGÓN
DAVID MAURICIO VALOYES PORRAS**



**UNIVERSIDAD EL BOSQUE
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
Bogotá, 2022**

**RETOÑOSAPP: UNA PLATAFORMA DE ENSEÑANZA VIRTUAL PARA
APOYAR LA ENSEÑANZA DE LA PROGRAMACIÓN EN EL AULA**

**DAVID STEVEN SALAMANCA SÁNCHEZ
JUAN SEBASTIÁN RODRÍGUEZ OBREGÓN
DAVID MAURICIO VALOYES PORRAS**

**Proyecto de Grado presentado como requisito para optar por el título de
INGENIERO DE SISTEMAS**

Director
MIGUEL ALFONSO FEIJÓO GARCÍA
Ing. Sistemas y Computación
Ing. Ambiental
Mag. Ingeniería de la Información
Mag. Docencia en la Educación Superior

**UNIVERSIDAD EL BOSQUE
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
Bogotá, 2022**

Este trabajo está dedicado a nuestras familias, quienes trabajaron para podernos brindar una educación profesional y de calidad, y que son quienes jamás han dejado de creer en nosotros y apoyarnos.

AGRADECIMIENTOS

A nuestro director de proyecto de grado, **Miguel Alfonso Feijóo García**, quien desde el primer momento nos recibió con los brazos abiertos acogiéndonos como sus tesis. Miguel se esforzó en compartir su tiempo, conocimiento y experiencia en cada etapa del proyecto, y quien se esmeró en ser nuestra guía y compañía durante todo el proceso que supone la realización de este trabajo, con sus virtudes, su paciencia, su constancia y su integridad no solo ayudó, sino que posibilitó, y con gran calidad, la consecución de lo que hoy podemos llamar RetoñosApp.

A los **profesores del programa de Ingeniería de Sistemas de la Universidad El Bosque**, por haber sido nuestros mentores y maestros durante toda nuestra formación profesional, brindar su tiempo, conocimiento y carisma profesional para que no solamente “viéramos una materia más”, sino para que fuéramos los mejores profesionales que podíamos ser. Por brindarnos consejos de vida, que, desde su propia experiencia, tanto profesional como personal, nos otorgaban para que no solo fuéramos excelentes ingenieros, sino también, y especialmente, buenas personas. Y en general, por haber estado presentes en prácticamente cada etapa del camino que ahora nos permite recibirnos como Ingenieros de Sistemas.

A la profesora **Carolina Rojas Celis**, quien, sin ser profesora del programa de Ingeniería de Sistemas en particular, ni de la Facultad de Ingeniería en general, nos apoyó en el planteamiento de investigación que luego se tornó en los instrumentos de recolección de datos que nos permitieron validar nuestra solución y por ende cumplir con el propósito del proyecto.

A nuestros **compañeros de carrera**, por haber recorrido este camino junto a nosotros. Por habernos apoyado cuando lo necesitábamos, brindarnos una amistad y un corazón sincero que hizo este viaje más sencillo. Por compartir sus ideas, perspectivas, opiniones y conocimientos propios, cada vez que fue posible, para sacar adelante cada pequeño trabajo o proyecto, incluyendo este. Por estar días, tardes y noches trabajando lado a lado.

Y finalmente, pero, ante todo, agradecemos a nuestros **padres y hermanos**, quienes, a pesar de la distancia, las dificultades, el esfuerzo, los días y las noches, el tiempo, el dinero o el cansancio, siempre estuvieron a nuestro lado como nuestros guías, compañeros, amigos, y motores para que pudiéramos cumplir nuestros sueños. Quienes con sus palabras de apoyo, aliento y motivación no permitieron que en ningún momento nos diéramos por vencidos. Quienes han dejado su corazón, sudor y sangre para podernos brindar una educación profesional y de calidad. Y, en el caso de nuestros padres, quienes no solo nos dieron la vida, sino que han dado la suya para que pudiéramos vivirla.

CONTENIDO

1. PROBLEMA	3
1.1 DESCRIPCIÓN DEL CONTEXTO A INTERVENIR	3
1.2 ANÁLISIS DEL CONTEXTO DESDE EL MODELO BIOPSIOSOCIAL Y CULTURAL (MBPSC).....	4
1.2.1 Análisis del contexto desde le MBPSC:	5
1.2.1.1 Hábitos:	5
1.2.1.2 Creencias:.....	5
1.2.1.3 Medios:	6
1.2.1.4 Artefactos:.....	6
1.2.2 Análisis del problema desde el MBPSC:	7
1.3 IDENTIFICACIÓN Y DESCRIPCIÓN DE LA PROBLEMÁTICA	8
2. SOLUCION DE INGENIERÍA	10
2.1 OBJETIVOS: GENERAL Y ESPECÍFICOS.	10
2.2 DESCRIPCIÓN DE LA SOLUCIÓN Y RESULTADOS ESPERADOS	11
2.3 ANÁLISIS DE LA SOLUCIÓN DESDE EL MBPSC	13
2.3.1 Análisis de la solución desde el MBPSC:.....	13
2.4 HIPÓTESIS DE INVESTIGACIÓN.....	14
2.5 TABLA DE ENTREGABLES	14
2.6 VARIABLES PARA MEDIR.....	18
2.7 METODOLOGÍA	19
2.7.1 Manifiesto ágil	20
2.7.2 SCRUM	21
2.7.2.1 Roles de SCRUM	21
2.7.2.2 Fases de SCRUM.....	21
2.7.2.3 Procesos de SCRUM.....	22
2.7.2.4 Artefactos y entregables de SCRUM	23
2.7.3 Lean	24
2.7.3.1 Roles de Lean.....	24
2.7.3.2 Fases de Lean.....	25
2.7.3.3 Principios de Lean	26
2.7.3.4 Procesos de Lean.....	27
2.7.3.5 Artefactos y entregables de Lean	27

2.7.4	Metodología final propuesta: Metodología híbrida entre SCRUM y Lean	27
2.7.4.1	Roles	27
2.7.4.2	Fases	28
2.7.4.3	Principios	30
2.7.4.4	Artefactos y entregables	30
2.7.4.5	Modelo y ciclo de vida	31
2.7.4.6	Estructura de desglose de trabajo (EDT/WBS).....	32
2.7.4.7	Aplicación de la metodología	32
2.7.4.8	Estudio sobre la población objetivo	34
2.7.4.9	Validación con la población objetivo	34
2.7.4.10	Gestión del trabajo en equipo	35
2.7.4.11	Tablero de gestión de tareas	36
2.7.4.12	Reuniones.....	36
2.7.4.13	Cronograma de actividades	36
2.8	COMPONENTE ÉTICO	37
3.	MARCO REFERENCIAL	39
3.1	ANTECEDENTES Y ESTADO DEL ARTE	39
3.2	MARCO TEÓRICO	43
3.2.1	La enseñanza y el aprendizaje de ingeniería y la programación.....	43
3.2.2	Estrategias de apoyo a la enseñanza-aprendizaje en ciencias de la computación	45
3.2.2.1	Constructivismo en la educación	45
3.2.2.2	Enseñanza y aprendizaje asistido por computadora	46
3.2.2.3	Utilización del e-learning como metodología para la enseñanza y el aprendizaje en ciencias de la computación.....	47
3.2.2.4	Bots conversacionales como tutores virtuales	47
3.2.2.5	Gamificación como metodología de enseñanza de la programación	48
3.2.2.6	Sistemas de recomendación como medio para ajustar las metodologías de enseñanza en la educación	49
3.2.3	Satisfacción: Que es y cómo se mide	50
3.2.4	Experiencia del usuario (UX).....	51
4.	DESARROLLO METODOLÓGICO	52

4.1	ESPECIFICACIÓN DE SPRINTS	52
4.2	ESTUDIO SOBRE LA POBLACIÓN OBJETIVO	56
4.2.1	Primera iteración del estudio sobre la población objetivo.....	57
4.2.2	Segunda iteración del estudio sobre la población objetivo	58
4.3	ESPECIFICACIÓN DE REQUERIMIENTOS	58
4.3.1	Requerimientos funcionales	59
4.3.2	Requerimientos no - funcionales.....	62
4.4	TRABAJO COLABORATIVO	64
4.5	GESTIÓN DEL TRABAJO EN EQUIPO	65
4.6	DESCRIPCIÓN DE LOS COMPONENTES UTILIZADOS DE CADA METODOLOGÍA QUE CONSTITUYE LA METODOLOGÍA HÍBRIDA DEL PROYECTO	66
4.7	APLICACIÓN DE SCRUM.....	66
4.8	APLICACIÓN DE LEAN.....	66
5.	RESULTADOS	68
5.1	DISTRIBUCIÓN DE PORCENTAJES DE COMPLETITUD PARA EL DESARROLLO DEL PROYECTO	68
5.2	RESULTADOS ESPERADOS	69
5.3	RESULTADOS PARCIALES (PROYECTO DE GRADO 1).....	73
5.3.1	Resultados parciales del proyecto correspondientes al periodo académico 2021-02, obtenidos en función de los objetivos planteados	73
5.4	RESULTADOS OBTENIDOS	75
5.4.1	Primera validación frente a la población objetivo	75
5.4.2	Segunda validación frente a la población objetivo (resultados finales) 76	
5.4.2.1	Cuestionario.....	76
5.4.2.2	Resultados obtenidos en la segunda validación frente a la población objetivo	79
5.5	VALIDACIÓN EN LA ACADEMIA Y VALIDACIÓN ESTÁTICA.....	83
6.	ANÁLISIS DE RESULTADOS / DISCUSIÓN	84
6.1	ANÁLISIS DE RESULTADOS CORRESPONDIENTES A LA PRIMERA ITERACIÓN DE VALIDACIÓN	84
6.2	ANÁLISIS DE RESULTADOS CORRESPONDIENTES A LA SEGUNDA ITERACIÓN DE VALIDACIÓN (RESULTADOS FINALES).....	85
6.2.1	Análisis de resultados	86

7.	CONCLUSIONES.....	90
7.1	CONCLUSIONES GENERALES	90
7.2	CONCLUSIONES CON BASE EN LOS OBJETIVOS	91
7.2.1	Conclusiones con base en el objetivo general	91
7.2.2	Conclusiones con base en los objetivos específicos.....	92
7.2.2.1	Conclusión frente al objetivo específico 1.....	92
7.2.2.2	Conclusión frente al objetivo específico 2.....	93
7.2.2.3	Conclusión frente al objetivo específico 3.....	94
7.2.2.4	Conclusión frente al objetivo específico 4.....	95
7.3	CONCLUSIONES DESDE LA PERSPECTIVA DEL MBPSC	95
7.4	CONSIDERACIONES FINALES	96
8.	RECOMENDACIONES Y TRABAJO FUTURO.....	97
9.	LECCIONES APRENDIDAS	99
10.	REFERENCIAS BIBLIOGRÁFICAS.....	100
11.	ANEXOS.....	107
11.1	ANEXO 1.....	107
11.2	ANEXO 2.....	107
11.3	ANEXO 3.....	107
11.4	ANEXO 4.....	107
11.5	ANEXO 5.....	107
11.6	ANEXO 6.....	107
11.7	ANEXO 7.....	107
11.8	ANEXO 8.....	107
11.9	ANEXO 9.....	108
11.10	ANEXO 10.....	108
11.11	ANEXO 11.....	108
11.12	ANEXO 12.....	108
11.13	ANEXO 13.....	108

LISTA DE TABLAS

Tabla 1. Entregables del proyecto RetoñosApp con su código y descripción correspondiente.	15
Tabla 2. Especificación Sprint 1 (Crear).	53
Tabla 3. Especificación Sprint 2 (Crear).	53
Tabla 4. Especificación Sprint 3 (Crear).	54
Tabla 5. Especificación Sprint 4 (Medir).	54
Tabla 6. Especificación Sprint 5 (Aprender).	54
Tabla 7. Especificación Sprint 6 (Híbrido).	55
Tabla 8. Especificación Sprint (Híbrido).	55
Tabla 9. Especificación Sprint 9 (Híbrido).	56
Tabla 10. Estructura del formato utilizado para la especificación de requerimientos funcionales.	59
Tabla 11. Indexación de los requerimientos funcionales.	61
Tabla 12. Estructura del formato utilizado para la especificación de requerimientos no - funcionales.	62
Tabla 13. Indexación de los requerimientos no - funcionales.	63
Tabla 14. Distribución de porcentajes de completitud para el desarrollo del proyecto	68
Tabla 15. Categorías representativas de los resultados de la etapa de validación dinámica.	71
Tabla 16. Resultados esperados en torno a las variables de medición de la etapa de validación dinámica.	71
Tabla 17. Distribución de las preguntas con relación a las subvariables de medición.	77
Tabla 18. Preguntas del cuestionario.	77
Tabla 19. Caracterización demográfica general.	79
Tabla 20. Caracterización demográfica de los estudiantes del curso de Fundamentos de Programación.	79
Tabla 21. Caracterización demográfica de los estudiantes del curso de Programación 1.	80
Tabla 22. Resultados finales discriminando los grupos de Fundamentos de Programación y Programación 1.	81
Tabla 23. Resultados finales globales.	81
Tabla 24. Promedio de las respuestas del cuestionario por grupo.	81
Tabla 25. Cálculo de los resultados.	82
Tabla 26. Categorización de las calificaciones obtenidas en cada pregunta del cuestionario.	88

LISTA DE FIGURAS

Ilustración 1. Jerarquía de las variables de medición.	19
Ilustración 2. Diagrama de las fases de la metodología SCRUM.....	22
Ilustración 3. Diagrama de las fases de la metodología Lean Startup.	26
Ilustración 4. Diagrama general de las fases de la metodología híbrida (Lean-Agile).	29
Ilustración 5. Diagrama de aplicación de las fases de la metodología híbrida (Lean-Agile).....	29
Ilustración 6. Diagrama del modelo en espiral de la metodología híbrida (Lean-Agile).....	31
Ilustración 7. Espacios del tablero Kanban para el product backlog del proyecto en función de las fases de Lean.	65
Ilustración 8. Espacios del tablero Kanban destinados a las actividades iterativas, en progreso y finalizadas en función de las fases de Lean.....	65

GLOSARIO

- **Software amigable:** Capacidad que tiene un producto de software para ser visualmente atractivo, y de fácil comprensión, aprendizaje y uso para el usuario. Referencia: [1], [2].

RESUMEN

RETOÑOSAPP: UNA PLATAFORMA DE ENSEÑANZA VIRTUAL PARA APOYAR LA ENSEÑANZA DE LA PROGRAMACIÓN EN EL AULA

Salamanca Sánchez David Steven ¹, Rodríguez Obregón Juan Sebastián ²,
Valoyes Porras David Mauricio ³

Resumen

Las dificultades de aprendizaje frente a las ciencias de la computación son una problemática de carácter multicausal que afecta no solo el desempeño académico de los estudiantes, sino que también tiene implicaciones psicológicas, emocionales y motivacionales sobre estos, en el corto y largo plazo. RetoñosApp nace como una plataforma web que apoya la enseñanza y el aprendizaje impartido en el aula de clase, cuyo principal fin, más allá de ser una herramienta y un medio de estudio y trabajo autónomo, es el poder otorgar un acompañamiento y una realimentación a los estudiantes durante su ejercicio académico, y una realimentación a los docentes sobre los procesos de aprendizaje de sus estudiantes. Para el desarrollo del proyecto se diseñó, implementó y desplegó un aplicativo web, que fue validado frente a los estudiantes de los cursos de Fundamentos de Programación y Programación 1 del programa de Ingeniería de Sistemas de la Universidad El Bosque, con el fin de validar su eficacia y eficiencia, desde la perspectiva de la experiencia de usuario. Como resultados tenemos que la población objetivo valoró como sobresaliente su experiencia de usuario frente al aplicativo, por medio de la usabilidad y la satisfacción de usuario como variables implícitas de la experiencia de usuario misma. Finalmente, como conclusión general, se establece que tanto el proyecto planteado, como especialmente el artefacto desarrollado promueven positivamente los ejercicios de aprendizaje y enseñanza en el aula de clase, puesto que efectivamente provee una asistencia asincrónica a los ejercicios de aprendizaje, al igual que una realimentación frente a los procesos de enseñanza, al igual que en general la población objetivo identificó como agradable su experiencia de usuario (en función de múltiples aspectos).

Palabra clave: Ciencias de la computación, enseñanza y aprendizaje, bot conversacional, sistema de recomendación basado en contenido, identificación de las dificultades de aprendizaje, metodologías educativas blended, aplicativo web.

Abstract

Learning difficulties in computer science are a multi-causal problem that affects not only the academic performance of students, but also has psychological, emotional, and motivational implications on them, in the short and long term. RetoñosApp was born as a web platform that supports teaching and learning in the classroom, whose main purpose, beyond being a tool and a means of study and independent work, is to provide support and feedback to students during their academic exercise, and feedback to teachers about the learning processes of their students. For the development of the project, a web application was designed, implemented, and deployed, which was validated with the students of the Fundamentals of Programming and Programming 1 courses of the Systems Engineering program at Universidad El Bosque, to validate its effectiveness and efficiency, from the perspective of user experience. The results show that the target population rated their user experience with the application as outstanding, by means of usability and user satisfaction as implicit variables of the user experience itself. Finally, as a general conclusion, it is established that both the proposed project, and especially the developed artifact, positively promote learning and teaching exercises in the classroom, since it effectively provides asynchronous assistance to learning exercises, as well as feedback to the teaching processes, and in general the target population identified their user experience as pleasant (based on multiple aspects).

Keywords: Computer science, teaching and learning, conversational bot, content-based recommender system, identification of learning difficulties, blended learning methodologies, web application.

INTRODUCCIÓN

Las dificultades de aprendizaje son un tópico de estudio común en campos como la psicología, la psiquiatría, la pedagogía, la sociología e incluso la filosofía [3]–[5]. Pero, desde la perspectiva de la ingeniería, las dificultades de aprendizaje comúnmente son abordadas desde un enfoque en el que estas deben solucionarse por medio de artefactos más allá de solamente estudiarse académicamente. Incluso, en muchos casos, la ingeniería no se enfoca en indagar sobre cuales o como son las dificultades de aprendizaje, o porque se dan, sino que le delegan esa labor a otros campos del conocimiento, con los cuales trabajan de la mano para desarrollar los artefactos que las solucionen, eviten o reduzcan [4], [5].

Sin embargo, considerando que la ingeniería es una rama del conocimiento humano cuyo principal enfoque es la transformación de la realidad a través de la aplicación de conocimientos de diferentes campos [6], sin tener prácticamente ninguna limitación en torno a los campos del conocimiento a utilizar [6], no es descabellado pensar que la ingeniería es igualmente capaz de identificar las dificultades de aprendizaje, su origen y proveer herramientas para manejarlas, reducirlas, evitarlas o corregirlas a través de los mismos artefactos que esta desarrolla.

El presente proyecto propone, diseña, implementa y valida una plataforma web interactiva que apoye la enseñanza y el aprendizaje, de la programación fundamental, impartido en el aula de clase. Para ello, tiene como punto focal la utilización de mecanismos interactivos y de inteligencia artificial para realizar las labores de enseñanza/tutoría de los estudiantes, pero con el objetivo de poder identificar las dificultades de aprendizaje de las temáticas y los conceptos de estos en los cursos de Fundamentos de Programación y Programación 1 del programa de Ingeniería de Sistemas de la Universidad El Bosque. Lo anterior tiene como finalidad proveer una herramienta de estudio que los estudiantes puedan utilizar para complementar su proceso de aprendizaje, y de la cual los docentes puedan obtener una realimentación en torno a los procesos de aprendizaje de sus estudiantes con la cual puedan sustentar ajustes, cambios o correcciones a las metodologías o procesos de enseñanza de sus cursos.

Teniendo como principales diferenciadores tres elementos particularmente: (1) la interactividad como elemento que gamifica la experiencia de aprendizaje propiciando la motivación, la retención de información, la comprensión de las temática y conceptos, y la individualización de las formas de aprendizaje, (2) el uso de bots conversacionales para la realización de labores de tutoría virtual en tiempo real, y (3) el uso de sistemas de recomendación para proveer realimentaciones en torno a los procesos de aprendizaje que consecuentemente permitan adaptar o conciliar los procesos de enseñanza en función de las necesidades individuales y grupales de los estudiantes.

El presente informe comprende la realización del proyecto desde los enfoques ingenieriles y académicos que lo componen. Partiendo de la identificación del contexto, la problemática, la población objetivo y la potencial solución, pasando por la construcción del marco referencial y el planteamiento metodológico, y llegando al diseño, construcción, despliegue, validación del artefacto y posterior análisis de los resultados obtenidos del proceso de validación dinámica.

1. PROBLEMA

En el presente capítulo se realizará un planteamiento inicial y un análisis de la problemática identificada sobre la cual se plantea la elaboración de la innovación tecnológica, estando dicha innovación diseñada y destinada como solución a esta. Lo anterior implica que la problemática identificada plantea a su vez una serie de necesidades, oportunidades y/o requerimientos de una población objetivo, al igual que responde y ocurre en un contexto específico, el cual determina y delimita la solución en términos conceptuales, metodológicos, de procesos de negocio, tecnológicos y, para el caso particular de este proyecto, también en términos académicos.

Al terminar este capítulo, el lector va a tener un entendimiento tanto general como específico de la problemática a solucionar, y del contexto en el cual esta se presenta. Lo anterior, parte en la sección 1.1 con la descripción del contexto a intervenir, como marco inicial y principal a considerar, que permite definir al proyecto desde la perspectiva del entorno sobre el cual este se va a desplegar. Luego, en la sección 1.2 se realiza un análisis textual sobre el modelo biopsicosocial (MBPSC), el cual se puede entender como una descripción de carácter sistémico y desglosada del mismo contexto descrito en la sección 1.1 y posteriormente el análisis de este para el planteamiento de la problemática también desde la perspectiva del MBPSC. Finalmente, en la sección 1.3 se describe la problemática puntual a solucionar, respondiendo a las preguntas de: ¿Dónde parte el problema?, ¿cuál es el problema?, ¿por qué es un problema? y ¿por qué debería solucionarse el problema?, donde aquellas preguntas son a su vez una explicación y una justificación del proyecto desde el punto de vista de la problemática.

1.1 DESCRIPCIÓN DEL CONTEXTO A INTERVENIR

El proyecto se plantea teniendo como población objetivo a los estudiantes y docentes de los cursos de Fundamentos de Programación y Programación 1 del programa de Ingeniería de Sistemas de la Universidad El Bosque. Los anteriores cursos corresponden, desde la perspectiva del plan de estudios del programa, a la etapa de inmersión a la vida universitaria, la cual consiste en los 3 primeros semestres académicos y que según el Proyecto Educativo del Programa (PEP) [7] es la etapa en la cual el estudiante adquiere las bases para poderse enfrentar a los demás cursos planteados en el plan de estudios, siendo dichas bases igualmente, los fundamentos para un correcto desempeño durante su vida profesional. Pero a su vez, y sin restarle importancia a los aspectos académicos y profesionales, la forma en la que se desarrollen y se perciban los primeros semestres académicos incide especialmente en los índices de éxito o deserción estudiantil [8], [9]. Igualmente, cabe aclarar que el proyecto se destina a todos los estudiantes de las dos asignaturas mencionadas, lo que implica que el contexto, en términos de

población objetivo, está compuesto tanto de los estudiantes de la jornada diurna como de la nocturna.

El contexto para intervenir presenta de forma general, en lo referente a los estudiantes, un solo grupo poblacional compuesto por aquellos estudiantes que se encuentran iniciando su plan de estudios en el programa de Ingeniería de Sistemas, y que específicamente están cursando las asignaturas de Fundamentos de Programación y Programación 1 pero de forma particular, este grupo poblacional se puede dividir en 2 subgrupos claramente diferenciados: los estudiantes pertenecientes a la jornada diurna y los pertenecientes a la jornada nocturna.

- En lo referente a los estudiantes de la jornada diurna, la población esta principalmente compuesta por jóvenes con 20 años o menos (aproximadamente el 75%), especialmente durante su primer semestre académico (aquel en el cual según el pensum se estaría cursando Fundamentos de Programación). Lo anterior a su vez está respaldado en que mayormente son estudiantes recién graduados del colegio, que se han tomado desde un semestre hasta un año sabático o que ya cursaron algún programa educativo formal o semiformal previo a la búsqueda de un título de pregrado universitario. Pero a su vez tienen la característica de que, casi en su totalidad, aún no están realizando alguna actividad laboral.
- En lo referente a los estudiantes de la jornada nocturna, la población está compuesta principalmente por estudiantes de 20 años o más. Lo anterior se ve reflejado en que el conocimiento, la experiencia, las necesidades, las dificultades horarias, personales, familiares, laborales, entre otras, la disponibilidad de tiempo, la independencia e incluso las motivaciones de ambos subgrupos poblacionales son considerablemente diferentes entre sí. Debido a lo expuesto, el proyecto debe no solo responder y desempeñarse de forma satisfactoria frente a toda la población objetivo (indistintamente de su jornada), sino que debe estar desarrollado de una forma íntegra para esta, puesto que por la promesa de valor del proyecto no se puede plantear de forma diferencial para cada uno de los subgrupos, sino considerarlos a ambos de forma simultánea e imparcial.

Por su parte, los docentes de las dos asignatura mencionadas anteriormente, constituyen un subgrupo poblacional especial, al estar sus necesidades, frente a la problemática y frente al contexto, ligadas a los estudiantes propiamente.

1.2 ANÁLISIS DEL CONTEXTO DESDE EL MODELO BIOPSIOSOCIAL Y CULTURAL (MBPSC)

En la presente sección se realiza la descripción del MBPSC identificado, con base en el análisis del contexto a intervenir y de la problemática identificada.

Particularmente esta sección está dividida en dos partes: primero la descripción del MBPSC diferenciando (separando) sus distintos elementos (hábitos, creencias, medios y artefactos), para luego realizar un análisis específico del MBPSC frente a la problemática identificada. Lo anterior quiere decir que esta sección busca establecer una línea lógica en la conformación del MBPSC, planteando primero las bases del modelo (las partes) para luego realizar el análisis del MBPSC frente a la problemática como un todo. Esto significa que se busca realizar una aproximación sistémica del MBPSC al plantearse primero las partes que lo componen de forma individual y luego plantear el problema como un todo, estableciendo la relación y clasificación de dichas partes para conformar un análisis de mayor escala.

1.2.1 Análisis del contexto desde le MBPSC:

En la presente sección se realiza la descripción de los elementos (las partes) que componen el MBPSC, de forma independiente. Esto para poder describir de forma más detalla y sustentada cada elemento, para luego poder realizar los análisis del MBPSC como un todo de forma más profunda, precisa y contextualizada. Cabe aclarar que estos elementos también son el punto de partida para realizar el análisis del MBPSC frente a la solución [Sección 2.3], pero que por efectos de no redundancia y de practicidad no se repetirán en dicha sección.

1.2.1.1 Hábitos:

Los estudiantes de programación en general, en cuanto a su aprendizaje independiente, tienen una tendencia a estudiar por medio de material digitales (i.e, foros, bibliografía o tutoriales), buscando aproximar este material al contexto de sus cursos y temáticas particulares [10], [11]. Hoy en día entre otras cosas, predomina el uso de tutoriales y explicaciones en vídeo (comúnmente en Youtube), el uso de sitios y foros en los cuales buscar (o realizar) preguntas y respuestas sobre conceptos o dudas específicas sobre programación (de los cuales se pueden nombrar, entre otros, a Stack Overflow o Quora), o simplemente utilizar referencias tales como documentos digitales (en PDF y presentaciones) o la documentación propia de las tecnologías y lenguajes a utilizar.

1.2.1.2 Creencias:

Aprender a programar, requiere principalmente, de entre otras cosas, tener la habilidad de entender y representar conceptos tangibles e intangibles por medio de modelos, poder traducir ideas a “código”, ser capaz de abstraer y entender las cosas como un todo, y como sus partes. Además, también depende de utilizar de forma simultánea el pensamiento lógico y creativo por la sinergia que estos pueden generar más allá de las diferencias evidentes de estas dos perspectivas [12], [13]. Lo anterior genera la idea de que aprender a programar es complicado, que las

personas tienden a volverse mecánicas o exclusivamente analíticas, que la programación es exclusivamente un proceso técnico, entre otros.

Esas creencias implican que la carrera de Ingeniería de Sistemas en particular en Colombia, tengan un nivel bajo de ingreso de estudiantes frente a otras carreras de ingeniería como por ejemplo la Ingeniería Industrial o Ingeniería Ambiental, pero aún más que el número de graduandos sea considerablemente bajo. Lo anterior no solamente se resume en que los índices de ingreso de nuevos alumnos a programas de Ingeniería de Sistemas sea bajo, sino que los índices de deserción también tienden a ser importantes [8], [9]. Según un artículo publicado por el periódico El Tiempo a finales del año 2020 [14] en Colombia existe un déficit de 80000 ingenieros informáticos (y afines), lo que representa y justifica la importancia de formar estudiantes con conocimientos de dicho campo de forma satisfactoria. Lo anterior, traduciéndose en conocimientos adecuados, bien construidos, íntegros pero que a su vez terminen su formación profesional.

1.2.1.3 Medios:

En el campo de la enseñanza y el aprendizaje de la programación, predominan las herramientas digitales, principalmente por la misma coherencia que tiene el utilizar medios tecnológicos para enseñar tecnología [15]. De la misma forma el uso de herramienta tecnológicas conlleva que el aprendizaje no depende solo de lo que enseñan los docentes en sus cursos, también depende del esfuerzo individual de los estudiantes de apropiarse de ellos (i.e, autonomía), principalmente por las implicaciones en cuanto a la disciplina, interés, motivación y responsabilidad que posee el hecho de enfrentarse a un conjunto de herramientas, y sin supervisión, mas no a personas (como lo son los docentes). Por otro lado, se pueden nombrar medios más tradicionales de enseñanza y aprendizaje como por ejemplo las lecturas, los talleres escritos, las presentaciones o el estudio de casos [16], los cuales parten de la educación misma que recibieron los docentes, pero que a su vez busca mantener técnicas educativas que ya han demostrado su eficacia.

Por su parte, en términos metodológicos, se parte desde una perspectiva constructivista del aprendizaje [17], en donde el estudiante no solamente es quien recibe los conocimientos y se apropia de ellos, sino que es partícipe de su proceso de aprendizaje siendo un actor activo en la edificación de su propio conocimiento (i.e, protagonista de su propio aprendizaje). A su vez, y considerando la perspectiva de la Universidad El Bosque, el modelo del aprendizaje significativo [7] constituye un modelo constructivista de la enseñanza, en donde, el PEP [7] describe que el estudiante no es solamente un receptor, sino que también es un creador y emisor del conocimiento para diversos contextos.

1.2.1.4 Artefactos:

Estando íntimamente relacionado con los medios, los artefactos de enseñanza y aprendizaje de la programación se ven categorizados en dos grandes grupos: (1) las herramientas y (2) las metodologías. Las herramientas, sean o no digitales, son aquellas en las cuales los estudiantes tienen un grado de control, al prácticamente poder determinar si utilizarlas o no, a menos que una situación en específico dictamine lo contrario. Por su parte, en lo referente a las metodologías, los estudiantes pierden un poco el grado de control, al estar estas más relacionadas con los docentes y su ejercicio académico. Así visto, desde el contexto de la Universidad El Bosque, el aprendizaje se ve estructurado, en cuanto a los artefactos, en torno al modelo del aprendizaje significativo, y el modelo biopsicosocial y cultural [7] que rige y guía el quehacer académico general de la universidad, al igual que por las herramientas tradicionales de enseñanza y por medios digitales tales como el aula virtual.

1.2.2 Análisis del problema desde el MBPSC:

En el anexo 1 [Sección 11.1], se presenta el diagrama del MBPSC desde la perspectiva del problema. Posteriormente, se realiza la descripción textual del MBPSC del problema, la cual es una descripción/explicación del diagrama presentado, pero a su vez es un análisis más detallado del mismo teniendo también como insumo los elementos del independientes del MBPSC.

La programación, como conocimiento básico y fundamental de la Ingeniería de Sistemas, presenta uno de los principales retos a los cuales el profesional de este campo debe enfrentarse, desde sus primeros cursos como estudiante, hasta en su vida profesional. Lo anterior, dado que es habitual tener que manejar diferentes tecnologías y herramientas (i.e., herramientas distribuidas entre lenguajes de programación, frameworks, servicios OnCloud, etc). Además, es inherente la necesidad de estar periódicamente aprendiendo y actualizando los conocimientos, al ser un campo del saber que está en constante cambio y acelerada evolución. Por lo anterior, el estudio individual se ve traducido en un hábito que el ingeniero de sistemas desarrolla al verse envuelto en un medio tan cambiante y extremadamente diverso.

En el caso del estudiante esto no es muy diferente, dado que, al entrar como novicios a la carrera, las creencias que tienen sobre lo que es la programación son muy diferentes a lo que van descubriendo a medida que avanzan en su carrera profesional. En primer lugar, dado que descubren que la programación no es un fin en la profesión, sino un medio y una herramienta para ejercer profesionalmente. En segundo lugar, porque ven que la programación es mucho más que solo aprender a escribir código, viéndose representada en nuevas formas de pensar, de expresar, de comunicar, etc. Y en tercer lugar porque la forma de estudiar en programación no consiste en solo estudiar lo técnico, sino en llevar de la mano a la teoría y a la práctica como factores hermanos. Lo anterior, en conjunto con la complejidad intrínseca de la programación y con una serie de causas extra, como pueden ser

falencias previas en otros conocimientos o aptitudes (i.e, conocimientos en matemáticas, análisis lógico y habilidades de lectoescritura), al igual que factores emocionales, se ve traducido para algunos estudiantes, en múltiples dificultades al momento de aprender a programar [8]. Además, lo anterior puede verse expresado en una serie de problemas emocionales y académicos en el corto y largo plazo [18], para el estudiante, la asignatura, el docente e incluso la universidad (i.e., programa académico).

Para sopesar todo lo anterior y enfrentar las dificultades que se le puedan presentar, el estudiante puede hacer uso de una serie de herramientas y fuentes de información que tiene a su disposición, que van desde la tradicional bibliografía, pasando por videotutoriales, llegando a plataformas de enseñanza más estructuradas. Sin embargo, el problema frente a lo anterior surge cuando todas estas herramientas llegan a ser en algunos casos insuficientes, lo que genera una deficiencia de apoyo al estudio. A su vez, en el caso de las herramientas y medios tecnológicos, tales como: videos, plataformas, aulas virtuales, otra problemática que surge es que llega a ser información parcializada, descontextualizada o que representa una enseñanza impersonal y que incluso puede ir en contra de las metodologías (e incluso temáticas) enseñadas en el aula (i.e, integración y apropiación de MVC como estilo arquitectural base para la enseñanza de la programación orientada a objetos en la Universidad El Bosque).

1.3 IDENTIFICACIÓN Y DESCRIPCIÓN DE LA PROBLEMÁTICA

Las dificultades que se presentan durante la conceptualización, entendimiento y aprendizaje de los principios y fundamentos de la programación y de la programación orientada a objetos, le supone al estudiante de Ingeniería de Sistemas, una dificultad posterior para poder enfrentarse, entender y por ende aprobar (i.e., de forma satisfactoria o en su totalidad) el curso en cuestión y los cursos posteriores del programa académico. De igual forma, estas dificultades desnivelan a los grupos de una o más asignaturas, desmotivan al estudiante, le impiden al docente dictar de forma eficiente la asignatura [19]–[22] e incluso reducen el nivel de la materia (entendiéndose nivel como el grado de calidad y dificultad del curso) y del programa académico, por el hecho de que una falencia en un conocimiento tan fundamental para la profesión implica una reacción en cadena a largo plazo.

El cambio de ambiente y de dinámica entre el colegio y la universidad, la metodología de enseñanza del docente, la dinámica interna del grupo de estudiantes, las bases previas de lógica, matemáticas e inglés, la capacidad de abstracción, e incluso la novedad que representa el uso de un lenguaje técnico, son solo algunas de las principales causas que dificultan o incluso imposibilitan el aprendizaje de la programación [22], [23] para el estudiante de Ingeniería de Sistemas durante su etapa de aprendizaje de los fundamentos de la carrera.

Además de lo anterior, esto le impide lograr al estudiante resultados excelentes o sobresalientes para aprobar las asignaturas en cuestión, y a su vez poder enfrentarse a todo el desarrollo curricular posterior de forma satisfactoria y sana [8], [9].

La necesidad y la importancia que representa el adquirir buenas bases en cuanto a los fundamentos y principios de la programación se ven traducidos, desde la perspectiva del estudiante, en la sensación de seguridad en cuanto a la carrera escogida, su desarrollo emocional y curricular, la consecución de sus logros académicos y posteriores metas profesionales [18], [19], [24]. Además, desde la perspectiva de la universidad, se ven representados en un aumento de la calidad del programa académico (i.e., tanto en los exámenes de educación superior, como en el ejercicio profesional de los egresados), la adquisición de mayor nivel de los cursos del programa y de una mejor imagen de la universidad en cuanto a los procesos de enseñanza y aprendizaje de la Ingeniería de Sistemas en la Universidad El Bosque.

En resumidas cuentas, el problema a solucionar son las dificultades multicausales de aprendizaje de los principios y fundamentos de programación como parte de la Etapa de Inmersión a la Vida Universitaria de la carrera de Ingeniería de Sistemas de la Universidad El Bosque, dado que estas dificultades implican un desnivel interno entre los grupos y en el largo plazo para la consecución de los objetivos de aprendizaje y la adquisición de conocimientos tanto básicos como avanzados de la profesión. Además, debe lograrse solucionar esta problemática identificada debido a que estas dificultades y sus consecuencias tienen implicaciones emocionales, motivacionales, personales, académicas, profesionales [18], al igual que afectan el rendimiento académico en el corto y el largo plazo de un estudiante, su grupo y en general afectan la calidad del programa académico respectivo.

Lo anterior, puede evidenciarse, de forma extendida, en el árbol de problemas ubicado en el anexo 2 [Sección 11.2].

2. SOLUCION DE INGENIERÍA

En el presente capítulo se realizará el planteamiento y descripción de la innovación propuesta, como solución de ingeniería, a la problemática planteada en el capítulo 1.

Al terminar este capítulo, el lector va a tener un entendimiento tanto general como específico de la solución propuesta desde los aspectos propios de esta, como actividad académica, como proyecto, como tecnología, como resultado de un proceso de ingeniería y propiamente como solución a una problemática identificada en un contexto específico. Lo anterior, parte en la sección 2.1 con el planteamiento del objetivo general del proyecto y sus correspondientes objetivos específicos, donde estos funcionan como línea guía, meta y definición de los alcances y límites de este. Luego, en la sección 2.2 se realiza la descripción puntual de cuál es la solución planteada vista desde la perspectiva del artefacto (aplicativo), al igual que la especificación de esta como tecnología. Posteriormente, en la sección 2.3 se enlazará y relacionará la solución a la problemática por medio del MBPSC, al presentarse la contraparte/complemento gráfico y textual del planteado en el capítulo 1, siendo igualmente una descripción de carácter sistémico y desglosada de esta. Después, en la sección 2.4 se resumirán, por medio de una tabla, los entregables del proyecto, tanto tecnológicos, documentales y académicos, siendo el primero el producto per se del proyecto, y los dos últimos el soporte, justificación y sustento de este. Posteriormente, en la sección 2.5 se describirá la variable a medir, que será el sustento cuantitativo del desempeño del artefacto frente a la población objetivo y por ende el soporte de la satisfactoria finalización del proyecto y su consecuente despliegue e implementación. Luego, en la sección 2.6 se describirá la metodología planteada para el desarrollo del proyecto estando esta enmarcada de forma simultánea en la Ingeniería de Software y la Gestión de Proyectos, pero considerando los aspectos netamente académicos que se deben tener en cuenta puesto que el proyecto constituye la realización de un proyecto de grado de la Universidad El Bosque. Finalmente, en la sección 2.7 se hará el planteamiento, análisis y la reflexión sobre el marco ético que regirá el desarrollo del proyecto, estando este alineado con el “Código de Ética y Conducta Profesional” [25] de la *Association for Computing Machinery (ACM)*, específicamente enmarcado en el “Código de Ética de la Ingeniería de Software” [26] desarrollado en conjunto por la ACM y la *Institute of Electrical and Electronics Engineers (IEEE)*, en particular por el *IEEE Computer Society (IEEE – CS)*. A su vez, y considerando el contexto particular colombiano, este marco ético será respaldado por el “Código de Ética para el ejercicio de la Ingeniería en general y sus profesiones afines y auxiliares” [27] del Consejo Profesional Nacional de Ingeniería (COPNIA).

2.1 OBJETIVOS: GENERAL Y ESPECÍFICOS.

OBJETIVO GENERAL:

Desarrollar una herramienta de enseñanza en línea, mediada por inteligencia artificial y un sistema de recomendación, que apoye, complemente y permita realimentar, por medio de reportes, la enseñanza y el aprendizaje de los cursos de Fundamentos de Programación y Programación 1 del programa de Ingeniería de Sistemas de la Universidad El Bosque.

OBJETIVOS ESPECÍFICOS:

1. Analizar las características del contexto de enseñanza y aprendizaje de las asignaturas Fundamentos de Programación y Programación 1, del programa de Ingeniería de Sistemas de la Universidad El Bosque, con el fin de determinar las necesidades, requerimientos y oportunidades específicas que presenta la población objetivo por medio de un estudio realizado a los estudiantes de los cursos de Fundamentos de Programación y Programación 1 en los periodos 2021-01, 2021-02 y 2022-01 y una entrevista con los docentes de estos cursos (tanto de los horarios diurnos como nocturnos).
2. Definir el diseño y arquitectura del sistema de información, de tal forma que estos permitan que el aplicativo sea amigable con el usuario tanto estéticamente como en cuanto a la experiencia del usuario (UX) y cuyo diseño sea coherente, en términos de interfaz (UI), con el objetivo general de la plataforma y las funciones que le provee al usuario.
3. Construir un aplicativo web que integre un bot conversacional que haga la tarea de tutor virtual, y un sistema de recomendación alimentado por información sobre los procesos de aprendizaje de los estudiantes generada de manera automatizada, con el fin de apoyar los procesos de enseñanza y aprendizaje de los estudiantes y los docentes dentro y fuera del aula de clase.
4. Realizar pruebas de verificación y validación, y con la población objetivo en términos de experiencia del usuario (UX), con el fin de demostrar la utilidad de la plataforma, su nivel de calidad y el grado de satisfacción de los estudiantes y docentes frente al aplicativo.

2.2 DESCRIPCIÓN DE LA SOLUCIÓN Y RESULTADOS ESPERADOS

La solución propuesta consiste en una plataforma de enseñanza en línea que busca apoyar los procesos de enseñanza y aprendizaje que se dan en el aula de clase en los cursos de Fundamentos de Programación y Programación 1 del programa de

Ingeniería de Sistemas de la Universidad El Bosque. Para lograr lo anterior, la plataforma tendrá tres características principales:

1. Contará con un espacio de enseñanza interactivo destinado a la explicación de las temáticas y conceptos propios de los cursos de Fundamentos de Programación y Programación 1. Se puede entender como el “tablero” interactivo sobre el cual se impartirán las temáticas y conceptos.
2. Implementará un bot conversacional que hará la tarea de un tutor virtual, con el cual el estudiante podrá interactuar por medio de preguntas sobre las temáticas y conceptos impartidos en la plataforma y a través de este podrá recibir respuestas a sus dudas e inquietudes, profundizar en los tópicos y en general una realimentación de su proceso de aprendizaje. Básicamente consistirá en un sistema sincrónico que actúa como un “docente” digital para acompañar, guiar e instruir al estudiante durante el proceso de aprendizaje en la plataforma. A su vez, este bot conversacional almacenará dichas preguntas (y cualquier otra entrada que el estudiante le otorgue al bot) para luego ser utilizadas en un sistema de recomendación.
3. Implementará en el fondo, un sistema de recomendación alimentado con la información (entradas) recolectada por el tutor virtual (y la plataforma en general) para realizar un proceso de analítica sobre los datos y así generar un reporte sobre dificultades de enseñanza identificadas, temáticas o conceptos de especial complejidad (o facilidad) para los estudiantes, preguntas más recurrentes, entre otros. Dicho reporte se les otorgará a los docentes de los cursos de Fundamentos de Programación y Programación 1 con el fin de poderlo utilizar para ajustar las metodologías de enseñanza o reforzar las temáticas y conceptos en los cuales se identifique que los estudiantes presentan dificultades.

Con lo anterior, se hará uso de herramientas, técnicas, conceptos y metodologías de la Ingeniería de Software, la Ingeniería y Analítica de Datos, y los Sistemas Inteligentes.

En cuanto a la estructura general, estará conformado por 3 módulos principales:

1. Administrador: Consiste en el módulo destinado a la gerencia administrativa de la plataforma, donde se manejan aspectos tales como: configuración de parámetros, gestión de usuarios, gestión de incidencias, verificación de la trazabilidad, visualización (o exportación) de reportes, verificación de funcionalidad de la aplicación, entre otros. Fundamentalmente se puede comprender como el “Panel de Control” de los administradores frente a la aplicación, lo que les permite gestionarla y manipularla de forma práctica, eficiente y profunda sin la necesidad de intervenir en el código o las bases de datos de forma directa (lo que implica que los administradores no necesariamente tienen que comprender de tecnología).

2. Docente: Consiste en el módulo destinado a toda la administración de la actividad docente por parte de los profesores, la cual incluye, pero no se limita a: creación de asignaturas, creación de temáticas, creación de sesiones de clase, gestión de los estudiantes de las respectivas asignaturas, y la visualización (o exportación) de reportes de sus asignaturas y de sus alumnos (estando esta parte estrictamente relacionada con el sistema de recomendación planteado previamente).
3. Estudiante: Consiste en el módulo mediante el cual los estudiantes realizarán el trabajo asincrónico propuesto como apoyo y complemento al proceso de enseñanza realizado en las sesiones sincrónicas propias del programa de Ingeniería de Sistemas de la Universidad El Bosque. Las acciones posibles en este módulo incluyen, pero no se limitan a: la visualización e interacción con los videos diseñados para cada una de las sesiones de clase, la visualización de las fichas bibliográficas creadas como elemento multimedia para estudio y complemento de dichas sesiones de clase, seguimiento de su progreso, y principalmente la interacción con el bot conversacional que hará la labor de tutor virtual y guía “sincrónico” y “autónomo” que acompañara al estudiante durante su proceso de aprendizaje en la plataforma.

2.3 ANÁLISIS DE LA SOLUCIÓN DESDE EL MBPSC

En la presente sección se realiza la descripción del modelo biopsicosocial y cultural (MBPSC) identificado con respecto a la innovación y producto de ingeniería planteado como solución a la problemática identificada [Sección 1.3].

Al igual que en el caso del análisis del MBPSC desde la perspectiva de la problemática, aquí se hace uso de los elementos individuales que componen al MBPSC (como sus partes) [Sección 1.2.1] para realizar este análisis de mayor escala. En este caso, para el desarrollo del MBPSC con respecto a la solución, aparte de intervenir dichos elementos fundamentales del MBPSC, también se hace uso del mismo análisis realizado, pero desde la perspectiva de la problemática, de la descripción del contexto a intervenir, de los objetivos planteados en el proyecto y de la descripción de la solución planteada.

2.3.1 Análisis de la solución desde el MBPSC:

En el anexo 3 [Sección 11.3], se presenta el diagrama del MBPSC desde la perspectiva de la solución. Posteriormente, se realiza la descripción textual del MBPSC de la solución, la cual es una descripción/explicación del diagrama presentado, pero a su vez es un análisis más detallado del mismo teniendo también como insumo los elementos del independientes del MBPSC.

Una plataforma de enseñanza representa un sistema de apoyo al aprendizaje tradicional impartido en el aula de clase, asistiendo así al docente en su quehacer y

brindando a los estudiantes un medio de estudio dinámico e integral con el cual complementar la enseñanza impartida en el aula. Estando implementada en conjunto con mecanismos de inteligencia artificial, la plataforma le permite al docente tener una asistencia asincrónica para la asignatura, y que desde la perspectiva del estudiante se ve como un medio de aprendizaje sincrónico al tener una guía, una tutoría y un proceso de retroalimentación virtual. Igualmente siendo internamente un sistema de recomendación impulsado por los datos y la información recolectada por parte del tutor virtual, resulta ser un medio para que el docente pueda identificar las principales dificultades de aprendizaje de los estudiantes, sus falencias, el grado de comprensión de las temáticas, etc.

Implementada de forma satisfactoria, la plataforma ayuda a reemplazar una serie de hábitos en cuanto a la búsqueda de información y a la utilización de determinadas herramientas de estudio que incluso pueden llegar a ser perjudiciales para el proceso de enseñanza y aprendizaje del estudiante. Así mismo al ser no solo otra plataforma de enseñanza, sino una asistencia al aprendizaje en aula y un sistema que le brinda al docente información sobre el aprendizaje de los estudiantes, esta tecnología puede beneficiar emocional, personal, académica y profesionalmente al estudiante [18], y a su vez beneficiar a los cursos y la calidad del programa de Ingeniería de Sistemas de la Universidad El Bosque. Lo anterior, puesto que la plataforma busca que el estudiante tenga un mayor grado de apropiación de las temáticas de los cursos de Fundamentos de Programación y de Programación 1 al proveer a este de un medio para complementar y reforzar la enseñanza que se le imparte en el aula de clase, e igualmente porque le ofrece al docente una realimentación del proceso de aprendizaje de sus alumnos. A su vez, la solución se enmarca en el modelo del aprendizaje significativo bajo el cual se diseñan y estructuran las estrategias y objetivos educativos del programa de Ingeniería de Sistemas, tal como lo plantea el PEP [7].

2.4 HIPÓTESIS DE INVESTIGACIÓN

La hipótesis de investigación para el proyecto RetoñosApp es la siguiente:

“El utilizar una plataforma de enseñanza cuya característica principal es que hace la labor de un "tutor virtual", beneficia satisfactoriamente tanto los procesos de enseñanza como de aprendizaje de la programación en el aula de clase, al funcionar fundamentalmente como un complemento a las actividades educativas desarrolladas dentro de esta, pero a su vez proveyendo una realimentación frente a los tópicos, en lo referente a los estudiantes, y de progreso de sus alumnos, en lo que se refiere a los docentes”.

2.5 TABLA DE ENTREGABLES

En la siguiente tabla [Tabla 1] se enlistan y describen los entregables planteados para el proyecto, referenciándolos mediante su nombre general y por medio de un código planteado para el desarrollo del proyecto mismo, el cual permite la trazabilidad, organización y estandarización de estos.

Tabla 1. Entregables del proyecto RetoñosApp con su código y descripción correspondiente.

Código	Nombre	Descripción
DOC001_ActaConstitucion	Acta de constitución	Es el documento que marca el comienzo formal del proyecto a desarrollar, incluyendo, pero no limitándose a la definición de: objetivos, tiempos, recursos (financieros, tecnológicos, teóricos, humanos, etc), involucrados, entregables y metodología.
DOC002_DocumentoSAD	Documento de Arquitectura de Software (SAD)	Consiste en el compendio de todos los diagramas, tablas de especificación, patrones, diseños y justificaciones que describen el sistema como un todo y sus partes, siendo básicamente todos los artefactos de ingeniería, diseño y desarrollo y gestión de software que soportan la construcción del producto de software.
DOC003_PlanCalidad	Plan de calidad	Documento que detalla los procesos y objetivos que garantizan la calidad del productos de software, estableciendo igualmente el o los modelos de calidad que harán la tarea de marcos y normatividades sobre los cuales se va a regir el desarrollo del

Código	Nombre	Descripción
		proyecto y el producto final.
DOC004_PlanPruebas	Plan de pruebas	Conjunto de casos de prueba a los cuales el producto de software debe someterse para validar su calidad y funcionalidad técnica, especificando de estos su definición, justificación, implementación y el procedimiento que debe realizarse en cada uno de estos.
DOC006_ManualUsuario	Manual de usuario	Explica, desde la perspectiva de los diferentes usuarios del aplicativo, el funcionamiento paso a paso y extremadamente detallado de la plataforma, sus diferentes módulos, pantallas y elementos de interacción, diferenciando el funcionamiento de cada módulo según cada uno de los roles que pueda hacer uso de cada uno de estos.
DOC007_ManualTecnico	Manual técnico	El paso a paso del proceso de instalación, implementación y configuración de la plataforma, estableciendo requerimientos mínimos del sistema, condiciones de uso, limitaciones, exigencias del entorno, entre otros, al igual que en este se relaciona el despliegue de la aplicación con su diseño infraestructural.

Código	Nombre	Descripción
DOC012_Bitacora	Bitácora	Resume hitos importantes en la realización del proyecto, como parte del proceso de gestión de este, al igual que como requerimiento de carácter académico de su desarrollo y como elemento que incentiva la toma de decisiones en pro de la ejecución del proyecto.
DOC013_ReflexionesPeriodicas	Reflexiones periódicas	Consiste en el documento en el cual se presenta las reflexiones periódicas realizadas en torno al proyecto, como parte del proceso de gestión de este, al igual que como requerimiento de carácter académico de su desarrollo y como elemento que permite realizar un proceso de realimentación y meditación de carácter personal y grupal.
DOC009_InformeFinal	Informe final	Documento de carácter académico que resume, soporta y justifica el proyecto como un todo y sus partes, haciendo especial énfasis en lo documental y de gestión de proyectos, sin eliminar la importancia de lo técnico. Igualmente, se puede entender como el documento final que soporta todo el proyecto de grado, haciendo la labor de "tesis" pero en el contexto del programa de Ingeniería

Código	Nombre	Descripción
		de Sistemas de la Universidad El Bosque frente a la modalidad de Proyecto de Grado.
No aplica puesto que no es solo un acta, sino un conjunto de actas	Actas de reunión con los stakeholders	Todas las actas de reunión realizadas con los stakeholders del proyecto.
No aplica puesto que no consiste en un documento/archivo propiamente	Repositorio de aplicación de la metodología	Conjunto de elementos que sustentan y evidencia la aplicación de la metodología propuesta, incluyendo, entre otras cosas, las actas de reunión, implementación de la metodología híbrida, bitácora, reflexiones periódicas y tablero de control.
No aplica puesto que no consiste en un documento/archivo propiamente, sino en un conjunto de piezas de código, y demás elementos tecnológicos que trabajan como uno solo	Artefacto de software	Producto de software finalizado, completamente funcional, que cumple a cabalidad con el plan de pruebas y de calidad entregados, que ya fue validado satisfactoriamente con la población objetivo y que responde a todos los elementos solicitados como productos clave en la realización del proyecto de grado.

Fuente: Elaboración propia.

2.6 VARIABLES PARA MEDIR

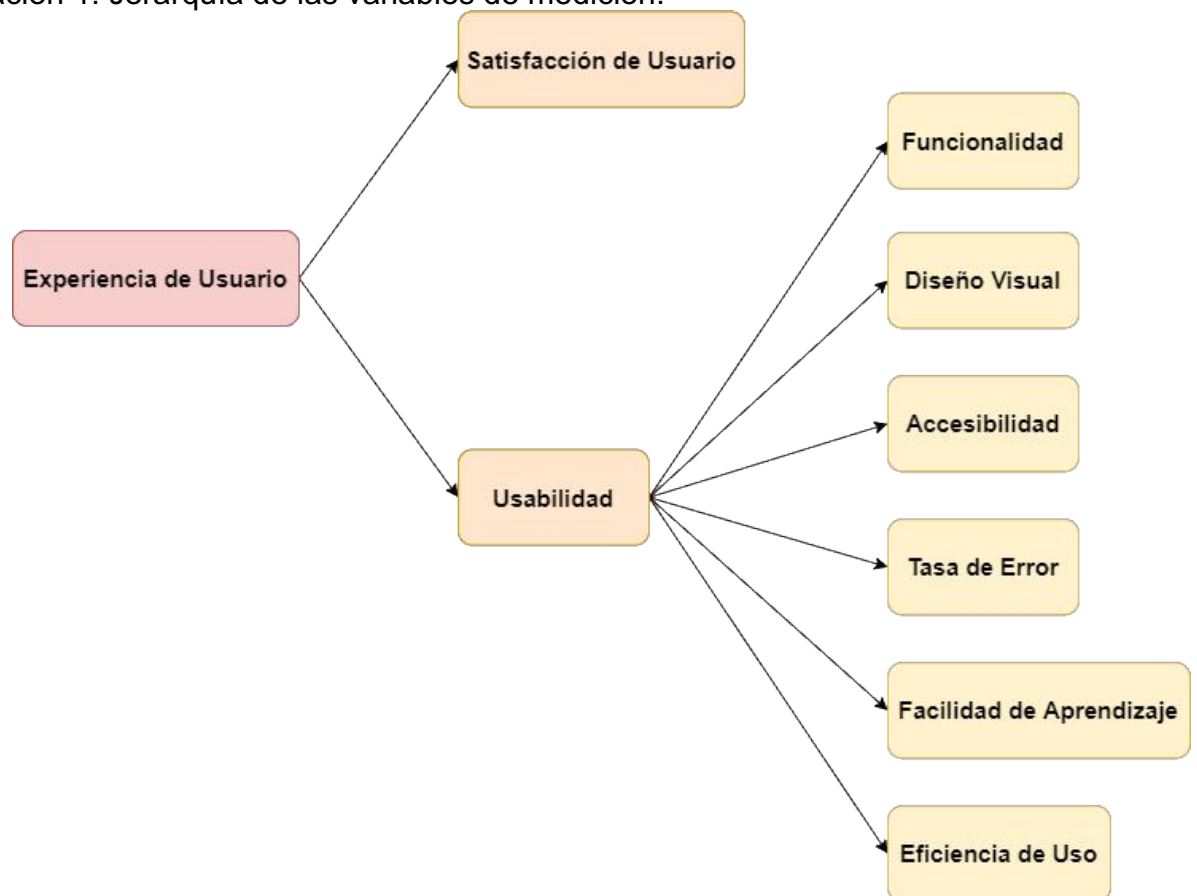
La variable de medición que se utilizará para validar la solución ante la población objetivo, durante la etapa de validación dinámica, consistirá en la experiencia de usuario (UX) [Sección 3.2.4], la cual se medirá utilizando escalas Likert y Net Promoter Score [Sección 3.2.3]. La medición de la experiencia de usuario se logrará a través de la medición del grado de satisfacción [Sección 3.2.3] que tienen tanto los estudiantes, como los docentes que imparten las asignaturas, frente a la

implementación de la herramienta desarrollada en los cursos de Fundamentos de Programación y Programación 1 y a su vez del nivel de usabilidad de la plataforma.

Así visto, y en coherencia con lo que se especifica en el marco teórico [Sección 3.2.4], la variable de medición (experiencia de usuario), será medida implícitamente a través de la satisfacción de usuario y la usabilidad del artefacto, donde a su vez la usabilidad se medirá a través de las siguientes variables implícitas: funcionalidad, diseño visual, accesibilidad, tasa de error, facilidad de aprendizaje y eficiencia de uso.

En la siguiente imagen [Ilustración 1] se presenta un diagrama que expone la jerarquía de las subvariables, en relación con la variable global (experiencia de usuario), que se utilizarán para la medición de esta:

Ilustración 1. Jerarquía de las variables de medición.



Fuente: Elaboración propia.

2.7 METODOLOGÍA

Las metodologías de desarrollo de software nacen como la “imposición” de un proceso disciplinado y organizado para la realización de procesos de construcción profesional de software [28]. En este campo existen dos grandes enfoques o categorías de las metodologías de desarrollo: las metodologías tradicionales y las metodologías ágiles. Las primeras se enfocan en un proceso robusto, muy procedimental, altamente documentado e íntegro de desarrollo, pero poco flexible y que presenta resultados en intervalos de tiempo considerablemente largos [28]; mientras que la segunda se enfoca en la presentación de resultados constantes en intervalos de tiempo mucho más cortos de forma iterativa, la flexibilidad en la aparición de cambios en el contexto, producto o condiciones de proyecto, y permite trabajar múltiples aspectos del proyecto de forma simultánea [28].

Para el presente proyecto, RetoñosApp, se ha optado por el diseño e implementación de una metodología híbrida compuesta por la metodología ágil SCRUM y la metodología Lean en general, tomando factores y elementos de diferentes variantes particulares de esa última. Lo anterior teniendo como resultado lo que se conoce como una metodología “Lean-agile”. Esta decisión se justifica en dos aspectos:

1. Se busca aprovechar las ventajas que ofrecen las metodologías de desarrollo ágiles, entre las que se encuentran, pero no se limitan a la presentación constante de avances y resultados del proyecto de forma iterativa, la flexibilidad de la adaptación de la metodología al proyecto en sí y al contexto en el cual este se desenvuelve, el trabajo en paralelo que estas posibilitan, el enfoque equilibrado entre el producto y las necesidades de los stakeholders, y el valor que le dan al equipo de trabajo [28].
2. Hacer uso de los principios filosóficos y prácticas sobre las cuales está desarrollada la metodología Lean como elemento que agrega valor al cliente y al producto, y que aportan calidad y satisfacción a los stakeholders, al igual que permiten gestionar y optimizar los procesos de desarrollo [29]–[33].

Para lograr lo anterior, en esta sección se define la estructura metodológica propuesta sobre la cual se va a trabajar, estableciendo inicialmente las características individuales de ambas metodologías para posteriormente precisar la metodología híbrida resultante de estas.

2.7.1 Manifiesto ágil

El manifiesto ágil consiste en un conjunto de 12 postulados o principios establecidos por 17 estudiosos de las metodologías de desarrollo de software que se reunieron para debatir sobre las futuras tendencias en la industria del software [28], [34]–[36]. En esa misma reunión se acuñó el término “Metodología Ágil” y se establecieron estos 12 principios como el pilar filosófico de estas metodologías [28], [34]–[36].

2.7.2 SCRUM

SCRUM es una metodología ágil enfocada en la entrega iterativa de prototipos funcionales que evolucionan hasta un MVP (i.e, haciendo una relación con la metodología Lean) y posteriormente a un producto final [37]. Sus ventajas radican en la posibilidad de realizar avances simultáneos en diferentes aspectos y partes de un proyecto, obtener una realimentación constante que permita asegurar la calidad del producto, enfocarse en la productividad y en proveer procesos más eficientes para el desarrollo del producto [37], [38]. Además, también trabaja de la mano con los stakeholders para mantener la integridad del proyecto [37]. Igualmente, dicha metodología trabaja sobre procesos libres y flexibles que se adaptan a los requerimientos y características propias del proyecto, del producto y del contexto [37].

2.7.2.1 Roles de SCRUM

Los roles existentes en SCRUM son los siguientes [39]:

1. *Product Owner*: Su labor es asegurar que el producto genere valor (o que el valor sea aún mayor) para los stakeholders siendo el encargado de tener una visión y entendimiento claro de las necesidades y requerimientos de estos, y del producto. De la misma forma es quien se encarga de desarrollar el plan estratégico o roadmap del proyecto, administra el product backlog y filtra los requerimientos del producto. Básicamente es quien se focaliza en los aspectos de negocio del proyecto y del producto.
2. *SCRUM Master*: Consiste en el líder del equipo de desarrollo en SCRUM, buscando la eficiencia del trabajo en el grupo, actuando como mentor, guía y mediador en las dificultades y problemas que se puedan originar dentro del equipo. Igualmente, en su labor de líder de equipo es quien actúa como intermediario o puente de comunicación entre el equipo de desarrollo y el Product Owner y se encarga de dirigir las reuniones internas del grupo y externas con los stakeholders y con el Product Owner.
3. *Development Team*: Es el grupo de desarrolladores calificados en las diferentes necesidades tecnológicas y metodológicas que se requieren para el desarrollo del proyecto en general y del producto en particular, estando distribuido entre las diferentes especialidades que se requieran. Igualmente, el SCRUM Master hace parte del equipo de desarrollo y el Product Owner puede o no ser parte de este.
4. *Stakeholders*: Son las partes interesadas en el desarrollo del proyecto y en la consecución del producto final, estando compuestos por inversores, clientes, población objetivo, evaluadores, entre otros.

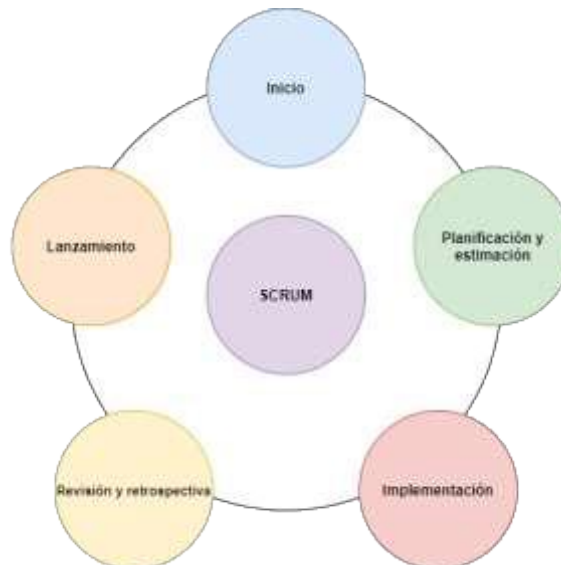
2.7.2.2 Fases de SCRUM

SCRUM se compone de las siguientes 5 fases iterativas [37]:

1. Inicio: Fase en la cual se hace una evaluación sobre el proyecto, se identifica el product backlog, se determinan los roles y las responsabilidades de cada miembro del equipo de desarrollo, se identifican a los stakeholders, se realizan las épicas y se realiza un plan de trabajo general.
2. Planeación y estimación: Se elaboran las historias de usuario (que se aprueban y asignan en la misma fase), se estiman las tareas a realizar, se estiman los requerimientos en términos de recursos y se distribuye el product backlog entre los sprints generando así los sprint backlog y se realiza la planeación sobre los sprints definiendo igualmente el tiempo de estos.
3. Implementación: Desarrollo de los entregables, ejecución de las diferentes tareas identificadas referentes a el desarrollo del producto, realización de las reuniones diarias (daily SCRUM) y mantenimiento del sprint backlog.
4. Revisión y retrospectiva: Fase en la cual se convoca a una o varias reuniones enfocadas a hacer un análisis retrospectivo de lo realizado en la fase de implementación y en el sprint en general, al igual que se validan los entregables realizados hasta el momento y se corrigen o ajustan en caso de ser necesario.
5. Lanzamiento: Fase en la cual se envían los entregables y se realiza un proceso de realimentación sobre el proyecto.

En la siguiente imagen [Ilustración 3] se presenta el diagrama que expone (en orden) las diferentes fases de la metodología SCRUM:

Ilustración 2. Diagrama de las fases de la metodología SCRUM.



Fuente: Elaboración propia.

2.7.2.3 Procesos de SCRUM

Como metodología, SCRUM tiene 5 fases bien definidas, pero el enfoque iterativo de SCRUM radica en que internamente estas fases se desarrollan por medio de lo que se conoce como sprint, los cuales son ciclos de entre 2 a 4 semanas en los cuales se desarrollan las diferentes tareas de las que se compone SCRUM [37]–[39]. Estos sprints consisten en las siguientes partes (sub-fases de SCRUM) [37]:

1. Daily SCRUM: Consiste en una reunión de 15 minutos de duración en donde cada desarrollador responde a las siguientes 3 preguntas: ¿Qué se hizo ayer? ¿Qué se tiene planeado hacer hoy? ¿Qué obstáculos se encontraron en el camino?
2. Sprint planning: El equipo de desarrollo del proyecto se reúne para planificar el sprint, asignar las tareas y definir el tiempo del sprint.
3. Product idea: Se hace una revisión sobre los requerimientos del producto para delimitar bien los objetivos y tareas de desarrollo, al igual que determinar qué evolución o incremento tendrá el prototipo actual.
4. Product backlog: Mapeo sobre las tareas generales del proyecto y se definen los entregables o resultados esperados de cada una.
5. Sprint backlog: Se definen las tareas particulares del sprint a realizar, las cuales provienen de las tareas generales del proyecto.
6. Sprint execution: Cada desarrollador realiza las tareas que le fueron asignadas y se realizan las reuniones espontaneas que surjan por una determinada necesidad o situación.
7. Sprint review: Estando dirigida por el SCRUM Master y por el Product Owner, consiste en la revisión general de las actividades realizadas en el sprint y se realiza la entrega del trabajo terminado.
8. Product increment: Se realiza la entrega a los stakeholders del nuevo incremento (o prototipo) desarrollado durante el sprint.
9. Sprint retrospective: Estando dirigida por el SCRUM Master y por el Product Owner, se realiza un revisión retrospectiva sobre el sprint finalizado, en la cual se ajustan metodologías, responsabilidades, dificultades y demás situaciones o aspectos identificados por los desarrolladores.

Un proyecto puede variar entre la cantidad de sprints requeridos y estos pueden tener una duración diferente entre sí, pero se debe ser claro sobre esto en la fase de planeación y estimación [37].

2.7.2.4 Artefactos y entregables de SCRUM

Los artefactos y entregables obligatorios que define SCRUM son [38]:

- Product backlog.
- Sprint backlog.
- Historias de usuario.
- Tablero de tareas.

- Prototipos incrementales.
- Producto Mínimo Viable (MVP).
- Producto final.

Aparte de los anteriormente mencionados, los equipos de desarrollo pueden agregar cualquier otro artefacto o entregable que consideren pertinente en el proyecto o que pueda ser necesario debido a la dinámica interna particular del equipo.

2.7.3 Lean

Lean consiste en una filosofía (principalmente) y una metodología para la organización del trabajo cuyo objetivo es agregar valor al cliente y al producto, y optimizar los procesos de producción [33], estando enfocada en la eliminación de desperdicios [33]. Lean, al ser fundamentalmente una filosofía destinada a los procesos de producción se ve reflejada en una serie de variantes cuyos enfoques industriales, organizacionales o metodológicos son diferentes, pero esencialmente pueden ser combinadas o utilizadas de forma simultánea [29]–[33].

Para el desarrollo del proyecto RetoñosApp se utilizarán aspectos extraídos de las variantes llamadas Lean Software Development [31], [32] (lo que es coherente con que el proyecto se enmarca en el campo de la Ingeniería de Software), Lean Startup [29], [30] y Lean Six Sigma [33]. Lo anterior considerando la característica previamente explicadas que permite combinar o utilizar de forma simultánea diferentes variantes de ser fundamentalmente iguales con enfoques distintos. En adición a esto, cabe aclarar que los tres enfoques de estas variantes de Lean se enmarcan en tres niveles diferentes de un proyecto de innovación (equipo, producto y empresa) [40], [41], y considerando que la propuesta de Proyecto de Grado de la Universidad El Bosque esencialmente consiste en un proyecto de innovación tecnológica, el utilizar aspectos de estas tres metodologías permite trabajar sobre esos tres niveles de una forma coherente y bien sustentada.

2.7.3.1 Roles de Lean

En general dentro de la metodología Lean se pueden encontrar los siguientes roles [29]–[33]:

1. Lean Champion: Es el líder de equipo, encargado de coordinar el despliegue de la metodología Lean, la distribución de recursos, el mantenimiento del plan de acción, gestionar el cambio, velar por el cumplimiento de los principios filosóficos de Lean y actuar mentor, guía y mediador para los miembros de equipo.
2. Lean Project Leader(s): Son los encargados de dirigir los diferentes proyectos que se estén ejecutando de forma simultánea siguiendo las indicaciones y

directrices del Lean Champion. La cantidad de líderes depende de la cantidad de miembros de equipo y especialmente de la cantidad de proyectos que se estén realizando.

3. Lean Team Members: Es el grupo de desarrolladores calificados en las diferentes necesidades tecnológicas y metodológicas que se requieren para el desarrollo del proyecto en general y del producto en particular, estando distribuido entre las diferentes especialidades que se requieran. Igualmente, el Lean Champion y los Lean Project Leaders hacen parte del equipo de desarrollo.
4. Sponsor: Encargado de seleccionar los proyectos sobre los que se va a trabajar y definir los equipos de trabajo para cada uno de estos, siendo a su vez quien propende por la calidad de los resultados de cada proyecto.

En particular dentro de Lean Six Sigma se pueden encontrar los siguientes roles, definidos en términos de cintas, tomando como inspiración el sistema de grados de las artes marciales [33]:

1. White Belt.
2. Yellow Belt.
3. Green Belt.
4. Black Belt.
5. Master Black Belt.
6. Champi3n.

La raz3n por la cual no se hace mayor descripci3n de estos roles consiste en que dependen de una certificaci3n que solo puede ser otorgada por la Asociaci3n Internacional de Certificaci3n de Six Sigma (IASSC por sus siglas en ingl3s) o las instituciones acreditadas para tal efecto.

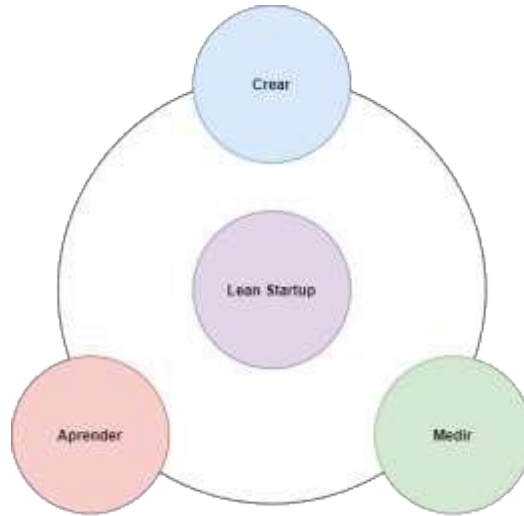
2.7.3.2 Fases de Lean

Extraídas de la metodología Lean Startup [29], [30], las fases (iterativas) de Lean son las siguientes:

1. Crear: Desarrollo del MVP realizando todas aquellas actividades que el equipo de desarrollo haya determinado para tal fin.
2. Medir: Fase en la cual se realizan las pruebas técnicas y humanas del MVP y/o donde este se prueba sobre la poblaci3n objetivo, obteniendo un grado de validaci3n de este.
3. Aprender: Análisis de los resultados obtenidos en la fase de medici3n con el objetivo de realizar correcciones, cambios y ajustes al MVP.

En la siguiente imagen [Ilustraci3n 4] se presenta el diagrama que expone (en orden) las diferentes fases de la metodología Lean:

Ilustración 3. Diagrama de las fases de la metodología Lean Startup.



Fuente: Elaboración propia.

2.7.3.3 Principios de Lean

Como una filosofía, los principales elementos de Lean consisten en principios. En particular, los principios Lean enfocados y adaptados al desarrollo de software son [31], [32]:

1. Eliminar los desperdicios.
2. Amplificar el aprendizaje sobre los errores, aciertos y retos identificados durante el proceso de desarrollo, para así optimizar los procesos y mejorar las prácticas de forma evolutiva y por medio de la realimentación.
3. Decidir lo más tarde posible como medio para que las decisiones se basen en hechos y no en suposiciones o pronósticos basados en la incertidumbre siempre presente.
4. Entregar lo más rápido que sea posible para apostar por una evolución y obtener un proceso de realimentación constantes.
5. Capacitar al equipo, siendo estos el pilar del proyecto.
6. Construir una integridad intrínseca responsabilizando al equipo de la calidad del producto, alcanzando así la madurez del grupo de trabajo y del proyecto.
7. Ver todo el conjunto donde el producto no es la suma de sus partes sino el producto de las interacciones de estas y considerando que al final todo el equipo gana o todo el conjunto pierde.

De forma específica y referente al primer principio, un desperdicio es todo aquello que no añade valor al cliente o al producto [32], para lo cual Lean en general establece 7 tipos de desperdicios [33]:

1. Sobreproducción.

2. Defectos.
3. Transporte excesivo.
4. Movimientos innecesarios.
5. Tiempo de espera.
6. Procedimientos innecesarios.
7. Inventario de sobra.

En particular, estos desperdicios traducidos al desarrollo de software consisten en [31], [32]:

1. Características y funcionalidades extra.
2. Defectos.
3. Trabajo terminado parcialmente.
4. Procesos innecesarios.
5. Tiempo de espera.
6. Cambio de tareas.
7. Burocracia para la toma de decisiones.

2.7.3.4 Procesos de Lean

Lean, al ser principalmente una filosofía enfocada a los procesos de manufactura y el desarrollo de proyecto no tiene propiamente procesos previamente establecidos. Lo anterior, debido a que el principal principio en cualquier variante de Lean consiste en eliminar los desperdicios para optimizar cualquier proceso, y uno de los desperdicios consiste en procesos innecesarios. Esto implica que el hecho de establecer procesos fijos puede derivarse en procesos que no se acoplen bien a un proyecto y que por ende sean innecesarios.

2.7.3.5 Artefactos y entregables de Lean

Por la misma razón por la que Lean no cuenta con procesos previamente establecidos, esta filosofía tampoco cuenta con artefactos o entregables, debido a que estos pueden no acoplarse adecuadamente a un proyecto y por ende verse traducidos en “inventario de sobra”, siendo así un desperdicio.

2.7.4 Metodología final propuesta: Metodología híbrida entre SCRUM y Lean

2.7.4.1 Roles

Para el desarrollo del proyecto RetoñosApp se va a hacer uso de los roles definidos por SCRUM, es decir:

1. Product Owner: Como Product Owner se tiene al director de proyecto de grado al ser este quien tiene pleno conocimiento del proyecto, puesto que ha estado presente, y de forma activa, en todo su planteamiento y desarrollo. A

su vez quien tiene toda la visión del desarrollo de las asignaturas de Proyecto de Grado 1 y 2, y consecuentemente de las particularidades del desarrollo de proyectos de grado en la Universidad el Bosque. Igualmente, porque tiene la experiencia tanto académica como práctica en el desarrollo de proyecto de ingeniería en general y de Ingeniería de Sistemas en particular. Y finalmente, porque tiene a su vez el conocimiento de primera mano del contexto sobre el cual se ha de desempeñar el artefacto y de la población objetivo a la cual está destinado, lo cual propende por el aumento de valor del producto.

2. SCRUM Master: Como SCRUM Master estará uno de los 3 estudiantes del grupo de trabajo, el cual será definido de forma consensuada por todo el equipo de trabajo y por el director de proyecto.
3. Development Team: Como equipo de desarrollo se plantean a los 3 miembros del equipo de trabajo, en donde los 3 tendrán responsabilidades en los tres frentes principales del proyecto: artefacto, documentación y gestión. Aunque se buscará que cada miembro maximice el valor de su participación y de los resultados de su trabajo (tanto documental como en lo concerniente al artefacto) por medio de la explotación de sus habilidades y conocimientos particulares, se busca igualmente que haya un trabajo balanceado, justo y que cada miembro participe de forma activa en cada aspecto del proyecto, siendo así coherentes con el hecho de que el proyecto es a su vez un ejercicio profesional y académico.
4. Stakeholders: Como Stakeholders se entienden, considerando la modalidad bajo la cual se plantea y desarrolla el presente proyecto de grado, a los docentes de las asignaturas Proyecto de Grado 1 y 2, al director de proyecto, a los jurados que revisarán y calificarán el proyecto y la población objetivo del proyecto [Sección 1.1].

2.7.4.2 Fases

La combinación entre SCRUM y Lean, en cuanto a sus fases, se da en el momento de la fase “crear” de Lean, la cual estará compuesta por todas las iteraciones necesarias de SCRUM. Lo anterior debido a que es en esa fase de Lean en la cual se desarrolla el MVP, o en este caso el proyecto, y son las fases de SCRUM las que constituyen el desarrollo de los artefactos y entregables del proyecto.

Lo anterior significa que, por cada iteración de Lean realizada, se llevarán a cabo todos los sprint necesarios de SCRUM para un determinado semestre académico. Así visto, la aplicación de las iteraciones de la metodología híbrida se ven reflejadas, en cuanto a las fechas del proyecto de grado de la siguiente manera:

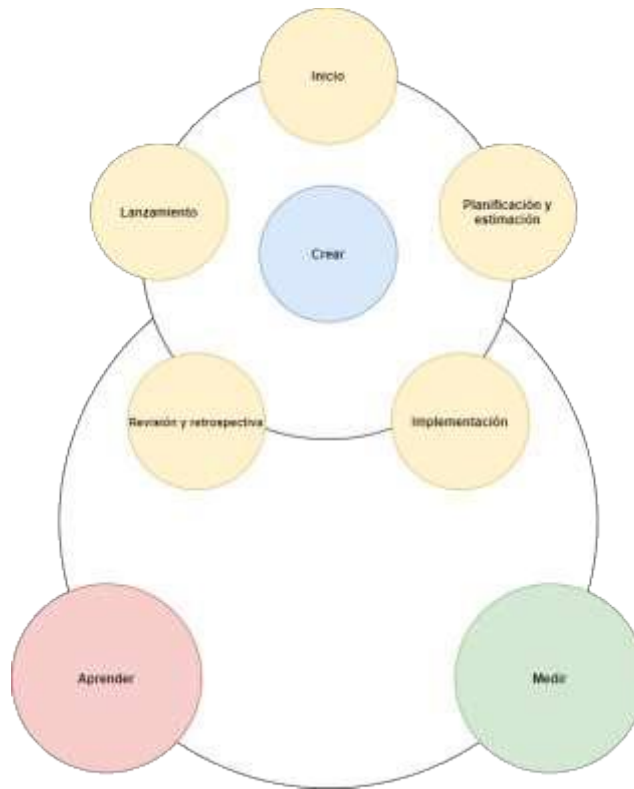
1. Las fases “medir” y “aprender” de Lean se llevarán a cabo, respectivamente, durante las lecturas, evaluaciones y/o sustentaciones con los lectores/jurados, y durante el proceso de realimentación posterior a esto, al igual que con las pruebas de software y la validación ante la población

objetivo. Lo anterior quiere decir que las fases “medir” y “aprender” de Lean se realizarán durante y al finalizar los cursos de Seminario de Investigación (semana 11 a 16), Proyecto de Grado 1 (semana 11 a 16) y Proyecto de Grado 2 (semana 3, 4, 7 y 8).

2. La fase “crear” de Lean se llevará a cabo durante las 11 primeras semanas del curso de Proyecto de Grado 1 y durante las semanas 1, 2, 5 y 6 del curso Proyecto de Grado 2, siendo estas semanas donde se realizarán las iteraciones de SCRUM.

En la siguiente imagen [Ilustración 5] se puede observar el diagrama que expone la combinación entre ambas metodologías que se va a implementar en el proyecto.

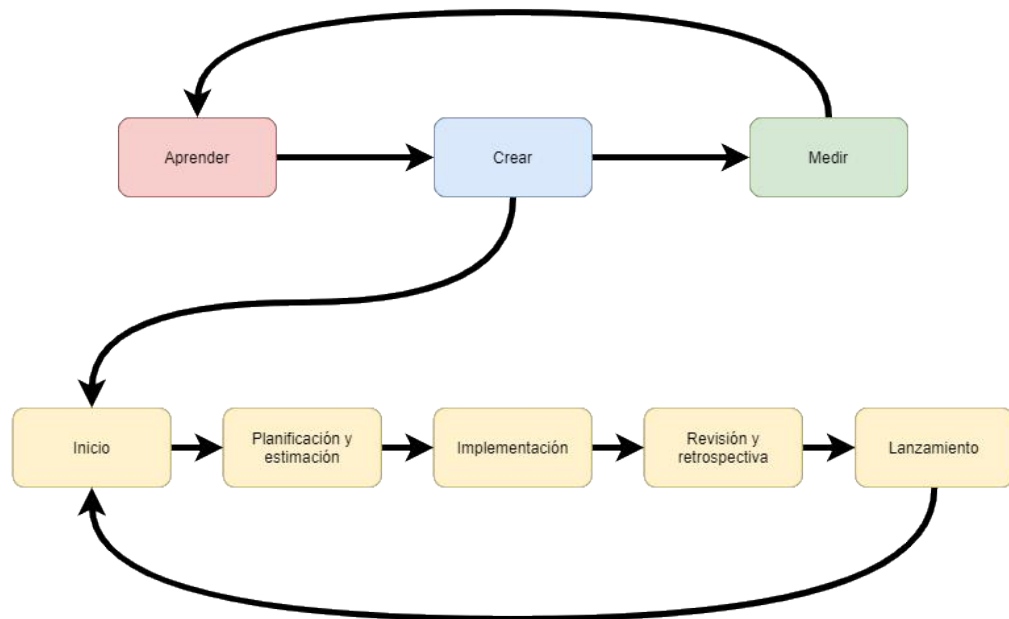
Ilustración 4. Diagrama general de las fases de la metodología híbrida (Lean-Agile).



Fuente: Elaboración propia.

Este diagrama se ve traducido, en cuanto a flujo de trabajo, en la siguiente imagen [Ilustración 6], donde se consideran las iteraciones, correspondientes a SCRUM, que se deben realizar en la fase “crear” de Lean.

Ilustración 5. Diagrama de aplicación de las fases de la metodología híbrida (Lean-Agile).



Fuente: Elaboración propia.

2.7.4.3 Principios

Para el desarrollo del proyecto RetoñosApp se considerarán simultáneamente los principios de SCRUM y los principios para el desarrollo de software dispuestos por Lean, de estos últimos considerando también los desperdicios. Esto puesto que los principios entre una metodología y la otra no se contradicen, sino que se complementan, entre otras cosas, de la siguiente manera:

1. Los principios de Lean se enfocan en agregar valor y optimizar procesos, mientras que los principios de SCRUM se enfocan en el desarrollo de un proyecto de calidad, pero de una forma ágil (rápida).
2. SCRUM al tener tiempos bien determinados para la entrega de avance es coherente con el principio de Lean de hacer entregas constantes y a su vez con la búsqueda por eliminar el desperdicio de “tiempo de espera”.
3. SCRUM con sus procesos de realimentación y sus etapas de retrospectiva es congruente con el principio de Lean de amplificar el aprendizaje de los errores, aciertos y retos identificados durante el proceso de desarrollo.
4. SCRUM está altamente enfocado en los productos (artefactos y entregables), mientras que Lean busca optimizar y aumentar el valor de dichos artefactos y entregables para el cliente.

2.7.4.4 Artefactos y entregables

Los artefactos que se entregarán como sustento y evidencia de la aplicación de la metodología planteada son:

- Formato de especificación de Sprints (que especificara el Sprint híbrido).
- Actas de reunión con los stakeholders.
- Tablero de gestión tareas.
- Prototipos incrementales.
- Producto Mínimo Viable (MVP).
- Producto final.

Dichos artefactos estarán distribuidos, según sea el caso, entre el informe final, el acta de constitución, el documento SAD y el documento de aplicación de la metodología.

Para el caso del primer semestre de realización del proyecto, correspondiente a la asignatura de Proyecto de Grado 1, los artefactos puntuales a entregar, como sustento y evidencia de la aplicación de la metodología, son los siguientes:

- Formatos de especificación de Sprints del tiempo transcurrido durante la asignatura de Proyecto de Grado 1.
- Tablero de gestión de tareas.
- Prototipo incremental.
- Producto Mínimo Viable (MVP).

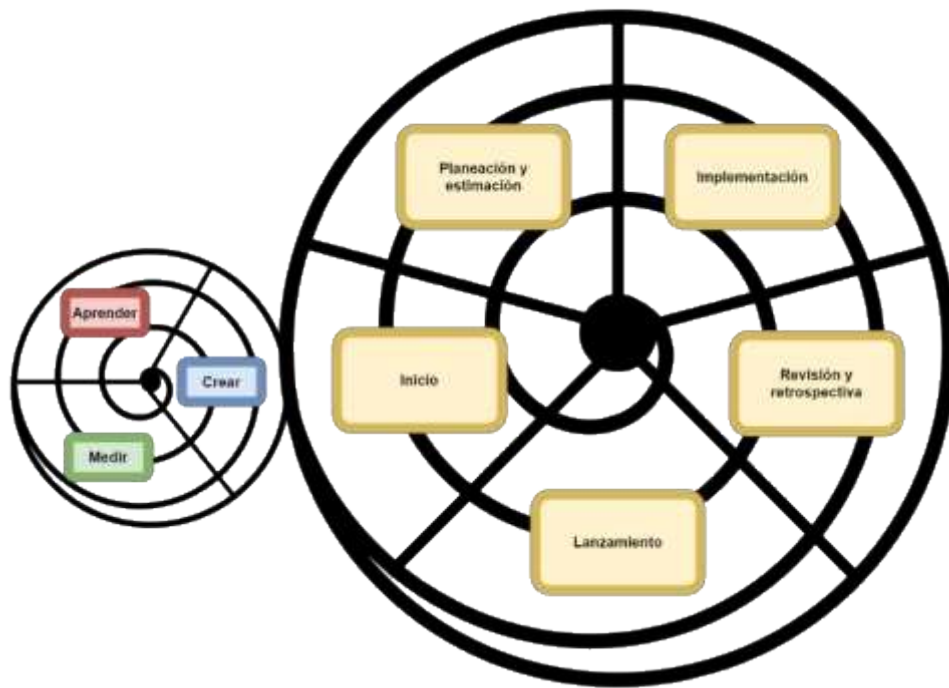
Para el caso del segundo semestre de realización del proyecto, correspondiente a la asignatura de Proyecto de Grado 2, se entregarán la totalidad de los artefactos que sustentan y evidencian la aplicación de la metodología. Lo anterior como componente de la finalización total del proyecto y la consecuente entrega de todos los elementos que lo constituyen.

2.7.4.5 Modelo y ciclo de vida

Para el desarrollo del proyecto RetoñosApp se utilizará un modelo en espiral de dos niveles, y un ciclo de vida iterativo e incremental. En lo referente a la espiral de dos niveles, esto se refiere a que el nivel principal representa las 3 fases de Lean y el nivel secundario representa las 5 fases de SCRUM, estando dicho nivel secundario derivado del nivel inicial (específicamente en la fase crear de Lean).

El modelo en espiral descrito anteriormente corresponde al presentado en el la siguiente imagen [Ilustración 7].

Ilustración 6. Diagrama del modelo en espiral de la metodología híbrida (Lean-Agile).



Fuente: Elaboración propia.

Por su parte el ciclo de vida iterativo e incremental es consistente con el enfoque de ambas metodologías de ser iterativas y trabajar bajo prototipos funcionales (prototipos de SCRUM y MVP de Lean) que se corrigen, se ajustan y cambian con el objetivo de evolucionar en un producto final.

2.7.4.6 Estructura de desglose de trabajo (EDT/WBS)

En el anexo 4 [Sección 11.4] se encuentra WBS que desglosa las tareas que se deben llevar a cabo para la realización del proyecto RetoñosApp, estando basado en el cronograma de actividades [Sección 2.7.7] e identificando de forma jerárquica las actividades a realizar. Igualmente se utiliza el mismo código de colores que se ha utilizado en los diagramas de las fases de la metodología [Figura 3] y [Figura 4] y en el modelo en espiral aplicado en función de la metodología [Figura 5].

2.7.4.7 Aplicación de la metodología

La metodología será desarrollada, será realizada por medio de Sprints, siguiendo el esquema general que plantea SCRUM para la ejecución de su metodología de desarrollo. Pero, al estar este proyecto desarrollado por medio de una metodología híbrida, los Sprint serán iteraciones de las fases planteadas por Lean, donde internamente, por medio de la fase crear, se realizarán las iteraciones planteadas por SCRUM. Específicamente, las iteraciones están planteada por medio de 8 Sprint, de los cuales 5 corresponden al curso de Proyecto de Grado 1, y 3

corresponden al curso de Proyecto de Grado 2. Dichos Sprint se distribuyen de la siguiente manera:

1. Durante Proyecto de Grado 1 se llevarán a cabo 3 Sprints correspondientes a la fase crear de Lean, lo que a su vez implica que internamente se están realizando 3 iteraciones de las fases de SCRUM. Estos 3 Sprints se organizan, en cuanto a las semanas, de la siguiente manera:
 - I. Semanas 1 – 4.
 - II. Semanas 5 – 8.
 - III. Semanas 9 – 11.

2. Durante Proyecto de Grado 1 se llevará a cabo 1 Sprint correspondiente a la fase medir de Lean y un Sprint correspondiente a la fase aprender de Lean. Estos 2 Sprints se organizan, en cuanto a las semanas, de la siguiente manera:
 - I. Medir: Semanas 12 – 14.
 - II. Aprender: Semanas 15 y 16.

3. Durante Proyecto de Grado 2 se llevarán a cabo 2 Sprints generales, lo que quiere decir que son 2 Sprint de 4 semanas en donde se realizará una iteración completa a todo el ciclo de Lean, pasando por sus tres etapas definidas, pero con la particularidad de que en ambas iteraciones se realizaran dos iteraciones internas en la fase crear. Lo anterior implica que estos 2 Sprints generales se organizan, en cuanto a las semanas, de la siguiente manera:
 - I. Sprint 1:
 - i. Crear: Semanas 1 y 2.
 - ii. Medir: Semana 3.
 - iii. Aprender: Semana 4.
 - II. Sprint 2:
 - i. Crear: Semanas 5 y 6.
 - ii. Medir: Semana 7.
 - iii. Aprender: Semana 8.

4. Finalmente, se cuenta con un “Sprint” de cierre, destinado a la revisión final del artefacto, los entregables restantes más allá del artefacto y a la preparación de la presentación/sustentación final. En este se busca finiquitar los detalles menores de cada elemento y se realizará una serie de simulaciones de sustentación, estando esta última destinada a la identificación de falencias menores en cualquiera de los entregables y a la preparación robusta de la sustentación final. Este Sprint se organiza

correspondería a la semana 9 y 10 de Proyecto de Grado 2, considerando especialmente que las sustentaciones comienzan desde la semana 11.

2.7.4.8 Estudio sobre la población objetivo

El estudio sobre la población objetivo, con el fin de realizar la recabación de requerimientos, identificación de necesidades y oportunidades. Para lo anterior, este se realizará de dos formas diferentes:

1. Un estudio, por medio de un cuestionario, realizado a los estudiantes de los cursos de Fundamentos de Programación y Programación 1 en los periodos 2021-01, 2021-02 y 2022-01.
2. Una entrevista con los docentes de los cursos de Fundamentos de Programación y Programación 1 cursos (de los horarios diurnos) para establecer su perspectiva frente a los cursos (en cuanto a lo metodológico, al temario y a los objetivos de cada curso) al igual que apelar al conocimiento acumulado que poseen al haber impartido dichos cursos por más de un periodo académico, el cual se puede ver traducido en un mayor entendimiento sobre las características generales y particulares de la población objetivo.

El análisis de estas dos variantes permitirá identificar de una forma más amplia y profunda los requerimientos y necesidades que posee la población objetivo frente al aplicativo propuesto, al igual que las oportunidades que presenta esta población. A su vez, al no estar limitada a un único grupo, considerablemente específico, o a un único periodo académico, el estudio proveerá una visión y una comprensión más integral de la población objetivo y de la problemática descrita para la cual se desarrolla el presente proyecto.

2.7.4.9 Validación con la población objetivo

Se diseñará e implementará una batería de pruebas de usuario, el cual será entregado a los estudiantes de los cursos de Fundamentos de Programación y Programación 1, el que medirá dos aspectos diferentes del aplicativo:

1. Una validación técnica que permita establecer el nivel de funcionalidad y calidad del aplicativo (definidas por los modelos de calidad establecidos [Sección 3.2.4]) al igual que para identificar cualquier falla existente a corregir previo a la entrega del producto final.
2. Una validación de satisfacción del usuario, la cual a través del concepto de experiencia del usuario [Sección 3.2.4] permita establecer el nivel de aceptación, satisfacción y funcionalidad práctica del aplicativo, evaluando el desempeño de este frente a la población objetivo y la percepción de esta

frente a la eficacia que tiene el aplicativo como herramienta de apoyo a la enseñanza impartida en el aula de clase.

Dicha batería de pruebas estará diseñada de forma integral (como un todo), pero el formulario de valoración que se le facilitará a los estudiantes estará dividido en dos partes: la primera parte será en la que se medirán los aspectos técnicos y funcionales, y la segunda parte medirá la satisfacción del usuario frente al aplicativo. A su vez, se le pedirá al docente una valoración, en términos de satisfacción del usuario, frente a la usabilidad del aplicativo y la información que este provee (reportes) referente a los procesos de aprendizaje de los estudiantes. Esto último se realizará por medio de un formulario de valoración diferente al que tendrán los estudiantes.

Los resultados de la validación frente a la población objetivo se determinarán por medio de un proceso estadístico, teniendo en cuenta que las pruebas en general y todas las respuestas en particular serán medidas a través de una o ambas escalas de las descritas posteriormente [Sección 3.2.3].

2.7.4.10 Gestión del trabajo en equipo

El trabajo en equipo se realizará y/o gestionará a través de las siguientes herramientas:

- Google Meet: Se utilizará para la realización de las reuniones con los evaluadores, el director de proyecto de grado y la población objetivo. Igualmente se utilizará para toda reunión que no haya sido previamente estimada por parte del equipo de trabajo y que por ende no esté planteada en el cronograma de actividades.
- Google Drive: Destinada a la gestión colaborativa de todos los archivos, diferentes al código fuente, correspondientes al proyecto de grado, tales como: documentos, diagramas, imágenes, presentaciones, videos, hojas de cálculo, entre otros.
- OneDrive: Apoyo a Google Drive, para el trabajo colaborativo del equipo de trabajo y con su director de proyecto, donde cualquier cambio realizado será actualizado en la carpeta oficial del proyecto ubicada en Google Drive.
- GitHub: Repositorio para la el versionamiento del código fuente y producto funcional a desarrollar.
- Trello: Medio para gestionar la realización de tareas y el cumplimiento de plazos de entrega de cada hito por parte de los integrantes del equipo, al igual que se relacionarán las tareas con las diferentes fase e incrementos (sprints) del proyecto.
- Discord: Útil en la realización de las reuniones y sesiones de trabajo grupales, del equipo de desarrollo, que no involucren al director de proyecto de grado o a alguna otra persona.

- WhatsApp: Medio de comunicación base entre el equipo de desarrollo y con el director del proyecto de grado.
- Gmail: Comunicación formal del equipo de desarrollo con su director de proyecto de grado, sus evaluadores, la población objetivo y cualquier otra persona que pueda intervenir en el proyecto en algún punto de la realización de este.
- Draw.io: Esta herramienta se utilizará para el diseño y desarrollo colaborativo de cualquier diagrama que se requiera durante la realización del proyecto, indistintamente de que estos diagramas se realicen por alguna necesidad especial o se requieran como parte de algún entregable.
- Mendeley: Para la gestión de todas las referencias bibliográficas que se utilicen en el transcurso del proyecto.

Cabe aclarar que estas herramientas están pensadas para una adecuada gestión del trabajo colaborativo en equipo, posibilitando incluso el trabajo colaborativo en tiempo real que facilite un adecuado versionamiento de documentos, archivos, código o diagramas.

2.7.4.11 Tablero de gestión de tareas

Para la gestión de tareas y el trabajo en equipo se hará uso del tablero Trello (<https://trello.com/es>), el cual funciona bajo un sistema Kanban, en donde se especificarán, aparte de las tareas a realizar, todos los sprint con los cuales cuenta el proyecto, para dentro de estos relacionar sus tareas específicas.

2.7.4.12 Reuniones

Las reuniones identificadas como parte del proceso de desarrollo y de trabajo en equipo incluye, pero no se limitan a:

- Reunión general semanal: Reunión en la cual se realiza un análisis general de lo realizado en la semana, se asignan las tareas a realizar la semana siguiente, se evalúan los entregables de cada miembro y se tocan todos los temas que los diferentes miembros consideren pertinentes.
- Reuniones con el director de proyecto.
- Reuniones con los lectores/jurados.
- Reuniones con la población objetivo para la realización del estudio, las pruebas y validación de la solución.

2.7.4.13 Cronograma de actividades

El cronograma de actividades relaciona las tareas específicas a realizar con las fechas de entrega determinadas para dichas tareas, siendo así un documento/elemento que está en constante cambio, evolución y movimiento. En el

caso del proyecto RetoñosApp se tiene diseñado su cronograma de actividades a través de Google SpreedSheets.

Para su visualización, el cronograma de actividades puede encontrarse en el anexo 5 [Sección 11.5].

2.8 COMPONENTE ÉTICO

El componente ético constituye un elemento transversal y fundamental tanto en la formulación, planteamiento y despliegue de un proyecto, indistintamente del campo del saber y/o aplicación al cual dicho proyecto pertenezca. Específicamente, el componente ético constituirá el marco y el estándar filosófico y moral frente al cual se va a ceñir el equipo de trabajo para la ejecución de cada fase del proyecto, al igual que sobre el cual se van a desarrollar cada uno de los entregables. Así visto, este será la descripción formal del conjunto de normas éticas que buscan regular el actuar del equipo de trabajo y la realización de cada elemento tanto tangible como intangible que compone el proyecto.

Para el caso del proyecto RetoñosApp, el componente ético estará constituido mediante los marcos éticos establecidos por el “Código de Ética y Conducta Profesional” [25] de la *Association for Computing Machinery* (ACM), específicamente enmarcado en el “Código de Ética de la Ingeniería de Software” [26] desarrollado en conjunto por la ACM y la *Institute of Electrical and Electronics Engineers* (IEEE), en particular por el *IEEE Computer Society* (IEEE – CS). A su vez, y considerando el contexto particular colombiano, este marco ético será respaldado por el “Código de Ética para el ejercicio de la Ingeniería en general y sus profesiones afines y auxiliares” [27] del Consejo Profesional Nacional de Ingeniería (COPNIA). Lo anterior implica que se van a utilizar, de forma directa, los elementos planteados en los dos primeros marcos mencionados, pero teniendo en cuenta las limitaciones o facultades que, estando presentes en el tercer marco mencionado, puedan ser de utilidad para el desarrollo del proyecto RetoñosApp.

Particularmente el proyecto considera de forma puntual cada uno de los 8 principios planteados en el “Código de Ética de la Ingeniería de Software” [26], pero igualmente, priorizando del “Código de Ética y Conducta Profesional” [25], los elementos planteados en su correspondiente sección 1. Lo anterior quiere decir que el desarrollo del producto de software estará éticamente limitado por los parámetros establecidos en el “Código de Ética de la Ingeniería de Software” [26]. Pero a su vez, se velará por el respeto a los principios éticos generales que debe de tener un profesional del área de la computación planteados en la sección 1 del “Código de Ética y Conducta Profesional” [25].

De forma específica, en lo referente a los 8 principios planteados en el “Código de Ética de la Ingeniería de Software” [26], se hará especial énfasis e hincapié en los siguientes elementos:

1. Principio 1: Apartados 1.01, 1.03, 1.04, 1.06 y 1.07.
2. Principio 2: Apartados 2.02, 2.03, 2.04, 2.05, 2.06, 2.07, 2.08.
3. Principio 3: Apartados 3.01, 3.025, 3.03, 3.04, 3.05, 3.06, 3.07, 3.08, 3.09, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15.
4. Principio 4: Apartados: 4.03.
5. Principio 5: Apartados: 5.01, 5.02, 5.03, 5.04, 5.05.
6. Principio 6: Apartados: 6.01.
7. Principio 7: Apartados: 7.01, 7.02, 7.04, 7.06, 7.07, 7.08.
8. Principio 8: Apartados: 8.01, 8.02, 8.03, 8.04, 8.05, 8.06, 8.07, 8.08, 8.09.

3. MARCO REFERENCIAL

En el presente capítulo se expondrán los antecedentes y el marco teórico que sustentarán al proyecto desde las perspectivas académicas, conceptuales, tecnológicas, procedimentales y contextuales. Igualmente, estos se enlazan intrínsecamente con el desarrollo metodológico y con los objetivos, el alcance y los resultados propuestos y esperados.

Al terminar este capítulo, el lector va a adquirir de conocimientos básicos y fundamentales para entender el proyecto desde la perspectiva bibliográfica, académica, histórica y conceptual. Lo anterior, parte en la sección 3.1 con los antecedentes y el estado del arte en donde se realiza una presentación/exposición sobre los trabajos previos y documentados, que se puedan considerar iguales, similares o que se relacionan con el presente proyecto ya sea por su idea, sus objetivos o su medio para llegar a estos. Esto puede respaldar el proyecto al plantear una guía base para su desarrollo, pero a su vez ofreciendo la información necesaria para encontrar y plantear un diferenciador. Finalmente, en la sección 3.2 se presenta el marco teórico, el cual sustenta el proyecto desde la bibliografía y la academia, al igual que plantea un sistema y conjunto integro, coordinado y coherente de conocimiento previo existente, validado y confiable que soporte el proyecto, lo oriente y lo justifique.

3.1 ANTECEDENTES Y ESTADO DEL ARTE

Considerando las características particulares del proyecto RetoñosApp, los antecedentes se pueden revisar desde tres perspectivas principales. Primero, la utilización de plataformas interactivas que sean un complemento y/o apoyo para los procesos de enseñanza y aprendizaje impartidos en el aula de clase o en las sesiones sincrónicas frente a las ciencias de la computación. Segundo, la utilización de bots conversacionales a modo de tutor virtual. Y finalmente, la implementación de sistemas de recomendación para la identificación de falencias y/o dificultades de enseñanza y aprendizaje. Cabe aclarar que, como referencias documentales, principalmente se encuentran artículos científicos, artículos a modo de memoria de un proyecto de investigación y/o ingeniería, o artículos resultado de ponencias en congresos de educación, congresos de ciencia y congresos de tecnología.

En lo referente a la utilización de plataformas interactivas que sean un complemento y/o apoyo para los procesos de enseñanza y aprendizaje se destacan los siguientes artículos:

Bedour Alshaigy [42], de la Universidad Oxford Brookes, realizó el planteamiento de una herramienta de e-learning interactiva para la enseñanza del lenguaje de

programación Python. En dicho proyecto Alshaigy implementaba, de forma digitalizada estrategias tales como: el utilizar problemas y ejercicios de aplicación de los conceptos, realizar desarrollo del pensamiento algorítmico sin tener presente un lenguaje de programación particular, la aplicación de desafíos y acertijos coherentes y alineados con las temáticas del curso, pero que a su vez mejoraran el desarrollo de las habilidades de pensamiento lógico de los estudiantes, y finalmente la visualización gráfica de los diferentes conceptos y el flujo de trabajo de los algoritmos buscando así una mejor apropiación de los mismos. Mediante lo anterior, sostiene que la interactividad propende por una comprensión más precisa, profunda y rápida de los conceptos, puesto que hace al alumno participe activo de su proceso formativo.

Dahlan Nariman [43], de la Universidad Asia Pacifico Ritsumeikan, realizó un estudio sobre la efectividad de la instrucción por medio de las plataformas de e-learning interactivas en los cursos de programación. Su estudio estuvo particularmente respaldado por la necesidad e importancia que representaba el aumento de conocimiento referente a la utilización de las herramientas digitales como parte de los procesos de enseñanza puesto que este se desarrolló en el año 2020 durante la pandemia del Covid-19. En este, Nariman expone que los estudiantes perciben un impacto positivo en sus procesos de aprendizaje al utilizar plataformas interactivas como medio de enseñanza. A su vez, expone que entre los principales atributos que intervienen en dicha percepción se encuentran la claridad gráfica y visual de los recursos presentados, la calidad de las explicaciones en video y la disponibilidad de la plataforma y las consecuentes explicaciones temáticas.

Viska Mutiawani [44], de la Universidad Syiah Kuala, presenta el planteamiento y la implementación de una herramienta de e-learning interactiva enfocada a la enseñanza en un curso de “Introducción a la Programación”. Lo anterior como consecuencia de la importancia y obligatoriedad que tienen los estudiantes en el campo de la informática y las ciencias de la computación en cuanto a adquirir y comprender los conocimientos de forma satisfactoria, como parte fundamental de su ejercicio profesional. Igualmente, en su artículo expone que en la actualidad las universidades, en cuanto a la utilización de plataformas digitales, apuestan a aquellas que consisten en “Sistemas de Gestión del Aprendizaje” (SGA), pero que resultan ser tanto rígidas como poco atractivas, al igual que solamente realizan la labor de intermediario asincrónico entre el docente y el estudiante. Por lo cual, y en respuesta a lo anterior, sustenta su proyecto en que las herramientas interactivas propenden por una mayor individualidad e independencia al momento del aprendizaje, al igual que benefician el interés y la disciplina por parte del estudiante.

Sasithorn Chookaew, Patcharin Panjaburee y Parames Laosinchai [45], del Instituto para el Aprendizaje Innovador de la Universidad Mahidol, en conjunto con Dechawut Wanichsan [45], de la Universidad Rambhai Barni Rajabhat, proponen un ambiente de aprendizaje personalizado por medio de e-learning como medio y herramienta para propiciar la habilidad de los estudiantes en torno al aprendizaje de los

conceptos de la programación. En el artículo, se expone que las plataformas de e-learning son óptimas para mejorar los procesos de aprendizaje, puesto que le permiten al estudiante individualizar su proceso, de tal forma que estos sean capaces de desarrollar y construir su conocimiento de forma personal. Dicha individualización se da, principalmente, por el hecho de que los estudiantes pueden hacer énfasis en sus necesidades particulares, lo que se ve traducido en que estos manejan su propio ritmo de aprendizaje y pueden enfatizar en las temáticas, conceptos o elementos que les sean de especial consideración.

En lo referente a la utilización de bots conversacionales a modo de tutor virtual se destacan los siguientes artículos:

Natalie Wellnhammer, Mateusz Dolota, Susanne Steigler y Gerhard Schwabe [46], de la Universidad de Zurich, realizaron un estudio sobre las características que han de tener los bots conversacionales para su correcto desempeño en ambientes virtuales de aprendizaje en general, y como tutores virtuales en particular. El artículo hace una división en torno a 5 aspectos principales a considerar en la construcción del bot conversacional que vaya a realizar la tarea de tutor virtual: (1) características técnicas, (2) características didácticas, (3) propósito como tutor, (4) tipos y mecanismos de diálogo y (5) características físicas (en caso de que el bot conversacional haga a su vez uso de un avatar o animación que acompañe al discurso). Posteriormente, subdivide cada aspecto en elementos y clasificaciones más atómicas con las cuales caracterizar mejor cada uno de dichos aspectos.

Christian Gabriel, Charlotte Hahne, Alina Zimmermann y Florian Lenk [47], de la Universidad Politécnica de Dresde, realizaron un estudio con el objetivo de identificar las tareas que deberían de realizar los tutores virtuales como parte de un entorno de aprendizaje colaborativo en línea (fundamentalmente un entorno de e-learning). El artículo se enfoca en presentar una visión generalizada de las tareas que los tutores virtuales pueden realizar en dichos entornos de e-learning, más allá de solamente limitarlos a la transmisión del conocimiento o “impartir” las clases. Igualmente, se plantea la posibilidad de la utilización de los bots conversacionales como medios para obtener información que luego puede ser utilizada para mejorar la experiencia de los estudiantes en la plataforma. A su vez, en el artículo se plantea un framework para el desarrollo de bots conversacionales enfocados a ser tutores virtuales, pero con la posibilidad de no tener las limitaciones previamente mencionadas.

Xavier Sánchez Díaz, Gilberto Ayala-Bastidas, Pedro Fonseca y Leonardo Garrido [48], del Tecnológico de Monterrey, plantean una metodología para el desarrollo de bots conversacionales a modo de tutores virtuales. La metodología expuesta se enfoca en el uso de formalismos matemáticos, específicamente enmarcados en la lógica de predicados, mediante los cuales representar el conocimiento del cual deben disponer los bots conversacionales, para luego plantear el flujo de

conversación de una forma híbrida entre una aproximación matemática, y una aproximación más “tradicional” como lo es el uso de *Intents* y *Entities*.

Finalmente, en cuanto a la implementación de sistemas de recomendación para la identificación de falencias y/o dificultades de enseñanza/aprendizaje, se destacan los siguiente artículos:

Paola Andrea Otero Cano y Edgar Camilo Pedraza Alarcón [49], de la Unidad de Tecnología e Innovación del Centro de Desarrollo Tecnológico CreaTIC, realizaron una revisión de la literatura existente en torno a la utilización de sistemas de recomendación en el área de la educación, teniendo como énfasis aquellos utilizados en los entornos de e-learning. Este estudio parte del hecho de que los entornos de e-learning permiten un desarrollo de los procesos de aprendizaje de los estudiantes de una forma más personal e individual, permitiendo que estos respondan a los estilos de estudio y aprendizaje particulares de cada estudiante, al igual que en su perfil. Frente a la revisión de la literatura, entre otras cosas, se concluye que la utilización de sistemas de recomendación en entornos de e-learning pueden ayudar a mejorar la educación por medio de una mejoría y/o adaptación de los procesos de enseñanza y de aprendizaje. Igualmente, mencionan que estos sistemas permiten ayudar a plantear rutas de aprendizaje más eficientes para los estudiantes (tanto como conjunto, como individualmente).

Gabriel de Souza Leitão, Eduardo Bezerra Valentin, Elaine Harada Teixeira de Oliveira y Raimundo da Silva Barreto [50], de la Universidad Federal del Amazonas, realizaron una revisión sistemática de la literatura donde plantean la utilización de sistemas cognitivos para realizar recomendaciones en entornos educativos. En el artículo se expone la popularización de soportar la educación a través de la tecnología, pero en particular expone el aumento de los sistemas cognitivos para hacer recomendaciones en torno a los procesos de aprendizaje, las técnicas de enseñanza y aprendizaje y los recursos que se utilizan en los cursos.

Steven Harris y Vivekanandan Kumar [51], de la Universidad de Athabasca, realizaron un estudio sobre la utilización de sistemas para identificar las dificultades de aprendizaje de los estudiantes implementados en entornos virtuales de aprendizaje. Lo particularmente interesante de este artículo es el cómo no solo se limitan a identificar las dificultades de aprendizaje, sino que también se busca identificar otros aspectos que intervienen en la disminución de la capacidad o posibilidad de los estudiantes para el entendimiento y apropiación de las temáticas y conceptos, como lo pueden ser la frustración o la confusión.

En general, los artículos expuestos de forma explícita en este apartado, en conjunto con algunos de los documentos que sustentan el marco teórico a presentar en el siguiente apartado, exponen individualmente la implementación de entornos de e-learning para acompañar enseñanza y aprendizaje en las ciencias de la computación, los bots conversacionales desarrollados para realizar la labor de tutor

virtual o los sistemas de recomendación para identificar falencias y/o dificultades de enseñanza y aprendizaje. Pero a su vez, en conjunto, permiten identificar la existencia de trabajos de investigación y de producción ingenieril en torno a la utilización de plataformas de e-learning y/o de la aplicación de técnicas de inteligencia artificial para acompañar, mejorar, adaptar o complementar los procesos de enseñanza y aprendizaje, más tradicionales, llevados a cabo en sesiones de clase sincrónica.

3.2 MARCO TEÓRICO

3.2.1 La enseñanza y el aprendizaje de ingeniería y la programación

La ingeniería es un campo de conocimiento y profesión inherentemente multidisciplinar, compuesto de ramas diferentes, cuyo objetivo es la transformación de la realidad y el mundo por medio de la aplicación conocimientos de diversos campos. Lo anterior se da mediante el diseño y la construcción de artefactos (que pueden o no ser tangibles), que afectan, modifican y/o transforman algún aspecto de la realidad en uno o varios contextos específicos [6].

Es así, que la enseñanza de la ingeniería constituye un grado de complejidad considerablemente alto, puesto que a los estudiantes no solo se les debe enseñar los aspectos propios de su profesión, sino que se les debe instruir en conocimientos, técnicas, metodologías, escenarios de acción, entre otros aspectos, de otras áreas del saber humano. Esto implica que se debe trabajar en el desarrollo de la identidad académica propia y característica de la ingeniería, estando fundamentada en la multidisciplinariedad y por ende trabajando de fondo con la identidad académica de otras disciplinas [52].

Así visto, cuando se habla de la enseñanza de la ingeniería, se debe hacer énfasis en tres puntos focales importantes, los cuales juntos constituyen la identidad académica propia y característica de la ingeniería mencionada anteriormente [53]–[55]:

En primer lugar, se deben establecer las líneas divisorias entre las diferentes ramas en las cuales se puede dividir de la ingeniería (también entendidas como subcampos de la ingeniería). Esto permite poder determinar los campos específicos del conocimiento de los cuales estas se apoyan y toman conocimiento, y que consecutivamente se deben enseñar [53], [54]. Lo anterior se debe a que cada ingeniería particular es fundamentalmente diferente a las demás en función de los campos del conocimiento de los cuales hace uso para su constitución como una profesión por sí misma [53], [54].

En segundo lugar, la ingeniería se apoya de una serie de “lenguajes” propios que le posibilitan su quehacer académico y profesional, y a su vez posibilita una

diferenciación entre las diferentes variedades de ingenierías. Esto implica que como parte del planteamiento académico de una ingeniería en particular se debe de establecer una o más asignaturas (según se requiera) destinadas a enseñar aquel lenguaje propio, para que luego este se pueda utilizar de forma natural durante el resto del proceso de formación [53] (e incluso durante el ejercicio profesional).

Finalmente, la ingeniería enseña de forma simultánea tres aspectos claves de la profesión: el pensamiento lógico y analítico, el pensamiento creativo, y el diseño. Siendo estos tres las guías y los pilares para la posterior utilización de los conocimientos multidisciplinarios con los cuales cuenta la ingeniería [53]–[55] puesto que es a través de la unión de ellos que la ingeniería es capaz de aplicar dichos saberes multidisciplinarios para luego poder realizar la transformación de la realidad mencionada previamente [53]–[55].

En particular, la Ingeniería de Sistemas, la Ingeniería Informática y otras ramas de la ingeniería afines a esta, desarrollan conocimientos en el área de la programación, siendo un saber fundamental y una herramienta indispensable para el ejercicio profesional. Donde la programación requiere de la adquisición de una serie extensa de conocimientos y especialmente de habilidades, entre las que se encuentran, pero no se limitan a la resolución de problemas, el diseño de técnicas, el pensamiento abstracto, el pensamiento matemático, el pensamiento sistémico, entre otras [56]. Pero es esa extensa serie de habilidades lo que genera un alto grado de complejidad para el aprendizaje de la programación, puesto que si no se adquieren o se dominan se encontraran dificultades en torno a: comprender la sintaxis de los lenguajes a utilizar, pensar de forma sistémica, utilizar estructuras básicas para conformar estructuras más avanzadas, generar una representación mental del problema y/o de la solución de este, traducir las ideas en productos (tangibles o intangibles), entre varias otras [22], [57], [58].

Por lo anterior, entre las estrategias para la enseñanza de la programación se pueden resaltar tres practicas principalmente. En primer lugar, la utilización de problemas y/o proyectos para aproximar a los estudiantes a contextos o puntos de aplicación de los conocimientos y habilidades, lo que ha demostrado efectividad, pero a la vez versatilidad y velocidad como técnica de enseñanza/aprendizaje [59] al igual que posibilita acercar a los estudiantes a la “realidad” mientras al mismo tiempo adquieren las competencias buscadas. Por otro lado, se busca atacar las dificultades técnicas que los estudiantes puedan presentar a través de talleres y ejercicios de codificación [42], utilización de bibliografía, aplicación de la competitividad y herramientas tecnológicas tales como foros, simuladores, entre otras [13], [21], [60]. Finalmente, la utilización de plataformas digitales en conjunto con la enseñanza presencial impartida en el aula para beneficiar los procesos cognitivos de los estudiantes [42], [61], [62] lo que a su vez es coherente puesto que se utiliza tecnología para enseñar tecnología [63].

Para terminar, cabe resaltar que las dificultades de la programación surgen en el individuo, pero se extienden al trabajo colaborativo que es indispensable en el ejercicio de la ingeniería. Por lo cual este trabajo colaborativo se aplica en el aula de clase para el desarrollo de las habilidades del estudiante referentes al trabajo en equipo [19], [64], al igual que para desarrollar desde ya una filosofía de trabajo que sea coherente con la realidad a la que se enfrentarán en su vida profesional.

3.2.2 Estrategias de apoyo a la enseñanza-aprendizaje en ciencias de la computación

3.2.2.1 Constructivismo en la educación

De forma general, el constructivismo es una teoría, corriente y filosofía de la pedagogía y la psicología que plantea que los individuos son actores activos en sus procesos de desarrollo personal, social, académico y profesional. Para lo anterior, el constructivismo sostiene que a los individuos se les han de otorgar las herramientas para que estos sean los responsables de formar sus propios conocimientos, experiencias, vivencias, aprendizajes, esquemas mentales, y de moldear su personalidad [65], [66] y en general de moldear su propio entendimiento del mundo [67].

En particular, el constructivismo en la educación tiene su enfoca a la formación del conocimiento en los alumnos, planteándolos como actores primarios y activos de la construcción de su propio conocimiento (i.e, protagonista de su propio aprendizaje). Lo anterior se logra mediante la facilitación de los instrumentos, elementos, espacios y “libertades” necesarias para que sea el estudiante quien desarrolle su propia experiencia de formación académica y desarrollo cognitivo [68].

A su vez, el constructivismo también recae en que los estudiantes no solo deben limitarse únicamente a las herramientas que se les facilitan, sino que sean también capaces de construir conocimiento a través de sus experiencias personales. Esto se da puesto que la asimilación, comprensión, entendimiento e incluso la memorización de conceptos, nociones, ideas, entre otros, parte de la interpretación y representación interna del mundo [65], [66], donde la interpretación y representación nacen de las experiencias vividas previamente y las conclusiones (aprendizajes) que quedan de cada una de ellas [65], [66], [68].

Igualmente, el hecho de que a los estudiantes se les faciliten las herramientas para que ellos desarrollen su aprendizaje, implica la necesidad y la posibilidad de que también tengan y puedan (respectivamente) desarrollar otras habilidades y capacidades. El desarrollo del pensamiento crítico, pensamiento analítico, pensamiento sistémico o pensamiento creativo (o simplemente del pensamiento en general), es un ejemplo de lo anterior, siendo también la principal habilidad que se desarrolla por medio y como consecuencia del constructivismo [67].

3.2.2.1.1 Constructivismo social

El constructivismo social, es una teoría, derivada del constructivismo, que plantea que el conocimiento no solo se construye y desarrolla de forma individual, sino que este proceso es de naturaleza colaborativa. Es decir, que el aprendizaje se forma no solo en la individualidad [Sección 3.2.2.1], sino que este requiere de la integración, la creación de relaciones, la interacción y la colaboración del individuo con los demás individuos que lo rodean [69].

En primer lugar, hay que resaltar la cooperación como parte fundamental en la teoría del constructivismo social, puesto que es gracias a ella que nace la extensión misma de “social” en la teoría constructivista. Lo anterior parte del hecho de que los estudiantes tienen la posibilidad de utilizar sus propios conocimientos, experiencias y esquemas mentales, para ayudar a otros a que desarrollen los suyos, y de forma recíproca recibir la misma ayuda para desarrollar los propios [69]–[71].

También, cabe aclarar que uno de los principales aportes del constructivismo social, frente al constructivismo tradicional, es el planteamiento del docente como una entidad mediadora y facilitadora del aprendizaje. Esto significa que el docente no está para definir o establecer el proceso o las limitaciones del aprendizaje [70], [71], sino que su labor es la de ser un guía, buscando mantener al estudiante centrado y principalmente siendo quien le provee estrategias de validación del conocimiento que el alumno construya [70], [71].

3.2.2.2 Enseñanza y aprendizaje asistido por computadora

La enseñanza asistida por computadora parte de la transformación tecnológica y digital frente a la cual todos los campos del conocimiento y trabajos se ven enfrentados, siendo dicha transformación una parte esencial del desempeño profesional futuro de los estudiantes. Dicha transformación tecnológica hace referencia a que conforme avanza y evoluciona la tecnología, los demás sectores (entre ellos el educativo) deben aprender a hacer uso de esta para hacer más práctico, variado, eficiente y productivo su quehacer [63].

Es así, que el uso de las herramientas de enseñanza asistida por computadora constituye una de las principales metodologías de enseñanza actuales. Estando distribuidas entre simuladores, plataformas de aprendizaje en línea, sistemas de realimentación, etc [72].

De igual forma, dicha aproximación tecnológica se ve traducida en un modelo constructivista del aprendizaje. Esto le permite a los estudiantes ser partícipes activamente de su proceso educativo, en donde estas herramientas le permiten enfrentar y superar sus dificultades de aprendizaje y/o alcanzar los objetivos académico [73], [74].

3.2.2.3 Utilización del e-learning como metodología para la enseñanza y el aprendizaje en ciencias de la computación

El e-learning, abreviación del término *electronic learning*, hace referencia a la enseñanza y aprendizaje soportado o por medio de plataformas en línea, siendo un medio por el cual las personas pueden complementar o reemplazar a la enseñanza tradicional. Siendo parte de las herramientas de la enseñanza asistida por computadora [75], y consecuencia directa de la expansión del internet y de la búsqueda de las personas por formarse académicamente [43], las plataformas de e-learning suponen una alternativa a los métodos clásicos de enseñanza, que le permite a las personas optar exclusivamente por este medio para su formación [45], [75] o les posibilita combinarlo con los métodos tradicionales de enseñanza formando lo que se conoce como *blended learning* [43].

En el caso de la educación en ciencias de la computación, el uso de herramientas digitales resulta ser una conclusión coherente con los conocimientos y herramientas de las cuales los estudiantes disponen para su labor académica y con su quehacer profesional como partícipes del desarrollo tecnológico.

A su vez, posibilita a los estudiantes apoyar su aprendizaje en el aula, con un medio personalizado que sea coherente con sus estilos de aprendizaje y personalidades individuales. Lo anterior implica que la enseñanza se centra en el estudiante y no solo en la teoría, lo que le permite al alumno desarrollar sus habilidades de aprendizaje en simultáneo con la apropiación de los conceptos y temáticas impartidas en clase [76].

De igual forma la personalización de la enseñanza, el enfoque constructivista que suponen estas herramientas y la educación centrada en el alumno, propende por el beneficio de la motivación en el aprendizaje [77], mientras que también se convierte en un apoyo tecnológico para el docente en el aula de clase [44], para complementar la instrucción e incluso identificar falencias conceptuales que los estudiantes puedan poseer.

3.2.2.4 Bots conversacionales como tutores virtuales

Los bots conversacionales (conocidos en inglés como chatbots) consisten en sistemas informáticos “inteligentes” capaces de sostener conversaciones fluidas en lenguaje natural (lenguajes humanos) de forma similar o prácticamente igual a como lo hace una persona real. Para ello utilizan mecanismos y técnicas propias de la inteligencia artificial y el machine learning [48], [78].

Estos tienen una extensa serie de aplicaciones tanto empresariales, comerciales o incluso de ocio. Sus aplicaciones incluyen, sin limitarse a, los videojuegos, llegando hasta la educación y el servicio al cliente [48], donde se puede resaltar, como una de las aplicaciones más conocidas, su implementación en los llamados “asistentes

de voz” tales como Cortana de Microsoft, Siri de Apple y Alexa de Amazon, entre otros [46].

Por otro lado, el uso de chatbots como tutores virtuales es una práctica ya conocida, en donde, desde una perspectiva pedagógica, los bots conversacionales ayudan a proveer un soporte al aprendizaje de los estudiantes y son una parte vital en la individualización de los procesos de aprendizaje. Esto quiere decir que permiten que cada estudiante tenga la posibilidad de manejar su ritmo de aprendizaje, al igual que establecer su propio estilo de estudio en función de su personalidad, sus conocimientos previo, sus habilidad, sus capacidades cognitivas e incluso de sus motivaciones [47].

Igualmente, el uso de chatbots como tutores virtuales, hace partícipe al estudiante en la construcción activa de su conocimiento, mientras a su vez promueve la conservación de la motivación durante el proceso de aprendizaje gracias a la interactividad que esta tecnología posibilita [46].

Ahora bien, las ventajas que los bots conversacionales poseen se pueden dividir principalmente en cinco: la posibilidad de individualizar el aprendizaje de los estudiantes, ofrecer una atención personalizada a cada alumno, atender simultáneamente a un gran grupo de estudiantes, estar disponibles 24/7 y obtener información referente a los procesos de aprendizaje de cada educando [48], [79], [80].

3.2.2.5 Gamificación como metodología de enseñanza de la programación

La gamificación consiste en una técnica y metodología (i.e., siendo básicamente un sistema o una herramienta), que busca aplicar conceptos, características y mecánicas de los juegos (e.g., de mesa, de video, tradicionales, etc). Dicha ampliación hacer referencia a extender las particularidades de los juegos a contextos que no sean necesariamente lúdicos [81], [82].

Este sistema tiene como objetivo aumentar el grado de motivación, persistencia, concentración, gusto, entre otros, que las personas tienen hacia una determinada tarea, actividad u ocupación. Lo anterior implica que la gamificación permite mantener al estudiante centrado y estimulado a través de una metodología de enseñanza “más lúdica” [81]–[83].

En algunos casos incluso se utiliza un enfoque de resolución de problemas y persecución de objetivos [83], al igual que resulta ser una herramienta que desarrolla el pensamiento creativo y lógico de forma simultánea [84], [85].

La gamificación ha tomado especial relevancia en el campo de la educación, en donde a través de ella se busca atacar la falta de interés y motivación que los estudiantes presentan en su proceso de aprendizaje. Lo anterior se da por medio

de hacer a los estudiantes partícipes de la edificación de su propio conocimiento, siendo así parte de una corriente teórica de la educación conocida como constructivismo y educación activa [81], [85].

La enseñanza de la ingeniería y de la programación no son la excepción, puesto que se ha demostrado que este sistema genera en los estudiantes de estas áreas un sentido de autosuperación y de trabajo en equipo (aspecto primordial para el ejercicio de cualquier rama de la ingeniería). Lo que quiere decir que la gamificación le enseña a los estudiantes a establecer metas y objetivos durante su proceso académico, al igual que les enseña a cooperar con sus pares para alcanzar dichas metas y objetivos [81], [86].

Igualmente desarrolla en estos, de forma simultánea, el pensamiento analítico y lateral, beneficiando así la capacidad de pensamiento sistémico, de utilización de procesos metodológicos para la realización de tareas y de resolución de problemas. Esto implica que la gamificación propende por el desarrollo intelectual y del pensamiento convergente y divergente de forma conjunta y como una sinergia, y no como formas de pensar opuestas, que les permiten realizar sus actividad y solucionar las dificultades a las que se enfrenten [84], [86].

3.2.2.6 Sistemas de recomendación como medio para ajustar las metodologías de enseñanza en la educación

Los sistemas de recomendación son una herramienta que establece un conjunto de criterios y valoraciones sobre un conjunto de datos para realizar predicciones sobre recomendaciones de elementos que puedan ser de utilidad o valor para el usuario. Esto le permite conocer opciones o elementos que puedan ser de su interés y así personalizar su experiencia en múltiples contextos [87].

En el campo de la educación se genera una gran cantidad de información proveniente de herramientas de aprendizaje virtuales, tales como: cursos en línea, sitios de networking para estudiantes, foros de diferentes temas, aplicaciones educativas, entre muchos otros. Dicha información parte de que estas herramientas virtuales de aprendizaje le permiten al usuario interactuar y/o participar, más que solo “buscar información o respuestas” [88].

Por todo anterior, una de las principales ventajas que los sistemas de recomendación presentan en el ámbito educativo consiste en que estos constituyen un medio y/o una guía para poder ajustar la estructura curricular o las metodologías de enseñanza en un determinado curso, convirtiéndolos así una herramienta que permite realimentar los procesos educativos. Dichos ajustes se realizan por medio de los sistemas de recomendación teniendo como insumo la información proveniente de estos mismos [89], [90]. De igual forma, estos permiten personalizar o adaptar la experiencia de aprendizaje en entornos de aprendizaje ya sean

virtuales o físicos, lo que beneficia los procesos de aprendizaje y la consecución de los objetivos académicos [49], [90].

3.2.3 Satisfacción: Que es y cómo se mide

La satisfacción, definida en general como un sentimiento de placer, bienestar y realización [91], [92], consiste en un estado mental y emocional que es considerado sociológica y psicológicamente como positivo para un individuo [92].

Específicamente, la satisfacción del consumidor consiste en un concepto primordial en los campos del marketing y el emprendimiento que hace referencia a la medida de la satisfacción percibida por los clientes en torno a los productos, servicios, atención, entre otros factores de una compañía[93]. En general se puede entender como la cantidad o porcentaje de consumidores que reportan una experiencia cuya valoración iguala o supera los niveles mínimos de satisfacción establecidos [93].

Hoy existen múltiples formas de medir la satisfacción de una persona, indistintamente del contexto en el cual se busque obtener dicha valoración. Lo anterior dado que existen aproximaciones cualitativas, cuantitativas, híbridas, entre otras [94], [95], para establecer medidas que puedan ser útiles y acertadas, entre otras cosas, en función del contexto en el cual se busque establecer dicha medición.

En la literatura se han expuesto estrategias varias para la medición o cálculo de la satisfacción orientada a productos o ideas. Siendo así, la escala Likert consiste en una de las metodologías más utilizadas para la medición, entre otras cosas, de la satisfacción u opinión de una persona frente a una determinada cosa, la cual consiste en una escala (valga la redundancia) que permite establecer puntos máximos y mínimos de valoración y una serie de puntos intermedios entre estos [94]. La anterior escala se estructura como un conjunto de posibilidades organizadas linealmente de menor a mayor (nivel o grado) en donde se utilizan explícita o implícitamente cantidades numéricas y/o etiquetas textuales que identifican los diferentes niveles de valoración de los cuales dispone una persona para realizar su calificación [96]. Por otro lado, también existe el Net Promoter Score (NPS), que permite, aparte de la ya establecida medición de la satisfacción, también la medida de la lealtad de un cliente frente a un producto, servicio o en general frente a una empresa [97]. Esa medida, ampliamente utilizada en el área del marketing, se enfoca principalmente en mantener a los clientes actuales más que ganar clientes nuevos y está enfocada en aspectos internos del producto o de la empresa más que en aspectos externos [97]. El Net Promoter Score consiste en una escala numérica del 0 al 10 que se ve reflejada en una fórmula matemática que se define como [98]:

$$\text{Net Promoter Score} = \% \text{ Clientes Promotores} - \% \text{ Clientes Detractores} = \% \text{ NPS}$$

En donde los clientes detractores consisten en aquellos que califiquen 0 – 6, los clientes promotores aquellos que califiquen 9 o 10 y los clientes neutros (que no se consideran en la ecuación) aquellos que califiquen 7 u 8 [98].

Una de las principales aplicaciones de la medida de la satisfacción, dejando de lado sus usos en el emprendimiento, marketing o en la prestación de servicios de salud, se da en el área de la educación, en donde gracias a esto se pueden obtener apreciaciones referentes a el grado de calidad de un curso, un docente, una actividad, una carrera, entre otros. En particular, la medición de la satisfacción es ampliamente utilizada para retroalimentar el grado de funcionalidad, utilidad, pertinencia y eficiencia de una herramienta de enseñanza digital [99], de tal forma que se puedan realizar ajustes para aumentar la calidad y productividad de estas herramientas.

3.2.4 Experiencia del usuario (UX)

La experiencia de usuario (UX por sus siglas en inglés) es la percepción de una persona al interactuar con un producto o servicio, siendo básicamente definido como la forma en la cual una persona se siente al interactuar con un sistema. Desde la perspectiva del desarrollo de software, se puede entender como una combinación o sinergia entre la usabilidad que presenta un producto de software y el grado de satisfacción que el usuario de este percibe al utilizarlo [100].

Cabe aclarar que según diferentes modelos de calidad, entre los que se encuentran la norma ISO 25000 [101], el modelo de calidad de Boehm [102], o el modelo de calidad de McCall [102], la usabilidad se entiende como la facilidad y comodidad de utilización de una herramienta de software, incluyendo la curva de aprendizaje que conlleva, la protección contra errores del usuario, la estética de la interfaz de usuario y la accesibilidad que presenta indistintamente de que el usuario posea o no alguna discapacidad.

La experiencia del usuario, como medida de validez y/o calidad de un software, se puede validar a través de los mismos modelos, métricas o escalas que funcionan para la medición de la satisfacción de una persona ante un determinado elemento. Lo anterior debido a la característica de la experiencia de usuario de ser básicamente el grado de satisfacción percibido por un usuario frente a un producto de software y el nivel de usabilidad de este [103]–[105].

4. DESARROLLO METODOLÓGICO

En el presente capítulo se explicará el desarrollo del proyecto desde la perspectiva de la metodología planteada [Sección 2.6], sus fases, artefactos, y elementos constituyentes. A su vez este capítulo presentará las evidencias y/o sustentos correspondientes que respalden la aplicación de la metodología durante el desarrollo del proyecto.

Al terminar este capítulo, el lector tendrá una visión clara sobre el proceso de desarrollo del proyecto, en concordancia con la metodología planteada [Sección 2.6], siendo esta el marco de trabajo que guía el desarrollo de este. Lo anterior, parte en la sección 4.1 con la especificación de Sprints, como elemento base y guía para todo el desarrollo del proyecto de grado. Luego, en la sección 4.2 se realiza la descripción del proceso del estudio sobre la población objetivo, al igual que el detalle de cada fase de este. Posteriormente, en la sección 4.3 se detallan los requerimientos funcionales [Sección 4.3.1] y los requerimientos no funcionales [Sección 4.3.2], siendo estos los dos elementos que definen y limitan el proyecto en términos de sus funcionalidades, tecnologías, aspectos de diseño, aspectos de negocio y aspecto gráficos. Finalmente, en la sección 4.4 se explica el planteamiento realizado, tanto de forma interna en el equipo de desarrollo, como con el director de proyecto de grado, para todo lo referente al trabajo colaborativo (tanto sincrónico, como asincrónico), siendo esto lo que determina, en una última instancia, una ejecución satisfactoria y eficiente del proyecto.

4.1 ESPECIFICACIÓN DE SPRINTS

Las siguientes tablas [Tabla 2]–[Tabla 9] corresponden a la especificación de Sprints pertenecientes al desarrollo metodológico a realizar en la asignatura de Proyecto de Grado 1.

Estas tablas constituyen una explicación global de las actividades a realizar en cada uno de los Sprints detallando a su vez: (1) su estimación en cuanto a número de semanas y fechas específicas en las cuales se va a desarrollar cada uno de los Sprints, (2) las entradas y salidas que se utilizan como insumo y que se generan como resultados de cada Sprint correspondientemente, (3) la especificación en cuanto a las fases de Lean y SCRUM a las cuales pertenece cada Sprint, y finalmente (4) la delimitación en cuanto al número de incremento presentado en cada Sprint.

A su vez, estas tablas, en conjunto con el cronograma de actividades y el tablero de control constituyen los elementos encargados de exponer, almacenar y especificar las actividades a realizar en cada etapa del proceso de desarrollo del proyecto. Lo anterior implica que estos tres elementos se complementan entre si al ofrecer en

conjunto una visión macro y microscópica del proceso de desarrollo del proyecto en cuanto al seguimiento de las fases planteadas frente a la metodología híbrida propuesta.

Tabla 2. Especificación Sprint 1 (Crear).

ID	SP - 001	Nombre	Sprint 1 (Crear)		
Duración	4 semanas	Fecha de inicio	16 de agosto	Fecha de finalización	12 de septiembre
Descripción	En este Sprint se realiza todo el planteamiento inicial y sustento base del proyecto, donde esto incluye pero no se limita a: la estructuración del equipo SCRUM, la estimación y especificación de Sprints, la definición de tecnologías y de características gráficas principales, el planteamiento de las reuniones periódicas, la definición del plan de trabajo general, al igual que se realiza el diseño y creación de los formatos documentales de: acta de constitución, documento SAD, formatos de especificación, actas de reunión, bitácora y reflexiones periódicas. A su vez, es durante este Sprint en donde se realiza el empalme del desarrollo propio del proyecto con las actividades netamente académicas de cada uno de los miembros del equipo de trabajo.				
Entradas	No existen entradas previas, puesto que es en este primer Sprint que se inicia formalmente con el desarrollo del proyecto.	Resultado esperado / Producto esperado / Entregables	<ul style="list-style-type: none"> • Especificación de Sprints. • Estructuración del equipo SCRUM. • Formatos documentales especificados. • Tecnologías plenamente definidas. Incremento sobre el aplicativo funcional: 0		
Fases de la metodología incluidas	<p style="text-align: center;">Lean: Crear (X) Medir () Aprender ()</p> <p style="text-align: center;">Scrum: Inicio (X) Planeación y estimación (X) Implementación () Revisión y retrospectiva () Lanzamiento ()</p>				

Fuente: Elaboración propia.

Tabla 3. Especificación Sprint 2 (Crear).

ID	SP - 002	Nombre	Sprint 2 (Crear)		
Duración	4 semanas	Fecha de inicio	13 de septiembre	Fecha de finalización	10 de octubre
Descripción	En este Sprint se comienza con todo el desarrollo técnico e ingenieril del proyecto, con énfasis en el artefacto. Es decir, es en este Sprint que se comienza con el desarrollo del producto de software, así como con la realización de toda la documentación correspondiente y pertinente. Lo anterior hace referencia a la construcción del artefacto que constituye la innovación/solución planteada, al igual que el desarrollo de los documentos que sustentan el proyecto tanto desde el ámbito ingenieril, como desde el ámbito académico. A su vez, se realizan los ajustes e incrementos necesarios y pertinentes a la documentación de soporte del proyecto realizada (o comenzada) en el Sprint anterior.				
Entradas	Resultados esperados, productos esperados o entregables del Sprint 1.	Resultado esperado / Producto esperado / Entregables	<ul style="list-style-type: none"> • Primer incremento del producto de software funcional. • Diagramas de la arquitectura, clases, modelos de bases de datos (referente a los modelos relacionales). Incremento sobre el aplicativo funcional: 1		
Fases de la metodología incluidas	<p style="text-align: center;">Lean: Crear (X) Medir () Aprender ()</p> <p style="text-align: center;">Scrum: Inicio () Planeación y estimación () Implementación (X) Revisión y retrospectiva () Lanzamiento ()</p>				

Fuente: Elaboración propia.

Tabla 4. Especificación Sprint 3 (Crear).

ID	SP - 003	Nombre	Sprint 3 (Crear)		
Duración	3 semanas	Fecha de inicio	11 de octubre	Fecha de finalización	31 de octubre
Descripción	En este Sprint se continúa con el desarrollo del producto de software y con la realización de toda la documentación correspondiente y pertinente (de la misma manera en que se realizó durante el Sprint 2), al igual que se hacen los ajustes necesarios y pertinentes a los diagramas y a las especificaciones resultantes del Sprint 2.				
Entradas	Resultados esperados, productos esperados o entregables tanto del Sprint 1 y Sprint 2.	Resultado esperado / Producto esperado / Entregables		<ul style="list-style-type: none"> • Segundo incremento del producto de software funcional. • Especificación de los requerimientos funcionales y no funcionales. • Diccionario de la base de datos. <p>Incremento sobre el aplicativo funcional: 2</p>	
Fases de la metodología incluidas	<p>Lean: Crear (X) Medir () Aprender ()</p> <p>Scrum: Inicio () Planeación y estimación () Implementación (X) Revisión y retrospectiva (X) Lanzamiento (X)</p>				

Fuente: Elaboración propia.

Tabla 5. Especificación Sprint 4 (Medir).

ID	SP - 004	Nombre	Sprint 4 (Medir)		
Duración	3 semanas	Fecha de inicio	1 de noviembre	Fecha de finalización	21 de noviembre
Descripción	En este Sprint se continúa con el desarrollo del producto de software y con la realización de toda la documentación correspondiente y pertinente, al igual que se hacen los ajustes necesarios y pertinentes a los diagramas y a las especificaciones resultantes del Sprint 2. A su vez se realiza el planteamiento de la arquitectura basada en atributos de calidad, como elemento que relaciona el artefacto con los atributos y parámetros de calidad propios del proyecto en general y del aplicativo en particular. Finalmente, se realiza la primera validación frente a la población objetivo, en la cual se medirá su percepción, en torno a la experiencia de usuario, frente a los videos interactivos y el bot conversacional.				
Entradas	Resultados esperados, productos esperados o entregables del los 3 Sprints anteriores.	Resultado esperado / Producto esperado / Entregables		<ul style="list-style-type: none"> • Incremento parcial con respecto tercer incremento del producto de software funcional. • Resultados de una validación parcial del artefacto frente a la población objetivo. <p>Incremento sobre el aplicativo funcional: 2.5</p>	
Fases de la metodología incluidas	<p>Lean: Crear () Medir (X) Aprender ()</p> <p>Scrum: Inicio () Planeación y estimación () Implementación () Revisión y retrospectiva () Lanzamiento ()</p>				

Fuente: Elaboración propia.

Tabla 6. Especificación Sprint 5 (Aprender).

ID	SP - 005	Nombre	Sprint 5 (Aprender)		
Duración	2 semanas	Fecha de inicio	22 de noviembre	Fecha de finalización	5 de diciembre
Descripción	En este Sprint se finaliza el desarrollo del producto de software y con la realización de toda la documentación correspondiente y pertinente a entregar como parte del 70% del proyecto establecido y solicitado en el desarrollo del proyecto. A su vez, en este Sprint se realiza la evaluación y valoración final de los docentes de la asignatura de Proyecto de Grado 1, así como de los jurados evaluadores del proyecto. Y como consecuencia de lo anterior, se obtiene una realimentación con la cual se realizarán los respectivos ajustes a la metodología, el plan de trabajo, los entregables realizados, el artefacto y en general al planteamiento del proyecto para así poder continuar con su desarrollo.				
Entradas	Resultados esperados, productos esperados o entregables de los 4 Sprints anteriores.	Resultado esperado / Producto esperado / Entregables	<ul style="list-style-type: none"> Despliegue inicial del aplicativo para presentación como parte de la evaluación de jurados correspondiente a la asignatura de Proyecto de Grado 1. Informe final. Incremento sobre el aplicativo funcional: 3		
Fases de la metodología incluidas	<p style="text-align: center;">Lean: Crear () Medir () Aprender (X)</p> <p style="text-align: center;">Scrum: Inicio () Planeación y estimación () Implementación () Revisión y retrospectiva () Lanzamiento ()</p>				

Fuente: Elaboración propia.

Tabla 7. Especificación Sprint 6 (Híbrido).

ID	SP - 006	Nombre	Sprint 6 (Híbrido)		
Duración	4 semanas	Fecha de inicio	8 de febrero	Fecha de finalización	5 de marzo
Descripción	En este Sprint se continúa con el desarrollo del proyecto, pero bajo los lineamientos de la asignatura de Proyecto de Grado 2. Lo anterior toma relevancia considerando que este periodo está principalmente enfocado a el despliegue y la validación del artefacto, al igual que en los aspectos académicos del proyecto. Este Sprint está destinado a realizar una iteración completa sobre el ciclo de Lean, con el fin de realizar un incremento completo en el artefacto. Igualmente, y desde la perspectiva de Scrum, la fase de "crear" de Lean hará especial enfoque a labores de implementación y validación en la academia.				
Entradas	Resultados esperados, productos esperados o entregables de los 5 Sprints anteriores.	Resultado esperado / Producto esperado / Entregables	<ul style="list-style-type: none"> Especificación de los casos de uso, restricciones, atributos de calidad y motivadores de negocio. Diagramas de la arquitectura, contexto, despliegue, paquetes y casos de uso. Incremento sobre el aplicativo funcional: 4		
Fases de la metodología incluidas	<p style="text-align: center;">Lean: Crear (X) Medir (X) Aprender (X)</p> <p style="text-align: center;">Scrum: Inicio () Planeación y estimación () Implementación (X) Revisión y retrospectiva (X) Lanzamiento ()</p>				

Fuente: Elaboración propia.

Tabla 8. Especificación Sprint (Híbrido).

ID	SP - 007	Nombre	Sprint 7 (Híbrido)		
Duración	4 semanas	Fecha de inicio	5 de marzo	Fecha de finalización	2 de abril
Descripción	Este Sprint esta destinado a la finalización del artefacto de software, teniendo especial enfoque en la construcción e integración del sistema de recomendación con el resto del aplicativo ya desarrollado, al igual que en la terminación del bot conversacional (especialmente en lo que se refiere a completitud de tópicos). A su vez, este Sprint realiza un progreso en términos documentales, especialmente en lo que se refiere al informe final y los documentos propios de ingeniería.				
Entradas	Resultados esperados, productos esperados o entregables de los 6 Sprints anteriores.		Resultado esperado / Producto esperado / Entregables	<ul style="list-style-type: none"> • Especificación de los escenarios operacionales y escenarios de calidad. • Diagramas de actividades, secuencia y componentes. Incremento sobre el aplicativo funcional: 5	
Fases de la metodología incluidas	<p style="text-align: center;">Lean: Crear (X) Medir (X) Aprender (X)</p> <p style="text-align: center;">Scrum: Inicio () Planeación y estimación () Implementación (X) Revisión y retrospectiva (X) Lanzamiento ()</p>				

Fuente: Elaboración propia.

Tabla 9. Especificación Sprint 9 (Híbrido).

ID	SP - 008	Nombre	Sprint 8 (Híbrido)		
Duración	4 semanas	Fecha de inicio	3 de abril	Fecha de finalización	30 de abril
Descripción	Este es el Sprint final del proyecto, cuya fecha de finalización está prevista para una semana antes de la entrega final del proyecto según el calendario de la asignatura de Proyecto de Grado 2. Durante este Sprint se realiza todo el proceso de validación del artefacto (en la academia, estática y dinámica), al igual que se finalizan todos los documentos de ingeniería y el informe final del proyecto. Este Sprint constituye un Sprint híbrido general entre todas las fases de Lean con el objetivo de realizar un barrido entre todos los aspectos que componen al proyecto: técnicos, documentales, ingenieriles, académicos y administrativos.				
Entradas	Resultados esperados, productos esperados o entregables de los 7 Sprints anteriores.		Resultado esperado / Producto esperado / Entregables	<ul style="list-style-type: none"> • Artefacto final. • Documentos de ingeniería: Documento SAD, plan de riesgos, plan de calidad, plan de pruebas y acta de constitución. • Documentos de soporte: Bitácora, reflexiones periódicas y actas de reunión. • Informe final. • Presentación final. Incremento sobre el aplicativo funcional: 6	
Fases de la metodología incluidas	<p style="text-align: center;">Lean: Crear (X) Medir (X) Aprender (X)</p> <p style="text-align: center;">Scrum: Inicio () Planeación y estimación () Implementación (X) Revisión y retrospectiva (X) Lanzamiento (X)</p>				

Fuente: Elaboración propia.

4.2 ESTUDIO SOBRE LA POBLACIÓN OBJETIVO

Como parte del planteamiento presentado en torno a el estudio sobre la población objetivo [Sección 2.6.4.8], se realiza un estudio, desarrollado en diferentes fases e instantes del desarrollo del proyecto, con el fin de realizar la recabación de requerimientos, identificación de necesidades y oportunidades. El objetivo principal de este es poder identificar de una forma más amplia y profunda los requerimientos y necesidades que posee la población objetivo frente al aplicativo propuesto, al igual que las oportunidades que presenta esta población.

El estudio sobre la población objetivo se ejecuta por medio de tres instrumentos de investigación: entrevistas, cuestionarios y revisión bibliográfica y/o documental:

1. En el caso de los cuestionarios, se hace uso del formulario como instrumento de investigación (desarrollado e implementado por medio de Google Forms).
2. En lo referente a las entrevistas, estas se realizan de forma sincrónica, a través de Google Meet como medio de comunicación, y con el registro manual (digital) de la información, comentarios, afirmaciones y/o respuestas que provee el entrevistado, o en el caso de ser posible se realiza de forma presencial en un espacio disponible de la universidad (salas de estudio, salas de reuniones de la facultad, entre otros).
3. En cuanto a la revisión bibliográfica y/o documental, esta se desarrolla por medio de la búsqueda documental de trabajos de investigación, tesis o proyectos de grados previos ya finalizados, publicaciones producto de conferencias o cualquier otro documento de carácter académico u oficial (como por ejemplo la documentación oficial de las tecnologías utilizadas). La revisión bibliográfica y/o documental a su vez se respalda, en el caso de los documentos de carácter académico, que estas constituyan tesis de universidades debidamente reconocidas y con una trayectoria académica demostrable y de preferencia que estos sean documentos de investigación y artículos que estén publicados en una revista que este clasificada como Q1 en Scimago Journal & Country Rank (SJR). Lo anterior implica que de cada uno de documentos académico que son obtenidos de una base de datos o buscador académico, se contrasta la clasificación de la revista la clasificación de la revista en la cual esta publicado el documento (a través de SJR), para filtrar absolutamente todos los documentos cuyas revistas sean Q4 Y Q3, permitiendo que como máximo el 10 % de los documentos sean Q2, aclarando así que como mínimo el 90 % de artículos académicos utilizados como referencias en el presente proyecto se encuentran publicados en revistas clasificadas como Q1. Igualmente se aceptan como elementos bibliográficos y/o documentales, cualquier libro de texto cuyo autor o autores tengan una trayectoria académica comprobable en el campo de estudio y en la escritura académica.

4.2.1 Primera iteración del estudio sobre la población objetivo

La primera iteración del estudio sobre la población objetivo se realizó durante las semanas 15 y 16 del periodo académico 2021-01, correspondientes al periodo de desarrollo de la propuesta de proyecto de grado. En esta iteración se implementaron las aproximaciones 1 y 2 de las variantes planteadas [Sección 4.2]. Específicamente se realizó la entrevista a los profesores de Fundamentos de Programación y Programación 1. A su vez, se implementó la primera versión del cuestionario diseñado para los grupos de Programación 1, para la identificación de requerimientos, necesidades y oportunidades. La razón para seleccionar como primer grupo de estudiantes participantes a aquel grupo de Programación 1 radica en el hecho de que se buscaba, en primera instancia, poder acarrear las percepciones de Fundamentos de Programación y Programación 1, y que a su vez los participantes pudieran hacer una comparación entre las percepciones de ambas asignaturas (esto implica que se buscó aplicar el cuestionario a un grupo de Programación 1). Pero, particularmente este grupo posibilitaba realizar un contraste entre la percepción y los comentarios del docente con la percepción y los comentarios de sus respectivos alumnos, teniendo así ambas perspectivas que más que contrarias son inherentemente complementarias.

4.2.2 Segunda iteración del estudio sobre la población objetivo

La segunda iteración del estudio sobre la población objetivo se realizó durante la semana 5 del periodo académico 2021-02, perteneciente a la asignatura de Proyecto de Grado 1. Específicamente se implementó la segunda versión del cuestionario diseñado para los grupos de Fundamentos de Programación, para la identificación de requerimientos, necesidades y oportunidades, siendo esta implementada específicamente en los dos grupos pertenecientes a la jornada diurna. El hecho de haber realizado el cuestionario en un momento tan temprano del semestre, y especialmente a la asignatura inicial de la línea de Desarrollo de Software [7], radica en que se busca tener la perspectiva de los estudiantes en su etapa más temprana, cuando se puede afirmar que solamente han comenzado a vivir su etapa universitaria, y cuando aún están comenzando a desarrollar sus conocimientos y/o habilidades. Lo anterior beneficia al proyecto puesto que se pueden identificar las características de los estudiantes de nuevo ingreso del programa de Ingeniería de Sistemas, lo que permite modelar el MBPSC de una forma más orgánica, detallada e integral. A su vez, se busca, en el largo plazo, contrastar con la percepción de estos mismos estudiantes luego de que hayan finalizado la asignatura, por lo cual se les aplicará otra versión de su respectivo cuestionario.

4.3 ESPECIFICACIÓN DE REQUERIMIENTOS

Particularmente, en este proyecto los stakeholders no constituyen en sí una figura de “cliente”, por lo cual los requerimientos (tanto funcionales como no - funcionales)

a su vez no son impuestos o solicitados por medio de reuniones o actas, sino que estos se identificaron a través de 4 medios:

1. El estudio sobre la población objetivo, por medio de la cual se identifican las necesidades que esta presenta, pero a la vez las oportunidades de implementación de una solución tecnológica frente a la problemática identificada, y por ende planteada, como punto de partida del proyecto.
2. La investigación bibliográfica con la cual se complementa el estudio sobre la población objetivo para la descripción del contexto a intervenir y la problemática identificada.
3. El planteamiento y consecutivo análisis del MBPSC, mediante el cual se puede describir la problemática y la solución de una forma sistémica. A su vez este punto toma especial relevancia a estar el proyecto destinado específicamente a una población de la Universidad El Bosque.
4. La bibliografía propia de la Ingeniería del Software y la Ingeniería de Requerimientos, puesto que esta expone, valga la redundancia, requerimientos funcionales y no - funcionales genéricos que todo producto de software debe de considerar para alcanzar estándares mínimos esperados en determinados puntos de calidad.

4.3.1 Requerimientos funcionales

Los requerimientos funcionales son la definición de las diferentes funcionalidades de un producto de software, siendo a través de estos que, a nivel documental, se enlistan, exponen y explican cada una de dichas funcionalidades [106]. Los requerimientos funcionales son un detalle que constituyen, a nivel técnico y lógico, un desglose de las acciones que las diferentes actividades descritas a través de los casos de uso [107], [108].

La descripción de los requerimientos funcionales le permite a el equipo de trabajo desarrollar el artefacto, desde la perspectiva tecnológica (su código fuente, los servicios externos a utilizar y de la consecución de determinados aspectos de calidad) frente a las condiciones o peticiones que plantean (directa o indirectamente) los stakeholders.

Con lo anterior, el equipo de trabajo realizó la especificación de requerimientos funcionales, utilizando un formato que se planteó como combinación de un formato dispuesto en semestres anteriores como insumo de las asignaturas de Ingeniería de Software 1 e Ingeniería de Software 2, modificado para acoplarse al proyecto. Dicho formato constituye la estructura presentada en la siguiente tabla [Tabla 7]:

Tabla 10. Estructura del formato utilizado para la especificación de requerimientos funcionales.

ID	RF - 000	Nombre	
Descripción			
Contribución al producto			
Entrada			
Salida			
Precondición(es)			
Postcondición(es)			
Prioridad	Alta () Media () Baja ()		
Excepción(es)			

Fuente: Elaboración propia.

Las secciones del formato previamente presentado tienen los siguientes objetivos:

- ID: Es el identificador único del requerimiento funcional. La estructura que se maneja es: RF – 000, siendo RF las siglas de “Requerimiento Funcional”, y 000 un diferenciador numérico que asciende consecutivamente (e.g., 001, 002, 003, etc).
- Nombre: Es el nombre puntual del requerimiento funcional. Siendo una forma natural u “orgánica” de identificar a un requerimiento en particular.
- Descripción: Es la descripción general del requerimiento funcional, la cual otorga una explicación como respuesta a la pregunta ¿qué es lo que hace el requerimiento?
- Contribución al producto: Es la descripción de cuál es la utilidad para el usuario, para el negocio, para la tecnología o para otros requerimientos funcionales, que otorga la existencia de un requerimiento funcional puntual.
- Entrada: Son las entradas, en términos de datos, acciones o información que recibe un requerimiento funcional para poder realizar sus operaciones.
- Salida: Es el resultado que provee un requerimiento funcional, ya sea desde la perspectiva de un retorno de datos e información, o de la perspectiva de un evento que ocurre luego de que dicho requerimiento se utiliza.
- Precondición(es): Son las condiciones y el estado que se deben de cumplir en cuanto a la tecnología, la configuración, la información disponible, o las acciones del usuario, para que un requerimiento funcional pueda ejecutar sus operaciones.

- Postcondición(es): Son las condiciones y el estado posterior que tendrá el sistema luego de ejecutadas las operaciones de un requerimiento funcional.
- Prioridad: Es el “nivel de importancia” o “nivel de urgencia” que tiene un requerimiento funcional para ser implementado. Es decir que especifica que tan importante o urgente es un requerimiento funcional de ser implementado. Para este proyecto, dicha prioridad es medida mediante una escala ternaria compuesta por los rangos: alto, medio y bajo.
- Excepción(es): Son los posibles “errores” que pueden surgir antes o durante la ejecución de las operaciones de un requerimiento funcional que imposibilitan que estas finalicen, los cuales son anticiparles para definir así un esquema de control que posibilita que el aplicativo pueda continuar su flujo de trabajo sin ningún problema.

En la siguiente tabla [Tabla 6] se encuentra una indexación de los requerimientos funcionales, a modo de resumen, identificados para el proyecto, los cuales pueden encontrarse, en detalle, en el anexo 6 [Sección 11.6] correspondiente al documento SAD [Documento DOC002_DocumentoSAD]:

Tabla 11. Indexación de los requerimientos funcionales.

Código (Identificador)	Nombre del requerimiento
RF - 001	Iniciar sesión
RF - 002	Cerrar sesión
RF - 003	Registrar estudiante
RF - 004	Registrar docente
RF - 005	Registrar administrador
RF - 006	Cifrar contraseña
RF - 007	Modificar/actualizar estudiante
RF - 008	Modificar/actualizar docente
RF - 009	Modificar/actualizar administrador
RF - 010	Generar reportes
RF - 011	Descargar reporte PDF
RF - 012	Visualizar reporte
RF - 013	Registrar estudiante
RF - 014	Registrar docente
RF - 015	Registrar administrador
RF - 016	Crear asignatura
RF - 017	Crear temática
RF - 018	Crear sesión de clase
RF - 019	Modificar/actualizar asignatura
RF - 020	Modificar/actualizar temática
RF - 021	Modificar/actualizar sesión de clase
RF - 022	Bot conversacional
RF - 023	Enviar mensaje

Código (Identificador)	Nombre del requerimiento
RF - 024	Obtener respuesta
RF - 025	Rich response
RF - 026	Sistema de recomendación
RF - 027	Mostrar tutorial
RF - 028	Interactuar con el video
RF - 029	Storytelling del video
RF - 030	Buscar asignatura
RF - 031	Buscar temática
RF - 032	Buscar sesión de clase
RF - 033	Completar sesión
RF - 034	Calcular progreso
RF - 035	Registrar auditoría
RF - 036	Registrar conversación con el bot conversacional
RF - 037	Cargar archivo
RF - 038	Registrar ficha bibliográfica
RF - 039	Modificar/actualizar ficha bibliográfica
RF - 040	Registrar cuestionario
RF - 041	Modificar/actualizar cuestionario

Fuente: Elaboración propia.

4.3.2 Requerimientos no - funcionales

De forma complementaria a los requerimientos funcionales, los requerimientos no – funcionales se enfocan en detallar restricciones que se deben de considerar para el desarrollo de la solución, los cuales pueden ser aspectos tecnológicos, de interfaz de usuario, presupuestarios, organizacionales, de negocio, de rendimiento, entre otros [106]–[108].

Fundamentalmente, y en contraparte a los requerimientos funcionales que buscan responder a la pregunta: ¿Qué hace el sistema?, los requerimientos no funcionales buscan responder dos preguntas principalmente: ¿Cómo hace el sistema para cumplir sus funcionalidades? Y ¿Qué limitantes deben de considerarse para que el sistema pueda cumplir con sus funciones? [106]–[108].

Con lo anterior, y de igual forma que con los requerimientos funcionales, se diseñó un formato puntual, para la especificación de requerimientos no – funcionales, en el marco del proyecto. Dicho formato constituye la estructura presentada en la siguiente tabla [Tabla 9]:

Tabla 12. Estructura del formato utilizado para la especificación de requerimientos no - funcionales.

ID	RNF - 000	Nombre	
Dependencia			
Descripción			
Clasificación	Producto () Organización () Externo ()		

Fuente: Elaboración propia.

Las secciones del formato previamente presentado tienen los siguientes objetivos:

- ID: Es el identificador único del requerimiento no - funcional. La estructura que se maneja es: RNF – 000, siendo RF las siglas de “Requerimiento No - Funcional”, y 000 un diferenciador numérico que asciende consecutivamente (e.g., 001, 002, 003, etc).
- Nombre: Es el nombre puntual del requerimiento no - funcional. Siendo una forma natural u “orgánica” de identificar a un requerimiento en particular.
- Dependencia: Es el stakeholder o la unidad (del equipo de desarrollo) encargada que identifica o puntualiza el requerimiento no – funcional.
- Descripción: Es la descripción general del requerimiento no - funcional, la cual otorga una explicación como respuesta a la pregunta ¿cuál es la limitante en cuestión?
- Clasificación: Es la diferenciación entre los posibles tipos de requerimientos no – funcionales que se pueden presentar entre aquellos que se dan por un factor externo, un factor organizacional o que se enfocan exclusivamente en el producto.

En la siguiente tabla [Tabla 10] se encuentra una indexación de los requerimientos no - funcionales, a modo de resumen, identificados para el proyecto, los cuales pueden encontrarse, en detalle, en el anexo 6 [Sección 11.6] correspondiente al documento SAD [Documento DOC002_DocumentoSAD]:

Tabla 13. Indexación de los requerimientos no - funcionales.

Código (Identificador)	Nombre del requerimiento
RNF - 001	Tipo de aplicativo
RNF - 002	Reportes en PDF
RNF - 003	Base de datos
RNF - 004	Desarrollo Frontend
RNF - 005	Desarrollo Backend
RNF - 006	Seguridad
RNF - 007	Documentación del código
RNF - 008	Documentación del proyecto
RNF - 009	Interfaz gráfica del aplicativo

Fuente: Elaboración propia.

4.4 TRABAJO COLABORATIVO

El trabajo colaborativo hace referencia al esquema y las herramientas de trabajo síncrono y asíncrono entre el grupo de trabajo y director de proyecto de grado, que permitieran realizar un trabajo en equipo fluido, eficiente y fructífero, aprovechando las tecnologías y herramientas que se encuentran a disposición.

Particularmente, las herramientas de trabajo colaborativo utilizadas se pueden dividir en 6 tipos principales:

1. Herramientas de reunión:
 - a. Google Meet: Para la realización de las reuniones con el director de proyecto y con los demás Stakeholders.
 - b. Discord: Para la realización de las reuniones entre el equipo de trabajo.
2. Herramientas de comunicación:
 - a. WhatsApp: Para la comunicación asíncrona informal entre el equipo de trabajo y con el director de proyecto de grado.
 - b. Correo electrónico: Para la comunicación asíncrona con el director del proyecto y con los demás Stakeholders.
3. Herramientas de gestión de documentos:
 - a. Google Drive: Para la gestión de los documentos del proyecto, donde la organización del “repositorio” es una amalgama entre las indicaciones dadas en la asignatura de Proyecto de Grado 1, junto con algunas aproximaciones propias planteadas por el equipo de trabajo.
 - b. OneDrive: Para la gestión de los documentos con la posibilidad de poder utilizar todas las herramientas y ventajas propias de Microsoft Word. Esta herramienta se utiliza especialmente para los documentos textuales.
4. Herramientas de trabajo colaborativo:
 - a. Draw.io: Para la construcción de los diagramas (software, académicos y/o complementarios) del proyecto.

5. Herramienta de gestión de tareas: Para la gestión de las tareas se utilizó Trello, el cual funciona como tablero Kanban.
6. Herramienta de gestión del código fuente: Para la gestión del código fuente se utilizó GitHub. Por lo anterior, la gestión, en este caso, debe comprenderse como control de versiones y a su vez como el despliegue de incrementos, modificaciones, actualizaciones, correcciones, funcionalidades o builds en función de las responsabilidades (módulos, archivos o clases) que cada miembro del equipo tenga en torno a la construcción del artefacto.

4.5 GESTIÓN DEL TRABAJO EN EQUIPO

Para la gestión de tareas, asignaciones y responsabilidades como parte del trabajo colaborativo, y tal como se especificó en su respectiva sección [Sección 4.4], se utilizó un tablero Kanban, desplegado por medio de la herramienta Trello, para el control de las actividades a realizar, en función de lo especificado según el cronograma de actividades.

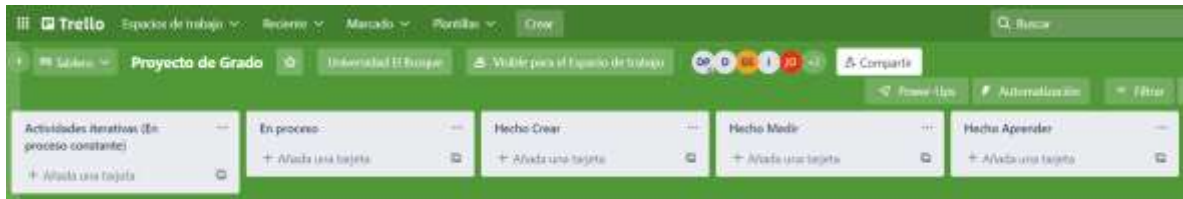
La organización del tablero definió tres espacios para la enumeración del product backlog (propio de SCRUM), pero en función de cada una de las 3 fases de Lean que constituían el ciclo principal (o primario) de la metodología híbrida construida [Sección 2.7.4]. Luego, se definían dos espacios para las actividades “en curso”. El primero de estos espacios se destinaba a las actividades que fueran de carácter iterativo, y que por ende se desarrollaran de forma constante y evolutiva a través del desarrollo de todo el proceso. Por su parte, el segundo de estos espacios se destinó a aquellas actividades que estaban en proceso de desarrollo, mas no eran de carácter iterativo-evolutivo. Finalmente, se definieron los 3 espacios para enumerar las actividades ya finalizadas, pero nuevamente en función de cada una de las 3 fases de Lean. La anterior organización del tablero Kanban del proyecto se puede evidenciar en las siguientes dos imágenes [Ilustración X], [Ilustración X]:

Ilustración 7. Espacios del tablero Kanban para el product backlog del proyecto en función de las fases de Lean.



Fuente: Elaboración propia.

Ilustración 8. Espacios del tablero Kanban destinados a las actividades iterativas, en progreso y finalizadas en función de las fases de Lean.



Fuente: Elaboración propia.

4.6 DESCRIPCIÓN DE LOS COMPONENTES UTILIZADOS DE CADA METODOLOGÍA QUE CONSTITUYE LA METODOLOGÍA HÍBRIDA DEL PROYECTO

Los siguientes elementos fueron aquellos elementos utilizados (o tenidos en cuenta) para el desarrollo del presente proyecto:

4.7 APLICACIÓN DE SCRUM

Para el desarrollo del proyecto RetoñosApp se utilizaron los siguientes elementos y conceptos definidos por SCRUM:

- Fases de SCRUM, las cuales componen la fase “crear” de Lean.
- Los roles de SRCUM (particularmente los roles de Product Owner, SCRUM Master y SCRUM Team).
- La determinación y especificación de Sprint.
- Determinación del producto backlog (que se realiza para cada fase de Lean).
- Generación de prototipos incrementales.
- Utilización de un tablero de tareas (específicamente un tablero estilo Kanban).
- Generación de un producto final.

4.8 APLICACIÓN DE LEAN

Para el desarrollo del proyecto RetoñosApp se utilizaron los siguientes elementos y conceptos definidos por Lean:

- Fases de Lean, las cuales componen el ciclo principal (las fases base) para el desarrollo del proyecto.
- Eliminación de desperdicios:
 - Eliminación, discriminación o combinación de cierto tipo de entregables potenciales.
 - Eliminación de procesos o partes de procesos que podían ralentizar el proceso de desarrollo del proyecto.

- Eliminación de la burocracia para la toma de decisiones.
- Eliminación de los defectos de software (lo cual se logró por medio de la construcción de una arquitectura “triple”, y la utilización de patrones de diseño y de asignación de responsabilidades).
- Una correcta identificación y especificación de tareas y actividades para evitar el cambio de estas.
- Amplificación del aprendizaje (lo cual se implementó por medio de la revisión y evaluación constante del artefacto y la documentación por parte del director de proyecto de grado, la profesora de la asignatura de Proyecto de Grado 1 y 2, y la población objetivo).
- Toma de decisiones constantes y rápidas sin requerir de reuniones extensas o disminuyendo la cantidad de votaciones necesarias.

5. RESULTADOS

En el presente capítulo se exhiben todos los aspectos referentes a los resultados, comenzando con la enumeración de los diferentes entregables del proyecto, pasando por el planteamiento de los resultados “objetivo” que se tienen para el proyecto, y finalizando con la presentación de estos.

Al terminar este capítulo, el lector va a conocer todos los resultados del proyecto RetoñosApp, comprendiendo cuales son los elementos que se entregarán como fruto del proyecto mismo, los resultados esperados (es decir el punto de comparación para determinar el éxito o fracaso del proyecto), y los resultados obtenidos luego del proceso de validación con la población objetivo. Lo anterior, parte en la sección 5.1 con el detalle de la distribución de porcentajes de completitud para el desarrollo del proyecto, en donde se relacionan los diferentes entregables de este con el periodo académico en el cual deben estar finalizados. Luego, en la sección 5.2 se describen los resultados esperados del proyecto en lo referente a las 3 fases de validación, pero especialmente en torno a la fase de validación dinámica, para lo cual se establecen los resultados esperados en términos de la variable de medición (y sus consecuentes subvariables). Finalmente, se exponen los resultados obtenidos durante la fase de validación dinámica, especificando cuales fueron los resultados finales, como se obtuvieron, como se calcularon, como están distribuidos y como se relacionan con la variable y subvariables de medición.

5.1 DISTRIBUCIÓN DE PORCENTAJES DE COMPLETITUD PARA EL DESARROLLO DEL PROYECTO

Según lo analizado y acordado en las reuniones con director de proyecto, los porcentajes de completitud del proyecto y los entregables (tecnológicos y documentales) del proyecto estarán distribuidos de la siguiente manera [Tabla 8]:

Tabla 14. Distribución de porcentajes de completitud para el desarrollo del proyecto

Asignatura	Porcentaje	Entregables correspondientes
Proyecto de Grado 1	70%	<ul style="list-style-type: none">• Producto Mínimo Viable (MVP): Corresponde al artefacto de software implementando las funcionalidades del bot conversacional funcional y los videos interactivos, al igual la primera versión viable del componente web “tradicional” constituida por los módulos de administrador, docente y estudiante.

Asignatura	Porcentaje	Entregables correspondientes
		<p>A su vez incluye la implementación de la base de datos relacional que respalda, en términos de persistencia, a el componente web mencionado anteriormente.</p> <ul style="list-style-type: none"> • Informe final a la fecha. • Artefactos de control a la fecha: <ul style="list-style-type: none"> ○ Bitácora. ○ Reflexiones periódicas. ○ Tablero de control. • Documento SAD a la fecha: <ul style="list-style-type: none"> ○ Propósito y audiencia. ○ Problema para resolver. ○ Descripción general del sistema a desarrollar. ○ Objetivos de la solución. ○ Stakeholders. ○ Descripción y justificación de la metodología a emplear con respecto al proyecto. ○ Recolección y especificación de requerimientos funcionales. ○ Colección y especificación de requerimientos no funcionales. ○ Diagramas de arquitectura. ○ Diagramas de clases. ○ Modelos de datos. ○ Diccionario de base de datos.
Proyecto de Grado 2	100%	Entregables descritos en la tabla de entregables del proyecto [Tabla 1].

Fuente: Elaboración propia.

5.2 RESULTADOS ESPERADOS

En congruencia con el planteamiento particular que tiene el programa de Ingeniería de Sistemas de la Universidad El Bosque, el proyecto RetoñosApp separa tres etapas de validación distintas:

1. Validación en la academia.
2. Validación estática.
3. Validación dinámica.

Para cada una de estas etapas se especificó una serie de variables de medición que respondan a su contexto y alcance particular, y que sean coherentes con la Ingeniería de Software, la Gestión de Proyectos, o los estudios de aceptación de productos o servicios por parte del cliente. Bajo ese enfoque, para cada una de las etapas de validación se determinaron las siguientes variables de medición:

1. Validación en la academia (las variables de medición propiamente se encuentran resaltadas en negrilla):
 - a. Prueba de humo: **Compilación.**
 - b. Prueba de humo: **Estabilidad.**
 - c. Prueba de sanidad: **Funcionalidad.**
 - d. Prueba unitaria: **Funcionalidad.**
 - e. Prueba de integración: **Funcionalidad.**
 - f. Prueba de integración: **Interacción.**
 - g. Prueba de seguridad: **Vulnerabilidades.**
 - h. Prueba de carga: **Estabilidad.**
 - i. Prueba de carga: **Integridad.**
2. Validación estática:
 - a. Completitud de los requerimientos funcionales.
 - b. Completitud de los requerimientos no funcionales.
 - c. Estandarización de los nombres de funciones.
 - d. Estandarización de los nombre de parámetros.
 - e. Estandarización de los nombres de las clases.
 - f. Versionamiento.
3. Validación dinámica:
 - a. Experiencia de usuario:
 - i. Satisfacción.
 - ii. Usabilidad:
 1. Funcionalidad.
 2. Diseño visual.
 3. Accesibilidad.
 4. Tasa de error.
 5. Facilidad de aprendizaje.
 6. Eficiencia de uso.

Considerando lo anterior, y en lo referente a los resultados esperados el proyecto RetoñosApp tiene dos orientaciones diferentes, la primera enfocada a la validación en la academia y la validación estática, y la segunda enfocada en la validación dinámica.

La primera orientación simplemente busca cumplir con los estándares generales de la industria del software en lo que se refiere a: (1) la consecución de un producto de calidad que responda tanto a un conjunto de atributos puntuales determinados a través de un modelo de calidad específico, (2) al igual que a la medición del grado

de cumplimiento frente a las necesidades y/o requerimientos del cliente y de la utilización de buenas prácticas de desarrollo y documentación de código. Por lo anterior, las variables de medición de la fase de validación en la academia están enfocadas en cuantificar una serie de características específicas del software que propendan por el alcance de un producto de calidad, mientras que las variables de la fase de validación estática están enfocadas en examinar si el producto final cumple o no con los requerimientos registrados en el documento SAD [Documento DOC002_DocumentoSAD], al igual que en si se utilizaron o no buenas prácticas durante el proceso de desarrollo de software y para el proceso de documentación del código fuente.

La segunda orientación busca medir cualitativamente la aceptación del artefacto final por parte de la población objetivo, a través de una “calificación” categórica que será una representación de las valoraciones o resultados cuantitativos que se obtengan al medir las distintas variables de medición de la etapa de validación dinámica. Esta valoración cualitativa tiene como objetivo una mejor y más sencilla interpretación de las respuestas dadas por los participantes y consecuentemente de los resultados del proyecto en general. Las categorías que se identificaron en función de los intervalos números conformados por los valores que constituirán las escalas Likert y Net Promoter Score a utilizar se presentan en la siguiente tabla [Tabla 15]:

Tabla 15. Categorías representativas de los resultados de la etapa de validación dinámica.

Valoración Cuantitativa	Valoración Cualitativa
8.6 – 10.0	Excelente
7.0 – 8.5	Sobresaliente
5.0 – 6.9	Aceptable
1.1 – 4.9	Insuficiente
0.0 – 1.0	Deficiente

Fuente: Elaboración propia.

Cabe aclarar que la idea para la utilización y el establecimiento de las anteriores categorías nace de los sistemas categóricos/cualitativos de calificación utilizados en algunos países del mundo y que anteriormente se utilizaban en Colombia.

Así visto, frente a las variables de medición especificadas para la fase de validación dinámica se esperan los siguientes resultados [Tabla 16]:

Tabla 16. Resultados esperados en torno a las variables de medición de la etapa de validación dinámica.

Variable	Escala para utilizar	Resultado esperado
Experiencia de usuario	Calificación global	Calificación promedio de mínimo 6.5. La calificación

Variable	Escala para utilizar	Resultado esperado
		promedio consistirá en el promedio de la calificación global de la variable de satisfacción de usuario y la calificación global de la variable de usabilidad.
Satisfacción del usuario	Calificación global	Calificación promedio de mínimo 6.0. La calificación consistirá en la calificación de la escala Likert de esta variable.
	Escala Likert	Calificación promedio de mínimo 6.0.
Usabilidad	Calificación global	Calificación promedio de mínimo 6.0. La calificación promedio consistirá en el promedio entre la escala Likert y el Net Promoter Score de esta variable junto con la medida otorgada por el System Usability Scale.
	Net Promoter Score	Porcentaje promedio de recomendación/promoción igual o superior al 50%.
	Escala Likert	Calificación promedio de mínimo 6.0, entre las 6 variables que componen a la usabilidad.
	System Usability Scale (SUS)	<p>Calificación promedio de 6.0. Considerando que la escala SUS hace uso de una escala Likert de 0 – 5, la calificación promedio final consistirá en:</p> $C_F = C_O \cdot 2$ <p>Donde C_F es la calificación promedio final y C_O es la calificación promedio obtenida mediante la escala SUS.</p>

Variable	Escala para utilizar	Resultado esperado
Funcionalidad	Escala Likert	Calificación promedio de mínimo 6.0.
Diseño visual	Escala Likert	Calificación promedio de mínimo 6.0.
Accesibilidad	Escala Likert	Calificación promedio de mínimo 6.0.
Tasa de error	Escala Likert	Calificación promedio de mínimo 6.0.
Facilidad de aprendizaje	Escala Likert	Calificación promedio de mínimo 6.0.
Eficiencia de uso	Escala Likert	Calificación promedio de mínimo 6.0.

Fuente: Elaboración propia.

En general, el resultado esperado de la experiencia de usuario, la cual es la variable de medición “global” del proyecto busca alcanzar o superar un umbral del 65% de aceptación por parte de los usuarios frente al aplicativo. Lo anterior indicando que los usuarios finales (población objetivo) perciben una experiencia de usuario promedio de 6.5 sobre 10, lo que se interpretaría como una percepción de experiencia de usuario aceptable en función de las categorías presentadas previamente [Tabla 15]. Este resultado final (la calificación final) corresponde al promedio entre la calificación de la experiencia de usuario obtenida en Fundamentos de Programación y Programación 1, considerando que todas las variables previamente mencionadas se medirían independientemente para cada uno de dichos grupos.

5.3 RESULTADOS PARCIALES (PROYECTO DE GRADO 1)

5.3.1 Resultados parciales del proyecto correspondientes al periodo académico 2021-02, obtenidos en función de los objetivos planteados

Con base en los objetivos planteados, los resultados parciales obtenidos a la fecha y como parte de la completitud del proyecto requerida como parte del proceso de desarrollo realizado en el marco de la asignatura de Proyecto de Grado 1 (periodo académico 2021-02) son:

1. Objetivo general: A la fecha se desarrolló el 80% del componente web del proyecto, que conforma el núcleo básico de la herramienta de enseñanza en línea, donde a su vez está debidamente desplegado y funcional el bot conversacional que realiza la labor de “tutoría virtual” para con los estudiantes, siendo este el mecanismo de inteligencia artificial plantado. A su vez, el componente web está en capacidad de hacer uso de videos

interactivos que conforman la explicación temática a modo de sesión de clase asincrónica sobre la cual trabaja el bot conversacional como tutor.

2. Objetivo específico 1: Como parte del proceso de identificación de la problemática, el contexto y los elementos que componen el MBPSC, se realizaron 2 iteraciones de estudio sobre la población objetivo. La primera de estas iteraciones se realizó durante el periodo 2021-01 a los estudiantes del curso de Programación 1, los cuales, estando ya finalizando el curso, estaban en capacidad de dar su perspectiva en torno a las asignaturas de Programación 1 (recién cursada) y Fundamentos de Programación (cursada un semestre atrás). A su vez, los estudiantes de este grupo podían otorgar su percepción personal en torno a la enseñanza y aprendizaje de dichas asignaturas, y finalmente podían proponer una comparación entre las dos o una visión sobre las dos de forma conjunta. Por otro lado, en el periodo 2021-02 se realizó la segunda iteración del estudio teniendo como enfoque a los estudiantes de nuevo ingreso (es decir aquellos que estaban comenzando a cursar Fundamentos de Programación). Esta iteración estuvo destinada a identificar las particularidades de la población de los estudiantes que conforman el grupo de primer semestre académico del programa de Ingeniería de Sistemas, al igual que la percepción de los estudiantes en torno al curso de Fundamentos de Programación y la proyección académica que tienen cada uno. Finalmente, esta segunda iteración también provee información en torno a las características, particularidades, conocimientos o experiencias previas con los cuales los estudiantes de nuevo ingreso llegan para poder enfrentarse al programa académico.
3. Objetivo específico 2: Se definió el diseño lógico, arquitectural y gráfico del aplicativo buscando ofrecer una experiencia satisfactoria en términos de experiencia de usuario e interfaz de usuario. En cuanto a el diseño lógico y arquitectural se utilizó el estándar UML, y los estilos arquitecturales de (1) "Clean Architecture" (particularmente una variación sobre la arquitectura hexagonal, también conocida como el patrón de puertos y adaptadores), y (2) una aproximación más tradicional por medio de la arquitectura por capas. Por su parte, en torno a la experiencia de usuario e interfaz gráfica se optó por ser coherentes con la metodología planteada y hacer una aproximación Lean para abordar el diseño de la experiencia de usuario y de la interfaz gráfica. Lo anterior se hace mediante la iteración de los prototipos funcionales (tanto en sus componentes como en su totalidad) para poder validar estos dos elementos. A su vez este acercamiento es coherente con el modelo en espiral y con el ciclo de vida iterativo e incremental planteados en la metodología [Sección 2.6.4].
4. Objetivo específico 3: Se construyó el componente web del aplicativo, haciendo uso de los videos interactivos como enfoque para la gamificación de la experiencia de aprendizaje, donde, igualmente, es en este componente que se despliega el bot conversacional desarrollado utilizando el servicio web proveído por la empresa SAP (SAP Conversational AI), que hace la labor de tutor virtual. Estos dos elementos igualmente son parte del 70% acordado

con el director de proyecto que se debe de entregar al finalizar la asignatura de Proyecto de Grado 1.

5. Objetivo específico 4: Se realizó la primera iteración de las pruebas de validación con la población objetivo (específicamente los estudiantes) frente a los videos interactivos y el bot conversacional. Para estas pruebas se desplegó una versión “beta” del aplicativo que ya implementara las funcionalidades previamente mencionadas, y que, a su vez, permitiera validar en una primera instancia la satisfacción y la sensación experiencial percibida frente a la utilización de estos dos elementos, para lo cual se utilizó un cuestionario para la recopilación de las diferentes “respuestas” posibilitando (y facilitando) su posterior análisis.

5.4 RESULTADOS OBTENIDOS

5.4.1 Primera validación frente a la población objetivo

Conforme lo especifica el cronograma de actividades, y a su vez como parte del planteamiento iterativo propio de la metodología de desarrollo escogida [Sección 2.6], se realizó una primera iteración de validación dinámica, por medio de una versión “beta” del artefacto, con el objetivo de recibir una realimentación por parte de los estudiantes en torno a el artefacto, con la cual se pudieran realizar ajustes al planteamiento de este en torno a las funcionalidades que ofrece y las mecánicas de uso que maneja. A su vez, la validación también tenía como meta la obtención un primer dataset mediante el cual se pudiera diseñar, construir y probar el sistema de recomendación propuesto. Dicho dataset consta de los logs resultantes del uso del bot conversacional frente al cual se enfrentaron los estudiantes durante la prueba llevada a cabo.

La validación se realizó en coherencia con la variable de medición planteada para el presente proyecto, por lo cual se buscó validar la experiencia de usuario en torno a el artefacto facilitado a los estudiantes.

Los resultados generales obtenidos de esta validación inicial son los siguientes:

1. En cuanto a la experiencia de usuario se tiene que en promedio una calificación de 6,9 sobre 10 bajo una escala Likert.
2. Particularmente el 26,2% de los estudiantes percibe una experiencia de usuario con una satisfacción del 70% frente a la versión beta, mientras que el 38,1% percibe una experiencia de usuario con una satisfacción igual o superior al 80%.
3. De forma general, el 64,3% de los estudiantes percibe una experiencia de usuario con una satisfacción igual o superior al 70%.

Frente al bot conversacional, se utilizaron sus archivos de log, y las métricas de uso que se pueden extraer de la plataforma de SAP Conversational AI (desde la cual se desarrolla el bot propiamente), para medir el “funcionamiento” y la “eficiencia” de este durante esta primera experiencia de utilización por parte de la población objetivo. Del análisis sobre estos dos elementos se obtienen los siguientes resultados:

1. Se obtuvo un 60% de coincidencia del bot conversacional frente a los Intents.
2. Se obtuvo un 54% de coincidencia del bot conversacional frente a las Entities.
3. Se obtuvo un 52% de asertividad del bot en lo que se refiere a las respuestas otorgadas frente a las preguntas obtenidas, lo que quiere decir que el 48% de las salidas retronadas por el bot conversacional eran correctas para las entradas de los usuarios.

5.4.2 Segunda validación frente a la población objetivo (resultados finales)

Para la segunda iteración de validación frente a la población objetivo, y nuevamente en concordancia con lo especificado en el cronograma de actividades y la metodología de desarrollo escogida [Sección 2.6], se realizó un ejercicio, sincrónico y presencial con los estudiantes de los grupos de Fundamentos de Programación y Programación 1, de la jornada diurna. Este ejercicio constituyó la etapa de validación dinámica del ciclo de transferencia de la tecnología definido por Gorschek, Wohlin, Garre y Larsson [109], que fue adoptado como modelo guía para los procesos de evaluación y validación para los trabajos de grado en el programa de Ingeniería de Sistemas de la Universidad El Bosque. Así visto, la validación por parte de la población objetivo consistió en dos partes: (1) una prueba del aplicativo desarrollado, no supervisado, teniendo solamente como guía al manual de usuario que se puede observar en el anexo 7 [Sección 11.7], y (2) la valoración (calificación) del aplicativo por medio de un cuestionario que constituye el instrumento de recolección de datos escogido para concluir con la etapa de validación dinámica en particular, y con la validación del aplicativo en general.

5.4.2.1 Cuestionario

Buscando que todo el ejercicio de validación se realizara en coherencia con la variable de medición planteada para el presente proyecto, y a su vez con las técnicas propias de la investigación [110], se construyó un cuestionario que le permitiera a los estudiantes evaluar el aplicativo en términos de la experiencia de usuario, estando está representada a través de las subvariables descritas anteriormente [Sección 2.5]. El cuestionario construido posee las siguientes características:

- Realiza una introducción textual al proyecto, al propio cuestionario, y al ejercicio de validación que se materializa en el cuestionario.

- Aclara la protección y privacidad de los datos personales de los participantes, y de las respuestas provistas por estos, mencionando 6 normas, estándares o legislaciones de carácter nacional o internacionales sobre el manejo de la información, la protección de la información personal y el derecho al *habeas data*.
- Posee 5 preguntas para la recolección de datos de carácter demográfico.
- Posee 22 preguntas enfocadas a la medición de la experiencia de usuario de la población objetivo frente al aplicativo desarrollado.
- Pregunta de aceptación de participación voluntaria y no coaccionada en el ejercicio (estudio).

La distribución y caracterización de preguntas enfocadas a la medición de la experiencia de usuario se especifica en la siguiente tabla [Tabla 17]:

Tabla 17. Distribución de las preguntas con relación a las subvariables de medición.

No°	Variable	Subvariable	Escala	Preguntas
1	Usabilidad	N/A	SUS	1 – 10
2		N/A	NPS	21 y 22
3		Funcionalidad	Likert	11
4		Diseño visual	Likert	12 y 13
5		Accesibilidad	Likert	14 y 15
6		Tasa de error	Likert	16
7		Facilidad de aprendizaje	Likert	17
8		Eficiencia de uso	Likert	18
9	Satisfacción	N/A	Likert	19 y 20

Fuente: Elaboración propia.

El cuestionario construido se puede observar en el anexo 8 [Sección 11.8].

Específicamente, las 22 preguntas enfocadas a la medición de la experiencia de usuario realizadas son las siguientes:

Tabla 18. Preguntas del cuestionario.

No°	Pregunta	Escala
1	Creo que me gustaría usar el sistema RetoñosApp con más frecuencia en el desarrollo del curso de Fundamentos de Programación	SUS
2	Encontré que el sistema RetoñosApp es innecesariamente complejo para ser utilizado sin previa instrucción	SUS
3	Pensé que el sistema RetoñosApp era más fácil de utilizar de lo que realmente fue	SUS
4	Creo que necesitaría el apoyo técnico de alguien para poder utilizar el sistema RetoñosApp	SUS

No°	Pregunta	Escala
5	Encontré que las diversas funcionalidades del sistema RetoñosApp estaban bien integradas/desarrolladas conforme a lo que esperaba que debiesen de hacer	SUS
6	Pensé que había demasiadas inconsistencias, incoherencias y contradicciones en el sistema RetoñosApp en cuanto a las funcionalidades que esperaba que el sistema tuviera con relación al propósito de este	SUS
7	Me imagino que la mayoría de la gente aprendería a utilizar el sistema RetoñosApp muy rápidamente	SUS
8	Encontré que el sistema RetoñosApp era muy complicado de utilizar	SUS
9	Me sentí muy confiado y/o cómodo al momento de utilizar el sistema RetoñosApp	SUS
10	Necesitaba aprender muchas cosas antes de poder utilizar el sistema RetoñosApp	SUS
11	¿Qué tan precisas, coherentes y pertinentes considera que fueron las funcionalidades que el aplicativo le proveyó frente a las funcionalidades que usted esperaba teniendo él cuenta el objetivo del sistema descrito al inicio (inicio del cuestionario y de la sesión)?	Likert
12	¿Qué tan coherente cree que es el estilo gráfico (colores, formas, imágenes y elementos de la interfaz gráfica de usuario) de la aplicación con respecto a la imagen institucional de la Universidad El Bosque?	Likert
13	¿Qué tan agradable cree que es el aplicativo en cuanto a su interfaz gráfica de usuario?	Likert
14	¿Qué tan complicado cree que fue para usted utilizar el aplicativo?	Likert
15	¿Qué tan sencillo de utilizar cree que puede ser el aplicativo para el público general?	Likert
16	¿Qué tan eficiente considera que el aplicativo fue en cuanto a la cantidad de errores que presentó?	Likert
17	¿Qué tan sencillo de aprender a utilizar cree que es el aplicativo?	Likert
18	¿Qué tan eficiente considera que fue el aplicativo durante la prueba que acaba de finalizar?	Likert
19	¿Qué tan satisfecho se siente luego de haber utilizado el aplicativo?	Likert
20	¿Qué tan satisfecho cree que se sentiría en el largo plazo, si el aplicativo fuera implementado como un recurso más dentro de los ejercicios de enseñanza y aprendizaje de la asignatura de Fundamentos de Programación? (Lo anterior implica que se	Likert

No°	Pregunta	Escala
	tendrían creados módulos para abarcar en su totalidad las diferentes temáticas y sesiones de clase de la asignatura)	
21	¿Qué tanto recomendaría a sus compañeros o amigos utilizar el aplicativo en caso de que existieran módulos para sus respectivos intereses/carreras/asignaturas/temáticas?	NPS
22	¿Qué tanto recomendaría a su profesor utilizar el aplicativo a modo de herramienta complementaria de la metodología propia de la asignatura de Fundamentos de Programación, estando su uso enfocado en brindar un espacio de estudio autónomo/independiente?	NPS

Fuente: Elaboración propia.

5.4.2.2 Resultados obtenidos en la segunda validación frente a la población objetivo

5.4.2.2.1 Caracterización de la población objetivo

La caracterización demográfica general de los estudiantes que participaron en el estudio se puede evidenciar en la siguiente tabla [Tabla 19]:

Tabla 19. Caracterización demográfica general.

Cantidad de respuestas posibles	188
Cantidad de respuestas obtenidas	136
Cantidad de estudiantes de Fundamentos de Programación	141
Cantidad de estudiantes de Programación 1	46
Porcentaje de respuestas obtenidas frente a la cantidad de respuestas posibles	72,34%
Promedio de edad	18 años

Fuente: Elaboración propia.

Las caracterizaciones demográficas específicas de los estudiantes que participaron en el estudio, discriminando las dos asignaturas, se puede evidenciar en las siguientes dos tablas [Tabla 20], [Tabla 21]:

Tabla 20. Caracterización demográfica de los estudiantes del curso de Fundamentos de Programación.

Cantidad de respuestas posibles	142
Cantidad de respuestas obtenidas	97
Porcentaje de respuestas obtenidas frente a la cantidad de respuestas posibles	68,30%

Promedio de edad	18 años
Cantidad de estudiantes por semestre	Primer semestre: 96 Segundo semestre: 1
Cantidad de estudiantes por grupo	Grupo 01D: 18 Grupo 02D: 33 Grupo 03D: 15 Grupo 04D: 31

Fuente: Elaboración propia.

Tabla 21. Caracterización demográfica de los estudiantes del curso de Programación 1.

Cantidad de respuestas posibles	46
Cantidad de respuestas obtenidas	39
Porcentaje de respuestas obtenidas frente a la cantidad de respuestas posibles	84,78%
Promedio de edad	19 años
Cantidad de estudiantes por semestre	Segundo semestre: 35 Tercer semestre: 3 Cuarto semestre: 1
Cantidad de estudiantes por grupo	Grupo 01D: 20 Grupo 02D: 19

Fuente: Elaboración propia.

Cabe aclarar que las tres últimas tablas [Tabla 19]-[Tabla 21] describen demográficamente solo a los estudiantes participantes (como se aclara previamente), entendiéndose participantes como aquellos estudiantes que completaron satisfactoriamente el cuestionario de evaluación construido [Sección 5.4.2.1], por lo cual la caracterización corresponde al 72,34% de los estudiantes.

5.4.2.2.2 Resultados finales

Luego de la finalización de la segunda iteración de validación, y teniendo como insumo las respuestas de los 136 estudiantes participantes, se procedió a extraer los resultados finales en términos de la variable de medición. Para lo anterior se utilizaron los cálculos que ofrece la estadística descriptiva, buscando la congruencia con lo descrito en la sección de resultados esperados [Sección 5.2]. Así visto, inicialmente se procedió a calcular los resultados finales de los dos grupos por separado, siguiendo las indicaciones especificadas en la tabla de resultados esperados [Tabla 16], para luego realizar un cálculo general con base en la calificación global de la usabilidad, la satisfacción de usuario y la experiencia de usuario obtenidas de los aquellos dos cálculos independientes.

En general, las respuestas de cada estudiante que completó satisfactoriamente el cuestionario de evaluación (durante el proceso de validación dinámica) se pueden encontrar registradas en su respectiva hoja de cálculo en el anexo 12 [Sección 11.12], en conjunto con los resultados finales calculados con base en esta. Particularmente, en esta sección se resumen los resultados obtenidos de dicha evaluación, es decir, los resultados finales calculados.

Los resultados finales obtenidos se evidencian en las siguientes tablas [Tabla 22], [Tabla 23]:

Tabla 22. Resultados finales discriminando los grupos de Fundamentos de Programación y Programación 1.

Grupo	Variable	Calificación	Diferencia de la calificación frente al valor mínimo esperado
Fundamentos de Programación	Usabilidad	8,37	2,37
	Satisfacción del usuario	8,48	2,48
	Experiencia de usuario	8,42	1,92
Programación 1	Usabilidad	7,92	1,92
	Satisfacción del usuario	7,84	1,84
	Experiencia de usuario	7,88	1,38

Fuente: Elaboración propia.

Tabla 23. Resultados finales globales.

Variable	Calificación	Diferencia de la calificación frente al valor mínimo esperado
Usabilidad	8,15	2,14
Satisfacción del usuario	8,16	2,16
Experiencia de usuario	8,16	1,65

Fuente: Elaboración propia.

El promedio aritmético e las 22 preguntas descritas previamente [Tabla 17], y cuyos resultados puntuales (promedio de cada pregunta) se relacionan en la siguiente tabla [Tabla 18]:

Tabla 24. Promedio de las respuestas del cuestionario por grupo.

No° de pregunta \ Grupo	Fundamentos de Programación	Programación 1
1	8,20	7,02
2	7,25	6,92
3	8,06	8,51
4	7,81	7,58

No° de pregunta \ Grupo	Fundamentos de Programación	Programación 1
5	8,45	7,89
6	7,25	6,05
7	8,92	8,76
8	8,63	8,25
9	8,47	8,30
10	8,10	7,69
11	8,43	7,89
12	8,65	8,51
13	8,44	8,25
14	8,11	8,64
15	8,36	8,43
16	8,29	7,38
17	8,77	8,71
18	8,59	8,05
19	8,46	8,15
20	8,50	7,53
21	8,57 (85,7%)	7,79 (77,9%)
22	8,48 (84,4%)	7,87 (78,7%)

Fuente: Elaboración propia.

Puntualmente, y en coherencia con lo descrito en los resultados esperados [Tabla 16], el cálculo de los resultados se realizó de la siguiente manera [Tabla 25]:

Tabla 25. Cálculo de los resultados.

Variable	Cálculo
Usabilidad	$U = \text{Promedio}(P_L + P_{SUS} + P_{NPS})$ <p>Donde, P_L corresponde al promedio de las preguntas 11-18, P_{SUS} corresponde al promedio de las preguntas 1-10, y P_{NPS} corresponde al promedio de las preguntas 21 y 22.</p>
Satisfacción del usuario	$S = \text{Promedio}(P19 + P20)$ <p>Donde $P19$ y $P20$ son la pregunta 19 y 20 respectivamente.</p>
Experiencia de usuario	$UX = \text{Promedio}(U + S)$ <p>Donde U y S corresponden al resultado de la usabilidad y de la satisfacción del cliente respectivamente.</p>

Fuente: Elaboración propia.

Luego, para hallar la diferencia entre la calificación (resultado) obtenido y el resultado esperado, simplemente se le restó a cada calificación final el valor especificado en la sección de resultados esperados [Tabla 16].

Finalmente, y de forma particular para el Net Promoter Score, su promedio se multiplicó por 10 ($NPS \times 10$) para obtener el valor de esta escala en porcentaje. Teniendo en cuenta que el Net Promoter Score busca establecer la probabilidad de que una persona recomiende a sus compañeros utilizar el aplicativo como herramienta de estudio, o a sus docentes como herramienta de apoyo para la asignatura (como parte de la metodología), es lógico presentar dicho valor de forma porcentual.

5.4.2.2.1 Artefacto

Como evidencia de la consecución (construcción y finalización) del artefacto se tienen los siguientes anexos: (1) repositorio de código y video de funcionamiento del artefacto [Sección 11.9], (2) manual de usuario [Sección 11.7], (3) manual técnico [Sección 11.10], y (4) documento de arquitectura de software (SAD) [Sección 11.6].

5.5 VALIDACIÓN EN LA ACADEMIA Y VALIDACIÓN ESTÁTICA

Los resultados de las pruebas realizadas en el marco de la validación en la academia y la validación estática se encuentran expuestos y detallados en el documento correspondiente al plan de pruebas ubicado en el anexo 13 [Sección 11.13], donde igualmente se puede encontrar un complemento al proceso de validación dinámica (validación frente a la población objetivo).

6. ANÁLISIS DE RESULTADOS / DISCUSIÓN

Para el proyecto RetoñosApp se realizaron dos momentos de validación (dos iteraciones de validación). La primera de estas correspondió a una versión beta del aplicativo que principalmente buscaba dos cosas: (1) obtener una realimentación que permitiera ajustar el aplicativo en función de las consideraciones y críticas de la población objetivo, buscando que este se ajustara realmente a su contexto y necesidades, y no solamente a lo que el equipo de desarrollo planteara o el MBPSC vislumbrara, y (2) obtener un primer dataset de logs que funcionara como base para el diseño, construcción y prueba del sistema de recomendación. Por su parte, la segunda correspondió a la validación final del aplicativo, siendo una etapa “más integral” en cuanto a que no solo se validó el artefacto frente a la población objetivo, sino también desde la perspectiva que ofrece la literatura, los estándares y técnicas de la industria del software, la gestión de proyectos y la Ingeniería de Sistemas. Así entonces, esta segunda iteración estaba dividida en tres sub-fases que permitieron validar el artefacto desde diferentes perspectivas y por ende evaluar diferentes características de este: fase de validación en la academia, fase de validación estática y fase de validación dinámica.

Los resultados obtenidos durante los diferentes momentos de validación (las distintas iteraciones de validación) realizadas para el aplicativo RetoñosApp se evidencian en los siguientes dos apartados:

6.1 ANÁLISIS DE RESULTADOS CORRESPONDIENTES A LA PRIMERA ITERACIÓN DE VALIDACIÓN

Los resultados obtenidos en general, arrojan que en promedio los estudiantes perciben una experiencia de usuario con una satisfacción del 69% frente a la versión beta del artefacto facilitada, donde particularmente el 64.3% de los estudiantes percibe una satisfacción en relación con su experiencia de usuario igual o superior al 70%. Lo anterior quiere decir que, aunque en promedio la satisfacción percibida efectivamente supera el umbral del 60% planteado como punto mínimo de satisfacción de usuario que se debe de alcanzar, igualmente es importante resaltar que, de ese promedio, casi el 65% (más de la mitad de los estudiantes) de los estudiantes percibe una satisfacción que llega a ser del 70% o más.

En cuanto al funcionamiento y la eficiencia del bot conversacional, el modelo para el instante de la prueba tuvo en general una eficiencia del 55%, en relación con su nivel de coincidencia frente a Intents y Entities, al igual que a su asertividad frente a las respuestas que retornaba en función de las entradas (preguntas) que los usuarios le generaban. Para un primer ensayo, y un primer momento de verificación el modelo desarrollado era considerablemente efectivo, principalmente por dos razones:

1. A diferencia de la algorítmica tradicional, la inteligencia artificial (IA) no busca ser 100% asertivo en cuanto a los resultados de sus procesos, especialmente por factores probabilísticos, por el hecho de que aprende de su contexto, ambiente, entorno y sus usuarios (donde todos son elementos altamente maleables y cambiantes) y por el hecho de que siempre hay un grado de incertidumbre que afecta a los resultados/respuestas. Lo anterior se traduce en que, dependiendo de los objetivos, del campo de aplicación y de los recursos computacionales disponibles, los modelos aceptables para ser implementados han de tener como mínimo una asertividad del 70%.
2. La prueba no solo permitió medir la funcionalidad y eficiencia del bot conversacional en un instante dado, sino que supuso un entorno para alimentar el modelo, lo que tanto en el corto como en el largo plazo lo hace aún más preciso. Por lo anterior, el hecho de que en este momento de validación el modelo tenga apenas una precisión del 55%, no quiere decir que esa es su precisión final, sino que el haberlo sometido a prueba permite que para futuros eventos de validación (especialmente la validación final), tenga mayor efectividad.

6.2 ANÁLISIS DE RESULTADOS CORRESPONDIENTES A LA SEGUNDA ITERACIÓN DE VALIDACIÓN (RESULTADOS FINALES)

Habiendo realizado la recolección de los datos, durante la fase de validación en la academia, y calculado los resultados con base en estos, se procede a realizar el análisis final de los resultados del proyecto.

El proceso de validación dinámica, es decir, de validación con la población se llevó a cabo el día 22 de abril de 2022, con los estudiantes de los grupos de la jornada diurna de los cursos de Fundamentos de Programación y Programación 1 del programa de Ingeniería de Sistemas de la Universidad El Bosque, durante sus respectivas sesiones de clase:

- Fundamentos de Programación: 9:00 – 11:00.
- Programación 1: 11:00 – 13:00.

El ejercicio realizado tuvo una duración de entre 45 minutos a 1 hora con 20 minutos aproximadamente, dependiendo del grupo, puesto que cada grupo tenía una cantidad distinta de alumnos (lo que suponía una cantidad variable de preguntas por parte de los estudiantes), al igual que la disponibilidad de tiempo otorgada por el docente era distinta entre los diferentes grupos dependiendo de la actividad que tuviera programada para ese día en su planificación curricular.

El ejercicio de validación tuvo 4 fases (momentos):

1. Una presentación del grupo equipo de desarrollo y del proyecto RetoñosApp, seguida de una explicación de la actividad que se va a realizar, sus objetivos, y su importancia para con el proyecto en cuestión
2. La fase de prueba, libre y no supervisada, del aplicativo desarrollado, solamente teniendo como guía el manual de usuario que se puede observar en el anexo 7 [Sección 11.7]. Esta fase de prueba tuvo una duración de entre 20 – 25 minutos dependiendo del tamaño de los grupos y el tiempo disponible.
3. La evaluación que se realizó mediante un cuestionario [Sección 5.4.2.1] desarrollado y distribuido a través de Google Forms, que manejando casi en su totalidad preguntas cerradas (a excepción de dos preguntas de tipo demográficas) les permitían a los estudiantes valorar diferentes aspectos del aplicativo, pero todo con relación a la experiencia de usuario.
4. Los agradecimientos por el espacio facilitado, y la colaboración y participación de los estudiantes, y los comentarios de cierre.

6.2.1 Análisis de resultados

El grupo poblacional objetivo (los estudiantes de los grupos de la jornada diurna de los cursos de Fundamentos de Programación y Programación 1 del programa de Ingeniería de Sistemas de la Universidad El Bosque) estaba compuesta por 188 estudiantes (142 estudiantes de Fundamentos de Programación y 46 estudiantes de Programación 1), donde solamente completaron el cuestionario 136 estudiantes (97 estudiantes de Fundamentos de Programación y 39 estudiantes de Programación 1), lo cual representa al 72,34% de la población total. Considerando que la participación era voluntaria por parte de los estudiantes, no se pudo obtener una participación del 100%, pero si se superó el umbral del 70%, el cual igualmente es relevante. Lo anterior igualmente quiere decir que, y siendo coherentes con lo expuesto por Sampieri [110], la muestra correspondió a una muestra de voluntarios.

De ese 72,34% de los estudiantes que completaron el cuestionario, se tienen las siguientes características generales:

- El promedio de edad ronda los 18 años.
- El 71,32% de los estudiantes pertenecen al grupo de Fundamentos de Programación, mientras que el 28,67% restante pertenecen al grupo de Programación 1.
- El 70,58% de los estudiantes está cursando primer semestre, el 26,47% segundo semestre, el 2,20% tercer semestre, y el 0,73% cuarto semestre.
- El 16,17% de los estudiantes son de sexo femenino, por lo cual el 83,82% son de sexo masculino.

En lo referente a la variable de medición global, los estudiantes de Fundamentos de Programación reconocieron haber percibido una experiencia de usuario de 8,42 sobre 10, mientras que los estudiantes de Programación 1 percibieron una experiencia de usuario de 7,88 sobre 10, ubicando dichos resultados, según lo especificado en la tabla de categorías para el análisis de los resultados [Tabla 15], una experiencia de usuario sobresaliente en ambos casos. Igualmente, en el caso de los estudiantes de Fundamentos de Programación, el resultado estuvo 1,92 puntos por arriba del resultado esperado para la variable global, mientras que el resultado para el grupo de Programación 1 estuvo 1,38 puntos por arriba.

Igualmente es importante recordar que para el caso de este proyecto la experiencia de usuario se definió como la sinergia entre la usabilidad del aplicativo y la satisfacción del usuario frente a este, por lo cual, de forma específica, los resultados en torno a estas dos subvariables de medición muestran lo siguiente:

- En lo referente a la usabilidad del aplicativo, los estudiantes de Fundamentos de Programación la evaluaron con un 8,37 sobre 10, mientras que los estudiantes de Programación 1 le dieron una calificación de 7,92 sobre 10, donde nuevamente, estos resultados se ubicarían en un sobresaliente [Tabla 15]. Igualmente, el resultado de la usabilidad en el caso del grupo de Fundamentos de Programación estuvo 2,37 puntos por arriba del resultado esperado, mientras que en el caso del grupo de Programación 1 estuvo 1,92 puntos por arriba.
- En lo referente a la satisfacción del usuario frente al aplicativo, los estudiantes de Fundamentos de Programación la evaluaron con un 8,48 sobre 10, mientras que los estudiantes de Programación 1 le dieron una calificación de 7,84 sobre 10, donde nuevamente, estos resultados se ubicarían en un sobresaliente [Tabla 15]. Igualmente, el resultado de la satisfacción del usuario en el caso del grupo de Fundamentos de Programación estuvo 2,48 puntos por arriba del resultado esperado, mientras que en el caso del grupo de Programación 1 estuvo 1,84 puntos por arriba.

Con base en lo anterior, la calificación final general (comprendiendo a los dos grupos) muestra que la población objetivo percibió una experiencia de usuario de 8,16 sobre 10, siendo una experiencia de usuario que se puede catalogar como sobresaliente [Tabla 15], estando a su vez 1,65 puntos por arriba del resultado esperado. Luego, la calificación de la usabilidad fue de 8,15, mientras que la calificación de la satisfacción del usuario fue de 8,16, lo que ubica a ambos resultados, nuevamente, en una valoración sobresaliente, estando 2,14 y 2,16 puntos por arriba de los resultados esperados respectivamente.

En cuanto al Net Promoter Score, que se resalta puesto que busca exponer la probabilidad de recomendación de uso de la aplicación, los estudiantes de Fundamentos de Programación tendrían una probabilidad del 85,30% de

recomendar el aplicativo, mientras que los estudiantes de Programación 1 presentarían una probabilidad del 78,33%. Estas probabilidades están un 35,30% y un 28,33% por arriba de lo esperado. A su vez, y de forma general la población objetivo tendría una probabilidad de recomendación del aplicativo de un 81,81%, estando esta un 31,81% por encima de lo esperado.

En cuanto a las calificaciones finales por pregunta, al relacionar cada uno con su categoría correspondiente [Tabla 15], se obtendría lo siguiente:

Tabla 26. Categorización de las calificaciones obtenidas en cada pregunta del cuestionario.

No° de pregunta \ Grupo	Fundamentos de Programación	Programación 1
1	8,20 (S)	7,02 (S)
2	7,25 (S)	6,92 (S)
3	8,06 (S)	8,51 (S)
4	7,81 (S)	7,58 (S)
5	8,45 (S)	7,89 (S)
6	7,25 (S)	6,05 (S)
7	8,92 (S)	8,76 (S)
8	8,63 (S)	8,25 (S)
9	8,47 (S)	8,30 (S)
10	8,10 (S)	7,69 (S)
11	8,43 (S)	7,89 (S)
12	8,65 (S)	8,51 (S)
13	8,44 (S)	8,25 (S)
14	8,11 (S)	8,64 (S)
15	8,36 (S)	8,43 (S)
16	8,29 (S)	7,38 (S)
17	8,77 (S)	8,71 (S)
18	8,59 (S)	8,05 (S)
19	8,46 (S)	8,15 (S)
20	8,50 (S)	7,53 (S)
21	8,57 (85,7%) (S)	7,79 (77,9%) (S)
22	8,48 (84,4%) (S)	7,87 (78,7%) (S)

Fuente: Elaboración propia.

Donde “D” corresponde a deficiente, “I” corresponde a insuficiente, “A” corresponde a aceptable, “S” corresponde a sobresaliente, y “E” corresponde a excelente.

Finalmente, y a modo de anotación, es interesante anotar que el grupo de Programación 1 califico la experiencia de usuario 0,54 puntos por debajo del grupo de Fundamentos de Programación, lo que en cualquier caso sigue dejando dicha

calificación con una valoración de sobresaliente. A modo de inferencia, se plantea que esta diferencia se debe a dos razones principales:

- El grupo que actualmente está en Programación 1, está constituido casi en su totalidad por los estudiantes que realizaron el primer ejercicio de validación parcial [Sección 6.1] durante el periodo 2021-02, es decir que probaron el aplicativo en su versión beta, lo cual les daba un punto de comparación al momento de probar y evaluar.
- El grupo de Programación 1 ya tiene un recorrido académico mayor, lo que les da más herramientas, criterios y puntos a considerar al momento de probar y evaluar un aplicativo.

7. CONCLUSIONES

Las conclusiones constituyen la síntesis de los puntos más relevantes del proyecto, siempre teniendo presente los resultados de este y su respectivo análisis, al igual que los diferentes hitos y elementos que lo constituyeron. En este apartado se brinda el cierre formal al proyecto y a la investigación, estableciendo la consecución de determinadas metas (objetivos), al igual que definiendo si el proyecto y la investigación finalizó o no satisfactoriamente.

En el caso del proyecto RetoñosApp, las conclusiones se dan desde tres perspectivas diferentes: (1) en la sección 7.1 se establecen las conclusiones con base exclusivamente en los resultados obtenidos y su correspondiente análisis, luego (2) en la sección 7.2 se entregan una serie de conclusiones relacionando los resultados obtenidos (o los logros alcanzados) con los objetivos (tanto general como específicos) del proyecto, y finalmente (3) en la sección 7.3, y en coherencia con la perspectiva y la filosofía académica de la Universidad El Bosque, se desarrollan las conclusiones pero relacionando los resultados obtenidos con el MBPSC planteado en las secciones 1 [Sección 1.2] y 2 [Sección 2.3] del presente documento.

7.1 CONCLUSIONES GENERALES

En la tabla de las categorías para representar, categóricamente, los resultados obtenidos [Tabla 15], se presentan 5 categorías que, por medio de intervalos numéricos, agrupan todos los posibles resultados que se podían obtener en el proyecto (reduciendo la precisión al primer decimal). Considerando que todos los resultados obtenidos en cuanto a las variables de experiencia de usuario, usabilidad del aplicativo y satisfacción del usuario frente al aplicativo estuvieron ubicados en la categoría de “sobresaliente”, tanto al diferenciar a los grupos de Fundamentos de Programación del grupo de Programación 1, como al momento de manejarlos de forma conjunto, inicialmente se puede decir que el proyecto, desde la perspectiva de los estudiantes, fue sobresaliente. Pero, desde su planteamiento, el proyecto buscaba validar o dar respuesta a la siguiente hipótesis de investigación [Sección 2.4]:

“El utilizar una plataforma de enseñanza cuya característica principal es que hace la labor de un “tutor virtual”, beneficia satisfactoriamente tanto los procesos de enseñanza como de aprendizaje de la programación en el aula de clase, al funcionar fundamentalmente como un complemento a las actividades educativas desarrolladas dentro de esta, pero a su vez proveyendo una realimentación frente a los tópicos, en lo referente a los estudiantes, y de progreso de sus alumnos, en lo que se refiere a los docentes”.

Por lo cual, y considerando el tiempo disponible, para poder validar dicha hipótesis se optó por la experiencia de usuario como variable de medición, mediante la cual se podía determinar si los estudiante perciben satisfactoriamente, o no, al aplicativo como una herramienta para apoyar/complementar sus procesos de aprendizaje, y por ende como una herramienta que los docentes pueden aplicar en la asignatura (como parte de la metodología propia del curso) para asistir su ejercicio de enseñanza.

Así visto, y considerando las calificaciones dadas por los estudiantes frente al aplicativo, se puede concluir lo siguiente:

- La población objetivo considera que el aplicativo, que es el producto que materializa el proyecto mismo, es sobresaliente en términos de experiencia de usuario en general, y de la usabilidad de este, y de la satisfacción que les genera como usuario en particular.
- Los estudiantes consideran que la plataforma beneficia satisfactoriamente sus procesos de aprendizaje, de tal forma que es probable, en un 81,81%, que lo recomienden a sus compañeros como parte de sus métodos de estudio, o a sus profesores como parte de las herramientas de apoyo metodológico para el desarrollo del curso.

En general, se puede concluir que el proyecto, desde la perspectiva de la población objetivo, **propicia positivamente el ejercicio educativo que se desarrolla en el aula de clase** (entendiéndose el aula de clase como el espacio educativo sincrónico ya sea virtual o presencial), Esto se resume en que según lo valora la población objetivo, **el aplicativo beneficia satisfactoriamente tanto los procesos de enseñanza como de aprendizaje de la programación en el aula de clase**. Lo anterior a su vez se ve potenciado en que esta plataforma, a diferencia de las otras herramientas disponibles [Sección 3.2], tiene la ventaja de que puede nutrir los procesos de enseñanza y aprendizaje puesto que incorpora el bot conversacional y el sistema de recomendación, proveyendo a través de estos una realimentación frente a los tópicos, en lo referente a los estudiantes, y del progreso de sus alumnos, en lo que se refiere a los docentes.

Conforme a lo anterior, se puede afirmar que la hipótesis de investigación que se ha manejado desde el planteamiento del proyecto es correcta.

7.2 CONCLUSIONES CON BASE EN LOS OBJETIVOS

Teniendo en cuenta los objetivos del proyecto, se pueden relacionar la siguientes conclusiones:

7.2.1 Conclusiones con base en el objetivo general

El objetivo general del proyecto dice lo siguiente:

“Desarrollar una herramienta de enseñanza en línea, mediada por inteligencia artificial y un sistema de recomendación, que apoye, complemente y permita realimentar, por medio de reportes, la enseñanza y el aprendizaje de los cursos de Fundamentos de Programación y Programación 1 del programa de Ingeniería de Sistemas de la Universidad El Bosque”.

Considerando todos los elementos descritos previamente en el presente documento, al igual que en el anexo 6 [Sección 11.6], y teniendo en cuenta que se dio cumplimiento a los 4 objetivos específicos [Sección 7.2.2], se afirma que se dio cumplimiento al objetivo general del proyecto.

7.2.2 Conclusiones con base en los objetivos específicos

7.2.2.1 Conclusión frente al objetivo específico 1

El primer objetivo específico del proyecto dice lo siguiente:

“Analizar las características del contexto de enseñanza y aprendizaje de las asignaturas Fundamentos de Programación y Programación 1, del programa de Ingeniería de Sistemas de la Universidad El Bosque, con el fin de determinar las necesidades, requerimientos y oportunidades específicas que presenta la población objetivo por medio de un estudio realizado a los estudiantes de los cursos de Fundamentos de Programación y Programación 1 en los periodos 2021-01, 2021-02 y 2022-01 y una entrevista con los docentes de estos cursos (tanto de los horarios diurnos como nocturnos)”.

Frente a este objetivo se puede afirmar que, durante los periodos académicos correspondientes a las asignaturas de Seminario de Investigación, Proyecto de Grado 1 y Proyecto de Grado 2 (2021-01, 2021-02 y 2022-01) se realizaron 4 ejercicios sincrónicos con los estudiantes de Fundamentos de Programación y Programación 1, siendo estos 4 ejercicios sincrónicos los siguientes:

1. La aplicación de un cuestionario, para la identificación de necesidades, requerimientos y oportunidades, a los estudiantes de Programación 1 de la jornada diurna del periodo 2021-01, y la aplicación de una entrevista no estructurada a un docente del curso de Fundamentos de Programación y a un docente de Programación 1, también de la jornada diurna. Ver sección 4 [Sección 4.2.1].
2. La aplicación de un cuestionario, para la identificación de necesidades, requerimientos y oportunidades, a los estudiantes de Fundamentos de Programación de la jornada diurna del periodo 2021-02. Ver sección 4 [Sección 4.2.1].

3. La validación de la versión beta del aplicativo para recibir una realimentación en cuanto a las mecánicas, diseño, funcionalidades, y experiencia de usuario, con los estudiantes de los dos grupos de Fundamentos de Programación de la jornada diurna del periodo 2021-02. Lo anterior con el fin de identificar aspectos nuevos a considerar, acierto obtenidos, aspecto a mejorar, y comprobar si la identificación de requerimientos y el planteamiento tanto conceptual como tecnológico de la herramienta efectivamente se ajustaba a las necesidades y oportunidades de la población objetivo. Ver sección 5 [Sección 5.4.1].
4. La validación final del aplicativo con toda la población objetivo, en donde se procedió a que los estudiantes realizaran una prueba, no supervisada, del aplicativo, en cuanto a sus mecánicas, funcionalidades y capacidades, y luego evaluaran (calificaran) el aplicativo en términos de la variable de medición global, estando está representada a través de sus correspondientes subvariables de medición [Sección 2.6]. Ver sección 5 [Sección 5.4.2].

Con base en lo anterior, tenemos que se cumplió el objetivo específico No° 1 planteado para el proyecto.

7.2.2.2 Conclusión frente al objetivo específico 2

El segundo objetivo específico del proyecto dice lo siguiente:

“Definir el diseño y arquitectura del sistema de información, de tal forma que estos permitan que el aplicativo sea amigable con el usuario tanto estéticamente como en cuanto a la experiencia del usuario (UX) y cuyo diseño sea coherente, en términos de interfaz (UI), con el objetivo general de la plataforma y las funciones que le provee al usuario”.

Frente a este objetivo se puede afirmar que, se planteó la arquitectura y se diseñó el artefacto con el fin de alcanzar altos estándares de calidad frente a diferentes enfoques, lo cual puede evidenciarse por medio del documento de arquitectura de software (SAD) [Sección 11.6]. Igualmente, y considerando la calificación obtenida en la subvariable de usabilidad (sobresaliente), y en la variable de experiencia de usuario (sobresaliente), los estudiantes afirman que su experiencia de usuario fue agradable, al igual que la aplicación tenía un alto grado de usabilidad. Lo anterior quiere decir que los estudiantes percibieron una buena experiencia de usuario, y una buena usabilidad.

Particularmente, y en lo que se refiere a la usabilidad, hay que tener en cuenta que tal como se menciona en el marco teórico [Sección 3.2.4], la usabilidad se define como la facilidad y comodidad en la utilización de una herramienta de software, considerando aspectos tales como la curva de aprendizaje, la protección contra errores, la apariencia estética de la interfaz de usuario y la accesibilidad que

presenta. Esto, se deriva en el concepto de “software amigable”, que consiste en el hecho de que un aplicativo sea estéticamente agradable, y que sea fácil de manejar. Frente a lo anterior, las preguntas 11-18 del cuestionario aplicado, correspondían a la evaluación de las siguientes características del aplicativo: funcionalidad, diseño visual, accesibilidad, tasa de error, facilidad de aprendizaje y eficiencia de uso, donde para todos los casos se tuvo igualmente una valoración de sobresaliente. A su vez, las preguntas 1-10 del cuestionario aplicado, correspondían a la evaluación de las siguientes características del aplicativo: coherencia de las funcionalidades, coherencia entre las funcionalidades, facilidad de aprendizaje, facilidad de uso y confianza frente al sistema, donde nuevamente para todos los casos se tuvo igualmente una valoración de sobresaliente.

Lo anterior se resume en que la población objetivo tuvo una experiencia de usuario satisfactoria, que el aplicativo fue amigable y estéticamente atractivo y coherente, al igual que las funcionalidades eran acordes con el objetivo de la plataforma.

Con base en lo anterior, se afirma que se alcanzó el objetivo específico No° 2 planteado para el proyecto.

7.2.2.3 Conclusión frente al objetivo específico 3

El tercer objetivo específico del proyecto dice lo siguiente:

“Construir un aplicativo web que integre un bot conversacional que haga la tarea de tutor virtual, y un sistema de recomendación alimentado por información sobre los procesos de aprendizaje de los estudiantes generada de manera automatizada, con el fin de apoyar los procesos de enseñanza y aprendizaje de los estudiantes y los docentes dentro y fuera del aula de clase”.

Frente a este objetivo se pueden afirmar 3 cosas:

1. Se desarrolló un aplicativo web que integra videos interactivos, un bot conversacional y un sistema de recomendación.
2. El bot conversacional realiza la labor de tutor virtual, posibilitándole a los estudiantes tener un acompañamiento sincrónico durante su experiencia educativa en la plataforma, puesto que el bot conversacional está en capacidad de recibir entradas (preguntas, consultas sobre conceptos o dudas), y retornar una respuesta que puede ser tanto textual como en forma de rich response (respuesta interactiva).
3. El sistema de recomendación se alimenta por la información proveniente tanto del bot conversacional (las entradas del usuario), como de la plataforma misma (por medio de un cuestionario que se crea para cada “sesión de clase”).

Dado lo anterior, dimos cumplimiento al objetivo específico No° 3 planteado para el proyecto.

7.2.2.4 Conclusión frente al objetivo específico 4

El cuarto objetivo específico del proyecto dice lo siguiente:

“Realizar pruebas de verificación y validación, y con la población objetivo en términos de experiencia del usuario (UX), con el fin de demostrar la utilidad de la plataforma, su nivel de calidad y el grado de satisfacción de los estudiantes y docentes frente al aplicativo”.

Frente a este objetivo se puede afirmar que durante los periodos 2021-02 y 2022-01 se realizaron pruebas de validación con la población objetivo, tanto para una versión beta del aplicativo, como para su versión definitiva respectivamente [Sección 5.4.1], [Sección 5.4.2]. Estas pruebas estuvieron enfocadas en validar el artefacto en función de la variable global de medición, al igual que a través de sus subvariables de medición.

Se logró el objetivo específico No° 4 planteado para el proyecto.

7.3 CONCLUSIONES DESDE LA PERSPECTIVA DEL MBPSC

El presente proyecto se desarrolló con el objetivo de brindar un medio y una herramienta para el apoyo a los ejercicios educativos que se desarrollan en el aula de clase, entendiéndose ejercicios educativos por los procesos de aprendizaje (por parte del estudiante) y los procesos de enseñanza (por parte del docente). Teniendo en cuenta las conclusiones generales [Sección 7.1] y las conclusiones con base en los objetivos del proyecto [Sección 7.2], se puede concluir que la plataforma RetoñosApp, efectivamente realiza una transformación al contexto principalmente por 4 razones:

1. El aplicativo desarrollado entra como un nuevo artefacto que propicia las estrategias de estudio y de enseñanza (según sean estudiantes o docentes respectivamente), puesto que les provee a los estudiantes un medio de estudio dinámico e integral con el cual complementar la enseñanza impartida en el aula, y les proporciona a los docentes una asistencia asincrónica para el curso. Esto a su vez modifica los hábitos de los estudiantes en cuanto a la búsqueda de bibliografía, fuentes y recursos para su estudio independiente, puesto que aparte de utilizar elementos externos, tendrán un instrumento con el cual asistir su aprendizaje, que esté a su vez alineado con la estructura curricular y metodológica del curso.
2. La plataforma, al proveer un reporte tanto para los estudiantes como para los docentes, en torno a los procesos de aprendizaje (personales o grupales

respectivamente), propicia la continuidad académica, y promueve la participación de los estudiantes en su educación, siendo así actores operantes para la construcción de su propio conocimiento. Lo particularmente interesante del reporte es que consiste en la recomendación de un tipo de “ruta de aprendizaje”, en función de una temática particular, de tal forma que se le da al estudiante una guía para reforzar o complementar un tópico de la asignatura en cuestión a través de aquellas temáticas que sean afines o próximas a dicha temática.

3. El aplicativo promueve no solamente el aprendizaje de la programación desde una perspectiva técnica (como normalmente se cree que es), sino también desde una perspectiva lógica, creativa y basada en ejemplos. Lo anterior busca que el estudiante desarrolle la capacidad de programar en general, pero que en particular sea capaz de entender la programación como un camino, mas no como el fin. Al igual que se promueve que los estudiantes sean capaces de apropiarse de los conceptos de la programación como una forma de pensar, abstraer, interpretar y traducir el mundo, los problemas y las necesidades de un contexto, más que como una actividad técnica y netamente procedimental, mecánica y/o estandarizada.

7.4 CONSIDERACIONES FINALES

Los resultados y las conclusiones del presente proyecto pueden ser utilizadas como parte de futuros procesos de investigación y/o desarrollo en torno a la enseñanza en ciencias de la computación en general y en particular en torno a la enseñanza de la programación. A su vez, es importante resaltar que por el contexto mismo en el que se desarrolla este proyecto, sus resultados y conclusiones igualmente pueden ser apoyo e insumo para el desarrollo de futuros proyectos que se enmarquen en el contexto específico del programa de Ingeniería de Sistemas de la Universidad El Bosque (como lo pueden ser, entro otros, futuros trabajos de grado).

8. RECOMENDACIONES Y TRABAJO FUTURO

En el presente trabajo se describe y detalla cada uno de los elementos que componen al proyecto RetoñosApp, al igual que todo el proceso de ideación, planteamiento, desarrollo y validación de este.

En lo referente a las recomendaciones, se identificó que el estudio sobre la población objetivo pudo haber aprovechado dos instrumentos más aparte de las entrevistas realizadas, los cuestionarios aplicados y la investigación documental efectuada: (1) un trabajo de campo bajo el rol de observador que se pudo efectuar mediante inmersión pasiva en una sesión de clase de los diferentes grupos de las asignaturas de Fundamentos de Programación y Programación 1, y (2) de la realización de grupos focales con una muestra de cada grupo de dichas asignaturas que pudiera revelar, colaborativa e interactivamente, nueva información que pudiera haber nutrido la identificación y descripción del contexto, la problemática, la población objetivo, el MBPSC, y la solución planteada. Igualmente, se plantea a modo de recomendación estudiar la posibilidad de realizar una prueba con estudiantes de al menos hasta 3er semestre para poder validar la validez de la tecnología incluso cuando los usuarios tienen una formación y consecuentemente un conocimiento más extenso en un campo particular. Esta última recomendación podría ser de utilidad en caso de que se desee implementar la tecnología en contextos más variados, y que este dirigida para un grupo poblacional más extenso.

Por su parte, en lo que respecta al trabajo futuro se han identificado seis áreas o enfoques iniciales a considerar para la evolución, escalamiento, mejoría, optimización y actualización del proyecto.

1. El sistema de recomendación actualmente está desarrollado como un sistema de recomendación que hace uso exclusivamente de un análisis sintáctico, es decir, del análisis de la estructura de las expresiones y la identificación de las palabras utilizadas. Pero, a modo de mejoría de este, se propone que el sistema de recomendación evolucione hacia un sistema de recomendación que haga uso también del análisis semántico, es decir de aquel que permite identificar el significado de las expresiones más allá de su estructura sintáctica.
2. El sistema de recomendación actualmente está limitado a un sistema de recomendación basado en contenido, es decir, aquellos que recomiendan ítems en función de experiencia (información) pasada, donde se compartan características similares. Pero, a modo de mejoría de este, se propone el extenderlo para que sea un sistema de recomendación por filtrado colaborativo, es decir que las recomendaciones que se le realicen a un estudiante A estén basadas tanto en la información que provee dicho estudiante, así como en la información que proveen otros estudiantes, B, C, D, y demás, que posean un perfil similar con el estudiante A (con

características similares). Incluso, el sistema de recomendación se puede extender a un sistema de carácter híbrido que recoja las virtudes de varios tipos de sistemas de recomendación ya existentes.

3. Se puede aprovechar el conocimiento que el sistema de recomendación otorga sobre cada estudiante para ordenar las asignaturas, temáticas por asignatura, o sesiones de clase por temática, con base en las necesidades que el sistema logre identificar para cada uno. Es decir, que las asignaturas, temáticas y sesiones de clase se le presenten a los estudiantes en el orden de prioridad de estudio/repaso (teniendo primero aquellas asignaturas, temáticas y sesiones que más se le dificulten y las que estén relacionadas a estas).
4. Se puede investigar y estudiar la aplicación de la tecnología desarrollada en otros campos del conocimiento o programas de pregrado, más allá de solo su aplicación en la enseñanza de las Ciencias de la Computación en general o la programación en particular. Este punto puede suponer la necesidad de que se modifiquen, extiendan y/o adapten algunas de las funcionalidades que ya estén desarrolladas, al igual de que se tengan que desarrollar nuevas funcionalidades.
5. La plataforma desarrollada, debido a su diseño y a las mecánicas de uso, puede ser implementada para prácticamente cualquier asignatura, indistintamente del programa o nivel académico, puesto que esta permite y se enfoca en la creación de asignaturas, y consecutivamente de módulos de aprendizaje (representados por temáticas y sesiones de clase). Lo anterior implica que a medida que el proyecto evolucione y se valide iterativamente, este puede ser implementado para los demás programas académicos de la Universidad El Bosque, o incluso se puede estudiar la posibilidad de ampliar su implementación a otros espacios, como por ejemplo otras instituciones educativas (incluyendo colegios).
6. Aunque el tiempo planteado para el proyecto imposibilitaba medir si el aprendizaje y los resultados académicos de los estudiantes tenían alguna mejoría identificable, comparando el antes y el después de la implementación aplicación de la tecnología en el contexto, esta labor se plantea como un elemento que se puede considerar para trabajo futuro. Lo anterior, especialmente considerando que el hecho de validar al aplicativo como una herramienta que efectivamente promueva y propicie el aprendizaje por parte de los estudiantes sería un paso lógico para seguir considerando la razón de ser y el propósito real de la tecnología en particular y del proyecto en general.

9. LECCIONES APRENDIDAS

Las lecciones aprendidas durante el periodo de desarrollo del proyecto giran en torno a 4 aspectos principalmente:

1. El trabajo en equipo.
2. La investigación como uno de los pilares para el desarrollo de un proyecto de ingeniería.
3. El desarrollo de un proyecto teniendo como limitantes y condicionantes a un contexto, problema y grupo poblacional específico.
4. El trabajo académico y documental como respaldo y soporte al trabajo “técnico” que conlleva el desarrollo de un proyecto de ingeniería.

De forma puntual, las lecciones aprendidas se pueden evidenciar de mejor forma a través del documento que comprende las reflexiones periódicas [Sección 11.11], las cuales manejan una estructura en donde se realiza una descripción detallada de cada semana de trabajo, se destacan los hechos relevantes de cada una, se resaltan los aspectos positivos y/o negativos identificados, los aprendizajes obtenidos, y se busca relacionar cada una de las semanas con una frase que “resuma”, de forma inspiracional, los sucesos de dicha semana. Igualmente, se diseñó un formato específico para la realización de una reflexión “general” para cada corte académico, de tal forma que se pudiera también hacer un análisis macroscópico a los eventos ocurridos durante el desarrollo del proyecto y a las enseñanzas que dichos eventos le dejan al equipo de trabajo.

10. REFERENCIAS BIBLIOGRÁFICAS

- [1] International Organization for Standardization, *Norma ISO/IEC 9126: Calidad en la Industria del Software*. .
- [2] Y. Y. Lee, A. Smith, L. Calvert, and E. Snajdr, "Innovative Data-Driven Methods to Improve Digital User Experience," *Qual. Quant. Methods Libr.*, vol. 5, no. June2016, 2016.
- [3] D. D. Hammill, J. E. Leigh, G. McNutt, and S. C. Larsen, "A new definition of learning disabilities," *Learn. Disabil. Q.*, vol. 11, no. 3, 1988, doi: 10.2307/1510766.
- [4] K. E. Stanovich, "The sociopsychometrics of learning disabilities," *J. Learn. Disabil.*, vol. 32, no. 4, 1999, doi: 10.1177/002221949903200408.
- [5] P. L. Holborow and P. S. Berry, "Hyperactivity and learning difficulties.," *J. Learn. Disabil.*, vol. 19, no. 7, 1986, doi: 10.1177/002221948601900713.
- [6] J. Pleasants and J. K. Olson, "What is engineering? Elaborating the nature of engineering for K-12 education," *Sci. Educ.*, vol. 103, no. 1, 2019, doi: 10.1002/sce.21483.
- [7] Universidad El Bosque, Facultad de Ingeniería, and Programa de Ingeniería de Sistemas, "Proyecto Educativo del Programa ," Bogotá, Oct. 2019.
- [8] G. López, M. Posada, C. Cardozo, and D. J. Cuartas, "Specific actions for desertion reduction, competence identification and guidance for new students of an engineering program, a case study," 2010, doi: 10.1109/ICEED.2010.5940784.
- [9] A. Salazar, J. Gosálbez, I. Bosch, R. Miralles, and L. Vergara, "A case study of knowledge discovery on academic achievement, student desertion and student retention," 2004, doi: 10.1109/itre.2004.1393665.
- [10] D. Chinn, A. Carbone, J. Sheard, and M. J. Laakso, "Study habits of CS 1 students: What do they do outside the classroom?," *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 103, 2010.
- [11] J. Sheard, A. Carbone, D. Chinn, and M. J. Laakso, "Study habits of CS 1 students: What do they say they do?," 2013, doi: 10.1109/LaTiCE.2013.46.
- [12] M. Romero, A. Lepage, and B. Lille, "Computational thinking development through creative programming in higher education," *Int. J. Educ. Technol. High. Educ.*, vol. 14, no. 1, 2017, doi: 10.1186/s41239-017-0080-z.
- [13] T. C. Hsu, S. C. Chang, and Y. T. Hung, "How to learn and how to teach computational thinking: Suggestions based on a review of the literature," *Comput. Educ.*, vol. 126, 2018, doi: 10.1016/j.compedu.2018.07.004.
- [14] Redacción Educación, "En Colombia faltan 80.000 ingenieros informáticos," *El Tiempo*, Bogotá, Nov. 24, 2020.
- [15] J. Figueiredo and F. J. García-Péalvo, "Teaching and learning strategies of programming for university courses," 2019, doi: 10.1145/3362789.3362926.
- [16] S. Xinogalos, "Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy," *Educ. Inf. Technol.*, vol. 21, no. 3, 2016, doi: 10.1007/s10639-014-9341-9.

- [17] N. R. Boyer, S. Langevin, and A. Gaspar, "Self direction and constructivism in programming education," 2008, doi: 10.1145/1414558.1414585.
- [18] L. Malmi, J. Sheard, P. Kinnunen, Simon, and J. Sinclair, "Theories and Models of Emotions, Attitudes, and Self-Efficacy in the Context of Programming Education," 2020, doi: 10.1145/3372782.3406279.
- [19] Y. Bosse and M. A. Gerosa, "Difficulties of Programming Learning from the Point of View of Students and Instructors," *IEEE Lat. Am. Trans.*, vol. 15, no. 11, 2017, doi: 10.1109/TLA.2017.8070426.
- [20] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *ACM SIGCSE Bull.*, vol. 37, no. 3, 2005, doi: 10.1145/1151954.1067453.
- [21] I. Milne and G. Rowe, "Difficulties in learning and teaching programming - Views of students and tutors," *Educ. Inf. Technol.*, vol. 7, no. 1, 2002, doi: 10.1023/A:1015362608943.
- [22] P. H. Tan, C. Y. Ting, and S. W. Ling, "Learning difficulties in programming courses: Undergraduates' perspective and perception," in *ICCTD 2009 - 2009 International Conference on Computer Technology and Development*, 2009, vol. 1, doi: 10.1109/ICCTD.2009.188.
- [23] L. N. Z. Puerta and M. C. Gomez-Alvarez, "Gamified teaching model to promote learning of algorithms and programming," in *Iberian Conference on Information Systems and Technologies, CISTI*, 2020, vol. 2020-June, doi: 10.23919/CISTI49556.2020.9141027.
- [24] A. Gomes and A. Mendes, "A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations," in *Proceedings - Frontiers in Education Conference, FIE*, 2015, vol. 2015-Febru, no. February, doi: 10.1109/FIE.2014.7044086.
- [25] B. Brinkman and K. Carter, "The ACM Code of Ethics and Professional Conduct," 2017, doi: 10.1145/3017680.3022340.
- [26] D. Gotterbarn, K. Miller, and S. Rogerson, "Software engineering code of ethics," *Commun. ACM*, vol. 40, no. 11, 1997, doi: 10.1145/265684.265699.
- [27] COPNIA, "Código de ÉTICA para el ejercicio de la Ingeniería en general y sus profesiones afines y auxiliares.," *An. Ing.*, vol. 531 a 533, 1938.
- [28] Mohamed Awad, "A Comparison between Agile and Traditional Software Development Methodologies ," The University of Western Australia, 2005.
- [29] Y. Mansoori, T. Karlsson, and M. Lundqvist, "The influence of the lean startup methodology on entrepreneur-coach relationships in the context of a startup accelerator," *Technovation*, vol. 84–85, 2019, doi: 10.1016/j.technovation.2019.03.001.
- [30] F. J. Llamas Fernández and J. C. Fernández Rodríguez, "La metodología Lean startup: desarrollo y aplicación para el emprendimiento," *Rev. Esc. Adm. Negocios*, no. 84, 2018, doi: 10.21158/01208160.n84.2018.1918.
- [31] P. Middleton, "Lean Software Development: Two Case Studies," *Softw. Qual. J.*, vol. 9, no. 4, 2001, doi: 10.1023/A:1013754402981.
- [32] M. Poppendieck and M. A. Cusumano, "Lean software development: A tutorial," *IEEE Softw.*, vol. 29, no. 5, 2012, doi: 10.1109/MS.2012.107.

- [33] M. P. J. Pepper and T. A. Spedding, "The evolution of lean Six Sigma," *International Journal of Quality and Reliability Management*, vol. 27, no. 2, 2010, doi: 10.1108/02656711011014276.
- [34] Larry Apke, *Understanding the Agile Manifesto: A Brief & Bold Guide to Agile*. Lulu Publishing, 2015.
- [35] O. Hazzan and Y. Dubinsky, "The Agile Manifesto," in *SpringerBriefs in Computer Science*, vol. 0, no. 9783319101569, Springer, 2014, pp. 9–14.
- [36] M. A. G. Darrin and W. S. Devereux, "The Agile Manifesto, design thinking and systems engineering," 2017, doi: 10.1109/SYSCON.2017.7934765.
- [37] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," in *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, 2017, vol. 2017-Janua, doi: 10.1109/CCAA.2017.8229928.
- [38] K. Schwaber, "SCRUM Development Process," in *Business Object Design and Implementation*, 1997.
- [39] S. Sharma and N. Hasteer, "A comprehensive study on state of Scrum development," 2017, doi: 10.1109/CCAA.2016.7813837.
- [40] M. Kapsali, "Systems thinking in innovation project management: A match that works," *Int. J. Proj. Manag.*, vol. 29, no. 4, 2011, doi: 10.1016/j.ijproman.2011.01.003.
- [41] S. Gasik, "A model of project knowledge management," *Proj. Manag. J.*, vol. 42, no. 3, 2011, doi: 10.1002/pmj.20239.
- [42] B. Alshaigy, "Development of an interactive learning tool to teach python programming language," 2013, doi: 10.1145/2462476.2465601.
- [43] D. Nariman, "Impact of the interactive e-learning instructions on effectiveness of a programming course," in *Advances in Intelligent Systems and Computing*, 2021, vol. 1194 AISC, doi: 10.1007/978-3-030-50454-0_61.
- [44] V. Mutiawani and Juwita, "Developing e-learning application specifically designed for learning introductory programming," 2014, doi: 10.1109/ICITSI.2014.7048250.
- [45] S. Chookaew, P. Panjaburee, D. Wanichsan, and P. Laosinchai, "A Personalized E-Learning Environment to Promote Student's Conceptual Learning on Basic Computer Programming," *Procedia - Soc. Behav. Sci.*, vol. 116, 2014, doi: 10.1016/j.sbspro.2014.01.303.
- [46] N. Wellnhammer, M. Dolata, S. Steigler, and G. Schwabe, "Studying with the Help of Digital Tutors: Design Aspects of Conversational Agents that Influence the Learning Process," 2020, doi: 10.24251/hicss.2020.019.
- [47] C. Gabriel, C. Hahne, A. Zimmermann, and F. Lenk, "The Virtual Tutor: Tasks for conversational agents in Online Collaborative Learning Environments," 2021, doi: 10.24251/hicss.2021.012.
- [48] X. Sánchez-Díaz, G. Ayala-Bastidas, P. Fonseca-Ortiz, and L. Garrido, "A knowledge-based methodology for building a conversational chatbot as an intelligent tutor," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11289 LNAI, doi: 10.1007/978-3-030-04497-8_14.

- [49] P. Andrea, O. Cano, E. Camilo, and P. Alarcón, "Recommendation Systems in Education: A review of Recommendation Mechanisms in E-learning Environments Recommendation Systems in Education: A review of Recommendation Mechanisms in E-learning Environments *," *Rev. Ing. Univ. Medellín*, vol. 20, no. 38, pp. 147–158, 2021, doi: 10.22395/rium.v20n38a9.
- [50] G. D. S. Leitao, E. B. Valentin, E. H. T. De Oliveira, and R. D. S. Barreto, "Survey on pedagogical resources recommendation using cognitive computing systems," in *Proceedings - Frontiers in Education Conference, FIE*, 2019, vol. 2018-October, doi: 10.1109/FIE.2018.8658793.
- [51] S. C. Harris and V. Kumar, "Identifying student difficulty in a digital learning environment," 2018, doi: 10.1109/ICALT.2018.00054.
- [52] C. Winberg, "Teaching engineering/engineering teaching: Interdisciplinary collaboration and the construction of academic identities," *Teach. High. Educ.*, vol. 13, no. 3, 2008, doi: 10.1080/13562510802045394.
- [53] C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey, and L. J. Leifer, "Engineering design thinking, teaching, and learning," *IEEE Eng. Manag. Rev.*, vol. 34, no. 1, pp. 65–90, 2006, doi: 10.1109/emr.2006.1679078.
- [54] C. L. Dym, "Learning engineering: Design, languages, and experiences," *Journal of Engineering Education*, vol. 88, no. 2. 1999, doi: 10.1002/j.2168-9830.1999.tb00425.x.
- [55] Z. Liu and D. J. Schönwetter, "Teaching creativity in engineering," *International Journal of Engineering Education*, vol. 20, no. 5. 2004.
- [56] E. Mehmood, A. Abid, M. S. Farooq, and N. A. Nawaz, "Curriculum, Teaching and Learning, and Assessments for Introductory Programming Course," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3008321.
- [57] C. S. Miller and A. Settle, "Learning to get literal: Investigating reference-point difficulties in novice programming," *ACM Trans. Comput. Educ.*, vol. 19, no. 3, 2019, doi: 10.1145/3313291.
- [58] Y. Qian, P. Yan, and M. Zhou, "Using Data to Understand Difficulties of Learning to Program: A Study with Chinese Middle School Students," 2019, doi: 10.1145/3300115.3309521.
- [59] A. Gomes and A. Mendes, "A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations," in *Proceedings - Frontiers in Education Conference, FIE*, 2015, vol. 2015-February, no. February, doi: 10.1109/FIE.2014.7044086.
- [60] W. Y. Hwang, R. Shadiev, C. Y. Wang, and Z. H. Huang, "A pilot study of cooperative programming learning behavior and its relationship with students' learning performance," *Comput. Educ.*, vol. 58, no. 4, 2012, doi: 10.1016/j.compedu.2011.12.009.
- [61] A. Alammary, "Blended learning models for introductory programming courses: A systematic review," *PLoS ONE*, vol. 14, no. 9. 2019, doi: 10.1371/journal.pone.0221765.
- [62] T. Altintas, A. Gunes, and H. Sayan, "A peer-assisted learning experience in computer programming language learning and developing computer programming skills," *Innov. Educ. Teach. Int.*, vol. 53, no. 3, 2016, doi:

- 10.1080/14703297.2014.993418.
- [63] C. González-Sacristán, P. Molins-Ruano, F. Díez, P. Rodríguez, and G. M. Sacha, "Computer-assisted assessment with item classification for programming skills," 2013, doi: 10.1145/2536536.2536554.
 - [64] R. P. Medeiros, G. L. Ramalho, and T. P. Falcao, "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," *IEEE Trans. Educ.*, vol. 62, no. 2, 2019, doi: 10.1109/TE.2018.2864133.
 - [65] W. W. Cobern, "Constructivism," *J. Educ. Psychol. Consult.*, vol. 4, no. 1, pp. 105–112, 1993.
 - [66] R. Fox, "Constructivism Examined," *Oxford Rev. Educ.*, vol. 27, no. 1, 2001, doi: 10.1080/03054980125310.
 - [67] A. Sutinen, "Constructivism and education: Education as an interpretative transformational process," *Stud. Philos. Educ.*, vol. 27, no. 1, 2008, doi: 10.1007/s11217-007-9043-5.
 - [68] E. J. Hyslop-Margison and J. Strobel, "CONSTRUCTIVISM AND EDUCATION: MISUNDERSTANDINGS AND PEDAGOGICAL IMPLICATIONS," *Teach. Educ.*, vol. 43, no. 1, 2007, doi: 10.1080/08878730701728945.
 - [69] W. Detel, "Social Constructivism," *Int. Encycl. Soc. Behav. Sci.*, vol. 2, pp. 228–234, 2015.
 - [70] M. Barak, "Science Teacher Education in the Twenty-First Century: a Pedagogical Framework for Technology-Integrated Social Constructivism," *Res. Sci. Educ.*, vol. 47, no. 2, 2017, doi: 10.1007/s11165-015-9501-y.
 - [71] K. H. Au, "Social constructivism and the school literacy learning of students of diverse backgrounds," *J. Lit. Res.*, vol. 30, no. 2, 1998, doi: 10.1080/10862969809548000.
 - [72] G. Santos, E. Mandado, R. Silva, and M. Doiro, "Engineering learning objectives and computer assisted tools," *Eur. J. Eng. Educ.*, vol. 44, no. 4, 2019, doi: 10.1080/03043797.2018.1563585.
 - [73] B. Dalgarno, "Interpretations of constructivism and consequences for Computer Assisted Learning," *Br. J. Educ. Technol.*, vol. 32, no. 2, 2001, doi: 10.1111/1467-8535.00189.
 - [74] Y. J. Seo and H. Woo, "The identification, implementation, and evaluation of critical user interface design features of computer-assisted instruction programs in mathematics for students with learning disabilities," *Comput. Educ.*, vol. 55, no. 1, 2010, doi: 10.1016/j.compedu.2010.02.002.
 - [75] R. Garcia, K. Falkner, and R. Vivian, "Systematic literature review: Self-Regulated Learning strategies using e-learning tools for Computer Science," *Comput. Educ.*, vol. 123, 2018, doi: 10.1016/j.compedu.2018.05.006.
 - [76] C. Serban and A. Vescan, "Advances in designing a student-centered learning process using cutting-edge methods, tools, and artificial intelligence: An e-learning platform," 2019, doi: 10.1145/3340435.3342716.
 - [77] K. M. Y. Law, V. C. S. Lee, and Y. T. Yu, "Learning motivation in e-learning facilitated computer programming courses," *Comput. Educ.*, vol. 55, no. 1,

- 2010, doi: 10.1016/j.compedu.2010.01.007.
- [78] S. Hobert, "Say hello to 'Coding Tutor'! Design and evaluation of a chatbot-based learning system supporting students to learn to program," 1984.
- [79] A. Oker, F. Pecune, and C. Declercq, "Virtual tutor and pupil interaction: A study of empathic feedback as extrinsic motivation for learning," *Educ. Inf. Technol.*, 2020, doi: 10.1007/s10639-020-10123-5.
- [80] S. George, "From sex and therapy bots to virtual assistants and tutors: How emotional should artificially intelligent agents be?," 2019, doi: 10.1145/3342775.3342807.
- [81] J. Díaz-Ramírez, "Gamification in engineering education – An empirical assessment on learning and game performance," *Heliyon*, vol. 6, no. 9, 2020, doi: 10.1016/j.heliyon.2020.e04972.
- [82] S. Diaz, J. Diaz, and D. Ahumada, "A gamification approach to improve motivation on an initial programming course," 2019, doi: 10.1109/ICA-ACCA.2018.8609701.
- [83] M. Carreno-Leon, A. Sandoval-Bringas, F. Alvarez-Rodriguez, and Y. Camacho-Gonzalez, "Gamification technique for teaching programming," in *IEEE Global Engineering Education Conference, EDUCON*, 2018, vol. 2018-April, doi: 10.1109/EDUCON.2018.8363482.
- [84] S. Azmi, N. A. Iahad, and N. Ahmad, "Attracting students' engagement in programming courses with gamification," 2017, doi: 10.1109/IC3e.2016.8009050.
- [85] Z. Zainuddin, S. K. W. Chu, M. Shujahat, and C. J. Perera, "The impact of gamification on learning and instruction: A systematic review of empirical evidence," *Educ. Res. Rev.*, vol. 30, 2020, doi: 10.1016/j.edurev.2020.100326.
- [86] F. D. de la Peña Esteban, J. A. Lara Torralbo, D. Lizcano Casas, and M. C. Burgos García, "Web gamification with problem simulators for teaching engineering," *J. Comput. High. Educ.*, vol. 32, no. 1, 2020, doi: 10.1007/s12528-019-09221-2.
- [87] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1. pp. 1–49, Oct. 2012, doi: 10.1016/j.physrep.2012.02.006.
- [88] S. Dwivedi and V. S. K. Roshni, "Recommender system for big data in education," 2017, doi: 10.1109/ELELTECH.2017.8074993.
- [89] A. C. Rivera, M. Tapia-Leon, and S. Lujan-Mora, "Recommendation systems in education: A systematic mapping study," in *Advances in Intelligent Systems and Computing*, 2018, vol. 721, doi: 10.1007/978-3-319-73450-7_89.
- [90] G. D. S. Leitao, E. B. Valentin, E. H. T. De Oliveira, and R. D. S. Barreto, "Survey on pedagogical resources recommendation using cognitive computing systems," in *Proceedings - Frontiers in Education Conference, FIE*, 2019, vol. 2018-Octob, doi: 10.1109/FIE.2018.8658793.
- [91] Real Academia Española, *Diccionario de la lengua española*, 23rd ed. Madrid: España, 2014.
- [92] Y. Kondo, "Customer satisfaction: How can I measure it?," *Total Qual. Manag.*, vol. 12, no. 7, 2001, doi: 10.1080/09544120120095981.

- [93] J. A. Czepiel and L. J. Rosenberg, "Consumer satisfaction: Concept and measurement," *J. Acad. Mark. Sci.*, vol. 5, no. 4, 1977, doi: 10.1007/BF02722070.
- [94] G. Albaum, "The Likert scale revisited: An alternate version," *Int. J. Mark. Res.*, vol. 39, no. 2, 1997, doi: 10.1177/147078539703900202.
- [95] J. L. Nicolau, J. P. Mellinas, and E. Martín-Fuentes, "Satisfaction measures with monetary and non-monetary components: Hotel's overall scores," *Int. J. Hosp. Manag.*, vol. 87, 2020, doi: 10.1016/j.ijhm.2020.102497.
- [96] D. F. Alwin, "Feeling thermometers versus 7-point scales: Which are better?," *Sociol. Methods Res.*, vol. 25, no. 3, 1997, doi: 10.1177/0049124197025003003.
- [97] N. I. Fisher and R. E. Kordupleski, "Good and bad market research: A critical review of Net Promoter Score," *Appl. Stoch. Model. Bus. Ind.*, vol. 35, no. 1, 2019, doi: 10.1002/asmb.2417.
- [98] M. A. Laitinen, "Net Promoter Score as Indicator of Library Customers' Perception," *J. Libr. Adm.*, vol. 58, no. 4, 2018, doi: 10.1080/01930826.2018.1448655.
- [99] D. Jing and D. C. Zhao, "Developing a Scale to Measure University Student's Satisfaction with MOOC," 2020, doi: 10.1145/3402569.3402586.
- [100] H. Tokkonen and P. Saariluoma, "How user experience is understood?," 2013.
- [101] Organización Internacional de Normalización, "Norma ISO 25000." <https://iso25000.com/> (accessed Apr. 18, 2021).
- [102] J. Moreno Chaustre, L. Bolaños Rengifo, and M. Navia Porras, "Exploración de modelos y estándares de calidad para el producto software," *Rev. UIS Ing.*, vol. 9, no. 1, 2010.
- [103] B. Fehnert and A. Kosagowsky, "Measuring user experience: Complementing qualitative and quantitative assessment," 2008, doi: 10.1145/1409240.1409294.
- [104] K. Schulze and H. Krömker, "A framework to measure User eXperience of interactive online products," 2011, doi: 10.1145/1931344.1931358.
- [105] R. Bernhaupt, M. Eckschlager, and M. Tscheligi, "Methods for evaluating games: How to measure usability and user experience in games?," in *ACM International Conference Proceeding Series*, 2007, vol. 203, doi: 10.1145/1255047.1255142.
- [106] I. Sommerville, *Ingeniería de Software*, 9th ed. México: Pearson Educación, 2011.
- [107] C. E. Vasquez, *Ingeniería de Requisitos: Software Orientado Al Negocio*, 1st ed. Independently published, 2016.
- [108] A. Cockburn, *Writing Effective Use Cases*. 2000.
- [109] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A model for technology transfer in practice," *IEEE Softw.*, vol. 23, no. 6, 2006, doi: 10.1109/MS.2006.147.
- [110] R. 'Hernández Sampieri, C. 'Fernández Collado, and M. del P. 'Baptista Lucio, *Metodología de la investigación*, 6th ed. México D.F: McGraw-Hill, 2014.

11. ANEXOS

11.1 ANEXO 1

[Modelo Biopsicosocial y Cultural – Problema](#)

11.2 ANEXO 2

[Árbol de problemas](#)

11.3 ANEXO 3

[Modelo Biopsicosocial y Cultural – Solución](#)

11.4 ANEXO 4

[Work Breakdown Structure / Estructura de Descomposición del Trabajo \(WBS / EDT\)](#)

11.5 ANEXO 5

[Cronograma de actividades](#)

11.6 ANEXO 6

[Documento de Arquitectura de Software \(SAD\)](#)

11.7 ANEXO 7

[Manual de Usuario](#)

11.8 ANEXO 8

[Cuestionario de evaluación](#)

11.9 ANEXO 9

[Repositorio de código y video de funcionamiento del artefacto de software](#)

11.10 ANEXO 10

[Manual técnico](#)

11.11 ANEXO 11

[Reflexiones periódicas](#)

11.12 ANEXO 12

[Resultados obtenidos del proceso de validación dinámica](#)

11.13 ANEXO 13

[Plan de pruebas](#)