

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

UNIDAD DE TITULACIÓN

**DESARROLLO DE LOS MÓDULOS DE PRESENTACIÓN DE LA
RECOMENDACIÓN Y DE SELECCIÓN DE TÓPICOS DEL SISTEMA
DE RECOMENDACIÓN PARA GRUPOS DE INVESTIGADORES
QUE SOPORTE LA TOMA DE DECISIONES Y EL CONSENSO**

**TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SOFTWARE**

RENATO FABRICIO PADILLA MOLINA

renato.padilla@epn.edu.ec

renato_bga@live.com

DIRECTORA: PhD. LORENA KATHERINE RECALDE CERDA

lorena.recalde@epn.edu.ec

Quito, octubre del 2023

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Renato Fabricio Padilla Molina, bajo mi supervisión.

PhD. Lorena Katherine Recalde Cerda
DIRECTORA DE PROYECTO

DECLARACIÓN DE AUTORÍA

Yo, Renato Fabricio Padilla Molina, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Renato Fabricio Padilla Molina

DEDICATORIA

El presente proyecto está dedicado a las personas que siempre creyeron en mi y me brindaron su apoyo.

A mi madre Isabel Molina, cuya dedicación y sacrificio ha sido mi mayor fuente de motivación para superarme cada día y no rendirme jamás.

A mis queridos hermanos Erick y Sabrina, les dedico este logro con cariño, y deseo que al igual que yo no se rindan y cumplan sus anhelos.

A mi abuela Yola, quien no puede estar presente para ser testigo de este acontecimiento pero siempre la llevo en mis pensamientos.

A Jomaira Montaluisa, quien ha estado junto a mi casi desde los inicios de mis estudios, siempre me brindó su apoyo incondicional y me motivó cuando sentía que no podía dar más de mi.

A todos los que han sido partícipes en mi trayectoria les brindo este logro con profunda gratitud y amor.

Renato Fabricio Padilla Molina

AGRADECIMIENTOS

En primer lugar, deseo expresar mi más sincero agradecimiento a mi madre, quien ha sido mi mayor fuente de inspiración y apoyo a lo largo de mi trayectoria académica. Valorando enormemente el sacrificio y dedicación que ha invertido para brindarme una educación de alta calidad, reconozco que su amor y aliento constante han sido el combustible que me motiva a buscar nuevas metas y a superar cualquier obstáculo que se presente.

Además, quiero extender un agradecimiento especial a la PHD. Lorena Recalde, quien ha sido una guía inestimable durante todo el proceso de elaboración de esta tesis. Su vasta experiencia, profundo conocimiento y paciencia ejemplar han sido pilares fundamentales en la consecución de los objetivos planteados. Su apoyo ha sido esencial para alcanzar el nivel de logro que hoy se presenta.

Renato Fabricio Padilla Molina

CONTENIDO

Resumen	1
Abstract	2
1 INTRODUCCIÓN	3
1.1 Planteamiento del problema	3
1.2 Justificación teórica	5
1.3 Justificación Metodológica	5
1.4 Justificación práctica	6
1.5 Objetivos	6
1.5.1 Objetivo general	6
1.5.2 Objetivos específicos	6
1.6 Alcance	7
1.7 Marco teórico	7
1.7.1 Sistema de recomendación grupal	7
1.7.2 Dinámica de grupo	7
1.7.3 Facilitación de grupos	8
1.7.4 Elicitación de preferencias	8
1.7.5 Herramientas utilizadas	9
2 METODOLOGÍA	11
2.1 Introducción a la metodología de desarrollo ágil	11
2.2 Ciclo de Desarrollo Iterativo del Software	13
2.3 Roles	13
2.4 Requisitos y especificaciones	15
2.5 Product Backlog	15
2.6 Planificación de los <i>sprints</i>	19
2.7 Sprint 0	19
2.7.1 Objetivos del sprint	19
2.7.2 Ejecución del sprint	19
2.7.3 Revisión y retrospectiva del sprint	26
2.8 Sprint 1	28

2.8.1	Objetivos del sprint	28
2.8.2	Ejecución del sprint	28
2.8.3	Revisión y retrospectiva del sprint	32
2.9	Sprint 2	32
2.9.1	Objetivos del sprint	32
2.9.2	Ejecución del sprint	33
2.9.3	Revisión y retrospectiva del sprint	35
2.10	Sprint 3	36
2.10.1	Objetivos del sprint	36
2.10.2	Ejecución del sprint	36
2.10.3	Revisión y retrospectiva del sprint	39
2.11	Sprint 4	40
2.11.1	Objetivos del sprint	40
2.11.2	Ejecución del sprint	40
2.11.3	Revisión y retrospectiva del sprint	43
2.12	Sprint 5	43
2.12.1	Objetivos del sprint	43
2.12.2	Ejecución del sprint	44
2.12.3	Revisión y retrospectiva del sprint	47
2.13	Sprint 6	47
2.13.1	Objetivos del sprint	47
2.13.2	Ejecución del sprint	47
2.13.3	Revisión y retrospectiva del sprint	49
3	RESULTADOS Y DISCUSIÓN	50
3.1	Producto final	50
3.1.1	Presentación y búsqueda simple de tópicos	50
3.1.2	Búsqueda combinada de tópicos	51
3.1.3	Reordenamiento y calificación de tópicos	52
3.1.4	Presentación de tópicos aceptados y descartados, junto con sus justificaciones	54
3.2	Nivel de usabilidad	54
3.3	Métricas de evaluación	56
4	CONCLUSIONES y RECOMENDACIONES	57
4.1	Conclusiones	57
4.2	Recomendaciones	58

5	REFERENCIAS BIBLIOGRÁFICAS	59
6	ANEXOS	62

ÍNDICE DE FIGURAS

Figura 1.1 Diagrama del Sistema de Toma de Decisiones y Consenso Grupal Aplicado a los Motores de Recomendación. Elaborado por el Autor.	4
Figura 2.1 Valores del manifiesto ágil. Elaborado por el autor.	11
Figura 2.2 Desarrollo dirigido por el comportamiento. Elaborado por el autor. . .	12
Figura 2.3 Agile Testing [24].	13
Figura 2.4 Jerarquía de los ítems del product backlog [26].	16
Figura 2.5 Product backlog. Elaborado por el autor.	17
Figura 2.6 Arquitectura técnica de los módulos <i>B</i> y <i>C</i> . Elaborado por el Autor. . .	20
Figura 2.7 <i>React native</i> levantado. Elaborado por el Autor.	22
Figura 2.8 Inicio de <i>Firebase</i> . Elaborado por el Autor.	22
Figura 2.9 <i>Jest</i> levantado. Elaborado por el Autor.	23
Figura 2.10 Obtención de votación de miembro. Elaborado por el Autor.	25
Figura 2.11 Actualización de votación de miembro. Elaborado por el Autor.	25
Figura 2.12 Obtención de tópicos recomendados. Elaborado por el Autor.	27
Figura 2.13 Actualización de tópicos recomendados. Elaborado por el Autor. . . .	27
Figura 2.14 Escenarios de comportamiento de la obtención y ordenamiento de tópicos. Elaborado por el Autor.	28
Figura 2.15 Prueba de la obtención de tópicos. Elaborado por el Autor.	29
Figura 2.16 Prueba del ordenamiento de tópicos. Elaborado por el Autor.	29
Figura 2.17 Colecciones para la obtención de tópicos. Elaborado por el autor. . . .	30
Figura 2.18 Servicios de obtención y ordenamiento de tópicos. Elaborado por el autor.	30
Figura 2.19 Interfaz de obtención y ordenamiento de tópicos web. Elaborado por el autor.	31
Figura 2.20 Interfaz de obtención y ordenamiento de tópicos móvil. Elaborado por el autor.	31
Figura 2.21 Resultado de la ejecución de las pruebas de las funciones de obtención y ordenamiento de tópicos. Elaborado por el autor.	32
Figura 2.22 Escenarios de comportamiento del redireccionamiento hacia Google Scholar de forma simple y combinada. Elaborado por el Autor.	33

Figura 2.23 Prueba de búsqueda simple en Google Scholar. Elaborado por el Autor.	33
Figura 2.24 Prueba de búsqueda combinada en Google Scholar. Elaborado por el Autor.	34
Figura 2.25 Interfaz de búsqueda en Google Scholar web. Elaborado por el autor.	34
Figura 2.26 Interfaz de búsqueda en Google Scholar móvil. Elaborado por el autor.	35
Figura 2.27 Función de búsqueda en Google Scholar. Elaborado por el autor.	35
Figura 2.28 Resultado de las pruebas ejecutadas sobre las funciones búsqueda simple y combinada de tópicos. Elaborado por el autor.	35
Figura 2.29 Escenario de comportamiento para el reordenamiento de tópicos. Elaborado por el Autor.	37
Figura 2.30 Escenario de comportamiento para el guardado de tópicos. Elaborado por el Autor.	37
Figura 2.31 Prueba de reordenamiento de tópicos. Elaborado por el Autor.	37
Figura 2.32 Prueba de guardado de tópicos. Elaborado por el Autor.	37
Figura 2.33 Interfaz de reordenamiento de tópicos web. Elaborado por el autor.	38
Figura 2.34 Interfaz de reordenamiento de tópicos móvil. Elaborado por el autor.	38
Figura 2.35 Función de reordenamiento de tópicos. Elaborado por el autor.	39
Figura 2.36 Servicio de guardado de tópicos. Elaborado por el autor.	39
Figura 2.37 Pruebas ejecutadas del reordenamiento y guardado de tópicos. Elaborado por el autor.	39
Figura 2.38 Escenario de comportamiento para la calificación de tópicos. Elaborado por el Autor.	40
Figura 2.39 Prueba de calificación de tópicos. Elaborado por el Autor.	41
Figura 2.40 Interfaz de calificación de tópicos web. Elaborado por el autor.	41
Figura 2.41 Interfaz de calificación de tópicos móvil. Elaborado por el autor.	42
Figura 2.42 Servicio de calificación de tópicos. Elaborado por el autor.	42
Figura 2.43 Función de calificación de tópicos. Elaborado por el autor.	42
Figura 2.44 Prueba ejecutada de la calificación de tópicos. Elaborado por el autor.	43
Figura 2.45 Escenario de comportamiento para la obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el Autor.	44
Figura 2.46 Prueba de obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el Autor.	44
Figura 2.47 Interfaz de obtención de tópicos aceptados y descartados, junto con sus justificaciones web. Elaborado por el autor.	45
Figura 2.48 Interfaz de obtención de tópicos aceptados y descartados, junto con sus justificaciones móvil. Elaborado por el autor.	45

Figura 2.49 Servicio de obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.	46
Figura 2.50 Función de presentación de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.	46
Figura 2.51 Prueba ejecutada de la obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.	46
Figura 2.52 Sesiones presenciales de prueba con los usuarios. Elaborado por el autor.	49
Figura 3.1 Presentación de tópicos. Elaborado por el autor.	51
Figura 3.2 Búsqueda simple de tópicos. Elaborado por el autor.	51
Figura 3.3 Selección de tópicos. Elaborado por el autor.	52
Figura 3.4 Búsqueda combinada de tópicos. Elaborado por el autor.	52
Figura 3.5 Reordenamiento y calificación de tópicos. Elaborado por el autor.	53
Figura 3.6 Mensaje de advertencia. Elaborado por el autor.	53
Figura 3.7 Tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.	54
Figura 3.8 Puntaje <i>SUS</i> [33].	55
Figura 3.9 Tiempo promedio para comprender e investigar los tópicos.	56
Figura 3.10 Tiempo promedio para la toma de la decisión final sobre los tópicos.	56

ÍNDICE DE TABLAS

Tabla 1.1	Etapas para el <i>Taller de Consenso</i>	9
Tabla 1.2	Herramientas utilizadas.	10
Tabla 2.1	Sprints.	14
Tabla 2.2	Roles y responsables del equipo Scrum.	14
Tabla 2.3	Requerimientos del módulo del sistema.	15
Tabla 2.4	Épicas.	16
Tabla 2.5	Características del sistema.	17
Tabla 2.7	Historias de usuarios.	18
Tabla 2.8	<i>Sprint planning</i>	19
Tabla 2.9	Convenciones y estándares de desarrollo y documentación.	21
Tabla 2.11	Recursos de software y hardware.	21
Tabla 2.12	Criterios para la ejecución de las pruebas de usabilidad.	48
Tabla 2.13	Preguntas de usabilidad [32].	48
Tabla 3.1	Evaluación SUS.	55

RESUMEN

Este trabajo aborda la problemática de la toma de decisiones y el consenso en grupos de investigadores dentro del ámbito académico. Se destaca la importancia del trabajo colaborativo en proyectos de investigación y cómo la falta de autoridad jerárquica puede generar desafíos en la toma de decisiones. Se mencionan factores como la comunicación, el conflicto de intereses y las diferencias de habilidades que pueden afectar la toma de decisiones grupales. Se propone el uso de una herramienta informática para guiar este proceso y lograr decisiones más equilibradas.

El objetivo general es desarrollar módulos para un sistema de recomendación que respalde la toma de decisiones y el consenso en grupos de investigadores. Se detalla el alcance del proyecto, que incluye la implementación de módulos de presentación de recomendaciones y selección de tópicos, sin abordar la implementación de algoritmos de recomendación.

La metodología BDD (Behavior Driven Development) se elige para el desarrollo del proyecto, debido a su enfoque en el comportamiento esperado del software y su adaptabilidad a los cambios en los requisitos del sistema. La metodología BDD se centra en pruebas de alto nivel y se destaca su flujo de desarrollo.

PALABRAS CLAVE: Sistema de recomendación grupal, Trabajo colaborativo, Toma de decisiones, Consenso grupal, BDD y Scrum

ABSTRACT

This work addresses the issue of decision-making and consensus in groups of researchers within the academic context. The significance of collaborative work in research projects is highlighted, along with how the absence of hierarchical authority can pose challenges in decision-making. Factors such as communication, conflicting interests, and differences in skills that can impact group decision-making are discussed. The use of an IT tool to guide this process and achieve more balanced decisions is proposed.

The overall objective is to develop modules for a recommendation system that supports decision-making and consensus in groups of researchers. The project scope includes the implementation of modules for presenting recommendations and selecting topics, without delving into the implementation of recommendation algorithms.

The BDD (Behavior Driven Development) methodology is chosen for project development, owing to its focus on the expected behavior of software and its adaptability to changes in system requirements. The BDD methodology centers on high-level testing, and its development flow is highlighted.

Keywords: Group recommendation system, Collaborative work, Decision-making, Group consensus, BDD and Scrum

1 INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

Dentro del mundo académico es muy común la generación de trabajo colaborativo para realizar proyectos de investigación. En el trabajo colaborativo, la noción de autoridad se diferencia claramente de una interacción jerarquizada, por cuanto no se impone la visión de un miembro del grupo por el sólo hecho de tener autoridad, sino que el gran desafío es argumentar puntos de vista, justificar e intentar convencer a los pares [1]. Algunas de las causas por las cuales esta actividad no se lleva a cabo de manera inmediata es la comunicación, el conflicto de intereses y las diferencias de habilidades de los involucrados al momento de llegar a un consenso de hacia dónde se va a dirigir un proyecto de investigación.

La toma de decisiones grupales es un proceso de análisis y estudio de las distintas alternativas sobre cómo se puede abordar una temática, intervienen factores como la experiencia individual y colectiva de los contribuidores. Aunque hay técnicas disponibles para facilitar la realización de este proceso, existe otro tipo de factores propios de las interacciones sociales entre los miembros que pueden ofuscar esta actividad. Las decisiones grupales dependen de múltiples variables dinámicas y tienen su propia complejidad, pues cada miembro del equipo opina, puede estar de acuerdo o en desacuerdo, y cada acción puede afectar las decisiones del equipo [2].

El uso de una herramienta informática que guíe el proceso de toma de decisiones y consenso grupal es una alternativa para evitar el conflicto social mencionado. Este proceso se puede dirigir mediante un sistema o herramienta informática que siempre tenga en cuenta las opiniones de todos los miembros, así como también los conocimientos e investigaciones que estos han realizado y publicado con anterioridad, para así de tener una decisión grupal más justa y equilibrada.

En virtud de lo expuesto, el presente proyecto propone el desarrollo de un aplicativo móvil

y web que implemente dos módulos para el Sistema de Toma de Decisiones y Consenso Grupal Aplicado a los Motores de Recomendación. En el diagrama presentado en la Figura 1.1 se puede observar un esquema general que ilustra la interacción entre los módulos y sus respectivos componentes, proporcionando una visión del funcionamiento del aplicativo.

El módulo A está basado en la adscripción de los usuarios como miembros de un grupo de investigación. El módulo B se especializa en la generación de recomendaciones grupales a partir de las preferencias obtenidas de los investigadores y luego las expone para ser consumidas. Finalmente, el módulo C es el encargado de presentar las recomendaciones al grupo de investigación y permitir que los miembros reordenen las recomendaciones a partir de sus criterios grupales e individuales.

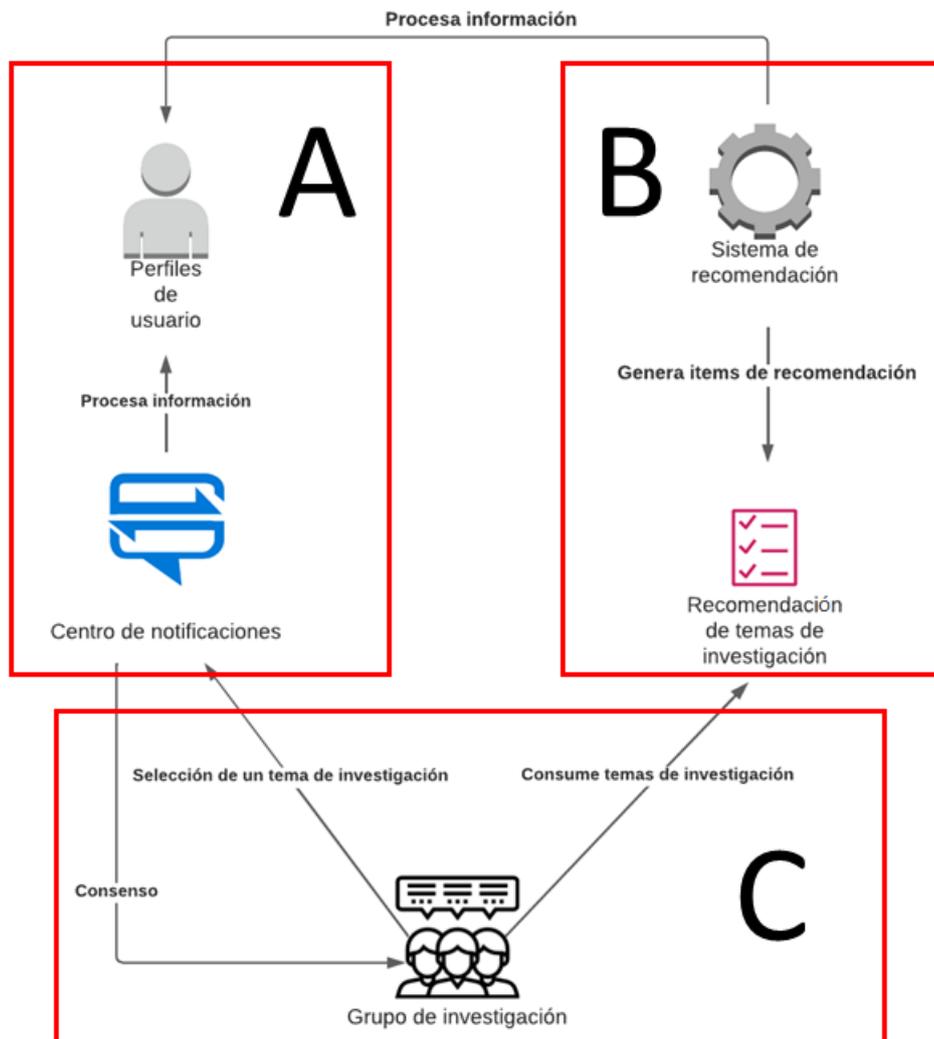


Figura 1.1: Diagrama del Sistema de Toma de Decisiones y Consenso Grupal Aplicado a los Motores de Recomendación. Elaborado por el Autor.

1.2 JUSTIFICACIÓN TEÓRICA

La justificación teórica de este proyecto se basa en la necesidad de abordar los desafíos y las limitaciones del trabajo colaborativo en el ámbito académico, especialmente en lo que respecta a la toma de decisiones grupales y el consenso en proyectos de investigación. En los GRS(Group Recommender Systems), se trata de considerar todo el proceso de toma de decisiones del grupo y apoyar al grupo en las diversas fases de toma de decisiones, que incluyen: la discusión de las opciones de los candidatos, el intercambio de opiniones e información, y la llegada a un decisión definitiva [3].

Aunque existen técnicas disponibles para facilitar este proceso, las interacciones sociales entre los miembros del grupo pueden complicarlo aún más. Las decisiones grupales son influenciadas por múltiples variables dinámicas y presentan su propia complejidad, ya que cada miembro del equipo tiene su opinión y sus acciones pueden afectar las decisiones del grupo. Por ejemplo, CHARM es una herramienta que permite el desarrollo de GRS ayudando a grupos de usuarios a decidir sobre un elemento/producto que el grupo experimentará [3].

Ante estos desafíos, el presente proyecto propone el desarrollo de un aplicativo móvil y web que implemente dos módulos para el Sistema de Toma de Decisiones y Consenso Grupal Aplicado a los Motores de Recomendación. Además, se busca establecer un proceso de toma de decisiones que permita llegar a un consenso sobre la recomendación presentada.

1.3 JUSTIFICACIÓN METODOLÓGICA

El desarrollo del presente proyecto se basará en la integración de Scrum y BDD como enfoques complementarios para el desarrollo de software. Scrum es un marco de trabajo iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones [4]. Por otro lado, BDD es un proceso de desarrollo de software ágil de segundo plano, que congenia con diversas metodologías ágiles, en especial, en la frecuencia de programación extrema para el desarrollo de software y microservicios [5]. La combinación de estas metodologías permite una mejor comprensión de los requisitos, una mayor colaboración entre los interesados y el equipo de desarrollo, y una mejora en la calidad del software a través de la ejecución de pruebas continuas. Para mayor detalle, la especificación de la metodología integrada está

en la Sección 2.1.

1.4 JUSTIFICACIÓN PRÁCTICA

Este proyecto se basa en la necesidad de abordar los desafíos prácticos que enfrentan los grupos de investigadores al seleccionar tópicos de investigación y llegar a un consenso. La falta de una herramienta eficiente y colaborativa para gestionar y evaluar opciones de investigación puede llevar a decisiones no óptimas, falta de consenso y disminución de la productividad del grupo. Al desarrollar un sistema de recomendación específico para grupos de investigadores, se espera proporcionar una solución práctica que mejore la eficiencia y la calidad de las decisiones de investigación, facilitando la selección de tópicos relevantes y promoviendo la colaboración y el consenso entre los miembros del grupo.

1.5 OBJETIVOS

1.5.1 Objetivo general

Desarrollar los módulos de presentación de la recomendación y de selección de tópicos para un sistema de recomendación para grupos de investigadores que soporte la toma de decisiones y el consenso.

1.5.2 Objetivos específicos

- Entender el estado del arte acerca de los sistemas de recomendación grupal y los mecanismos para mediación y consenso
- Comprender los datos relacionados a la información académica de los investigadores y su uso en un sistema de recomendación grupal
- Desarrollar y evaluar los módulos de presentación de la recomendación y de selección de tópicos
- Validar el funcionamiento de todo el sistema mediante pruebas funcionales y pruebas de aceptación

- ❑ Generar documentación del sistema y del desarrollo del proyecto

1.6 ALCANCE

El presente proyecto propone el desarrollo de una aplicación web y móvil que implemente dos módulos para el sistema de recomendación para grupos de investigadores que soporte la toma de decisiones y el consenso. El presente componente (B y C) tiene como alcance desarrollar el módulo de presentación de la recomendación (pre-calculada) y el módulo de selección de tópicos. Es así, que no se contempla la implementación de algoritmos de recomendación.

1.7 MARCO TEÓRICO

1.7.1 Sistema de recomendación grupal

Un sistema de recomendación de grupo (SRG) es un sistema que recomienda artículos a un grupo de usuarios de forma colectiva, dadas sus preferencias [6]. De este modo un sistema de recomendación grupal es una variante de los sistemas de recomendación que tiene como objetivo proporcionar recomendaciones de manera colaborativa para un grupo de usuarios en lugar de individualmente. En lugar de centrarse en las preferencias individuales de cada usuario, un sistema de recomendación grupal tiene en cuenta las preferencias y características del grupo en su conjunto.

Con lo cual se desea agilizar el proceso que conlleva el consenso y la toma de decisiones aplicado a grupos de investigadores. En ese contexto, estos grupos buscan la generación de items de investigación considerando que los sistemas de recomendación son sistemas de apoyo a la decisión que ayudan a los usuarios a identificar uno o más elementos que satisfagan sus deseos y necesidades [7].

1.7.2 Dinámica de grupo

La *Dinámica de grupo* adquiere relevancia debido a las relaciones interpersonales que se establecen entre las personas, permitiéndonos comunicarnos y colaborar. Como seres so-

ciales, nuestra capacidad de expresión se lleva a cabo a través de la palabra, pero estas palabras no se emiten en un contexto aislado. Los grupos son el entorno que enmarca nuestras interacciones sociales e individuales, y desempeñan un papel significativo en nuestras experiencias diarias. Desde un punto de vista social un grupo es un número determinado de miembros, quienes para alcanzar un objetivo común, participan durante un periodo de tiempo prolongado [8].

Los grupos tienen un impacto importante en la definición y socialización de las personas, tanto a nivel individual como colectivo. El conocimiento de la dinámica de grupos se utiliza para intervenir y realizar investigaciones en estos contextos. Así, se puede definir a la *Dinámica de grupo* como el conjunto de procedimientos y medios para ser utilizados en situaciones grupales, con el fin de que se manifieste de manera más expresa los fenómenos, hechos y procesos grupales [8].

1.7.3 Facilitación de grupos

La forma en que interactúan los miembros de un grupo de investigadores es fundamental para tener un proceso de toma de decisión y consenso más óptimo. La facilitación de grupos es un conjunto de herramientas, técnicas y habilidades para garantizar el buen funcionamiento de un grupo, tanto en la consecución de sus objetivos y realización de su visión colectiva, como en la creación de un clima relacional donde reine la confianza y una comunicación fluida, empática y honesta [9].

En este sentido, una de las técnicas existentes es el *Taller de Consenso* el cual se trata de un método de planeamiento creativo, útil para crear planes de acción, tomar decisiones y generar opciones tanto en situaciones normales como de conflicto [9]. El objetivo del sistema a desarrollar es desempeñar el papel de facilitador para mejorar el proceso de toma de decisiones de los grupos de investigadores. El sistema consta de una serie de etapas adaptadas del *Taller de Consenso* descritas en la Tabla 1.1.

1.7.4 Elicitación de preferencias

A medida que los usuarios interactúan con sistemas o aplicaciones, se generan registros que contienen información relevante. Estos registros se suelen utilizar para fines de auditoría, pero también tienen aplicaciones prácticas y útiles para los propios usuarios. Los

Etapa	Descripción
Contextualizar	El facilitador comunica el propósito de la actividad que se llevará a cabo y se asegura de que todos los miembros del grupo entiendan el contexto de los ítems recomendados.
Lluvia de ideas	En esta etapa, el facilitador se encarga de garantizar que todos los miembros del grupo comprendan plenamente la situación del problema.
Resolver	El último paso permite que los miembros del grupo reflexionen, valoren y, sobre todo, lleguen a un consenso final.

Tabla 1.1: Etapas para el *Taller de Consenso*.

sistemas pueden capturar estas interacciones y utilizar la información recopilada para recomendar objetos de preferencia a los usuarios.

En el contexto de los sistemas de recomendación, el proceso de elicitación de preferencias tiene como objetivo principal permitir que los usuarios finales revelen sus preferencias a medida que interactúan con el sistema a lo largo del tiempo [10]. Sin embargo, en el caso de los sistemas de recomendación grupal, este proceso se aborda de manera colectiva durante la fase de consenso, en contraste con los sistemas de recomendación individuales donde las preferencias se determinan de manera individual [11].

Las preferencias de usuario pueden ser tanto implícitas como explícitas pero en los SRG, se tiene la tarea adicional de combinar las preferencias individuales de los miembros del grupo en un modelo grupal de preferencias conjuntas para generar recomendaciones apropiadas [3].

1.7.5 Herramientas utilizadas

A continuación, en la Tabla 1.2 se proporcionará un detalle de las herramientas utilizadas.

Herramienta	Descripción
React Native	Es un marco para aplicaciones móviles de código abierto creado por Facebook. Se usa para desarrollar aplicaciones multiplataforma y proporciona controles de interfaz de usuario nativa y acceso completo a la plataforma nativa [12].
Javascript	Es un lenguaje de programación o de secuencias de comandos que permite implementar funciones complejas en páginas web [13].
Expo	Es un marco de trabajo de código abierto para aplicaciones que se ejecutan de forma nativa en Android, iOS y la Web expo .
Node	Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. [14].
Git	Git es un sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. [15].
Firebase	Firebase es una plataforma de desarrollo de apps que ayuda a compilar y desarrollar apps y juegos. [16].
Fontawesome	Fontawesome es un conjunto de iconos ampliamente utilizado que brinda imágenes vectoriales escalables que se pueden personalizar con CSS [17].
Cucumber	Cucumber es una herramienta de BDD (<i>Behavior Driven Development</i>) que permite el desarrollo de casos de prueba basados en escenarios de comportamiento [18] .
Jest	Jest es un framework de pruebas que permite asegurar la correctitud del código generado en Javascript y Typescript [19].
Typescript	TypeScript es un lenguaje de programación de código abierto, pensado para ejecutar grandes proyectos y que permite trabajar de manera estructurada. [20].

Tabla 1.2: Herramientas utilizadas.

2 METODOLOGÍA

En este capítulo se pretende explicar la metodología que se utilizará para llevar a cabo el proyecto de tesis, con el objetivo de asegurar una entrega de alta calidad y fácil mantenimiento del producto desarrollado.

2.1 INTRODUCCIÓN A LA METODOLOGÍA DE DESARROLLO ÁGIL

El enfoque adoptado para este proyecto de tesis será ágil, lo que implica una gestión dinámica y flexible del trabajo del equipo de desarrollo. Se seguirá la filosofía del Manifiesto Ágil, que establece cuatro valores y diecisiete principios para lograr la agilidad en el desarrollo de proyectos [21]. Los cuatro valores del Manifiesto Ágil se presentan en la Figura 2.1.



Figura 2.1: Valores del manifiesto ágil. Elaborado por el autor.

Entonces, el proyecto de tesis seguirá un enfoque basado en agilidad como se mencionó anteriormente, esta decisión se tomó porque los requerimientos del sistema tienen características mutables que se las realizarán frecuentemente a lo largo del desarrollo del sistema. Es por esta razón que todo el ciclo de vida de desarrollo de software para el sistema se lo ejecutará según la orientación de la metodología BDD [22].

BDD es una metodología que se centra en el comportamiento esperado del software a construirse y no la manera en la que debe ser desarrollada la lógica del sistema. Además, se considera que BDD es una mejora de TDD [23] que involucra la construcción de pruebas antes de programar el sistema y refactorizar el código antes de terminar la solución del sistema. Sin embargo, TDD se centra en las pruebas unitarias y la diferencia es que BDD se centra en pruebas de más alto nivel como la funcionalidad, aceptación y la pregunta fundamental de *¿Cuál es el problema para definir?* mas no la solución de antemano [22]. La Figura 2.2 muestra el flujo de desarrollo que se utilizará bajo la metodología BDD.

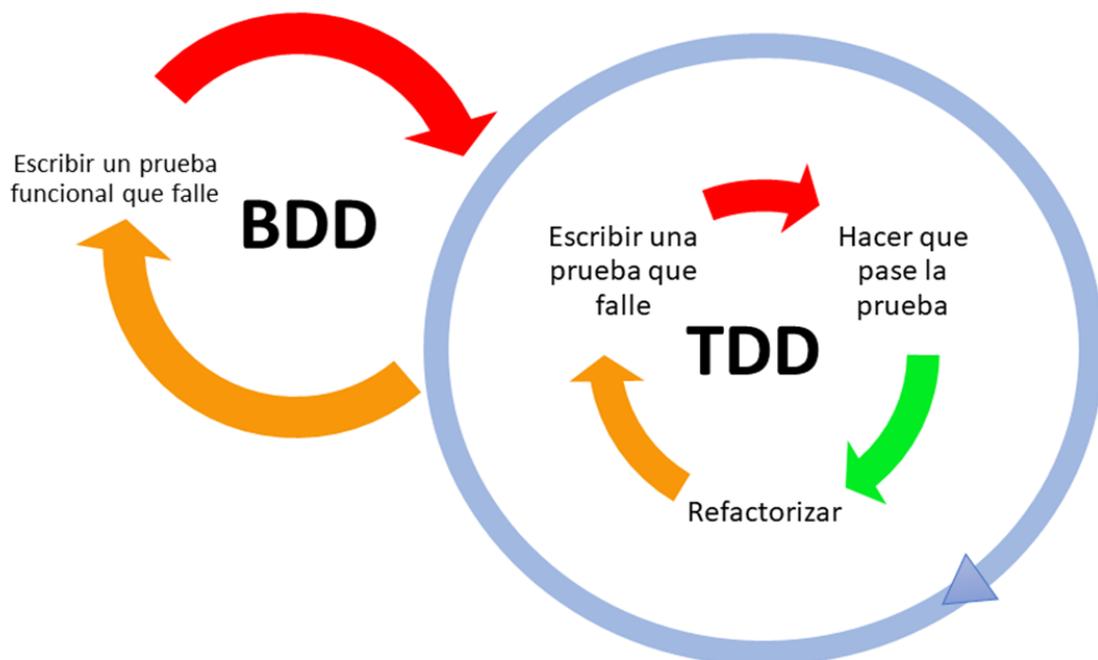


Figura 2.2: Desarrollo dirigido por el comportamiento. Elaborado por el autor.

La integración de BDD y Scrum se basa en la idea de que el BDD puede ayudar a mejorar la colaboración y la comunicación entre los miembros del equipo de Scrum, lo que a su vez puede mejorar la calidad del software y acelerar su desarrollo. En lugar de depender de una documentación detallada y especializada, el equipo de Scrum puede utilizar la descripción del comportamiento deseado escrita en un lenguaje común para guiar el desarrollo del

software en cada sprint, como se puede observar en la Figura 2.3. Esto puede ayudar a asegurar que todos los miembros del equipo estén en la misma página y que se entiendan mutuamente las necesidades del negocio y los requisitos del sistema.

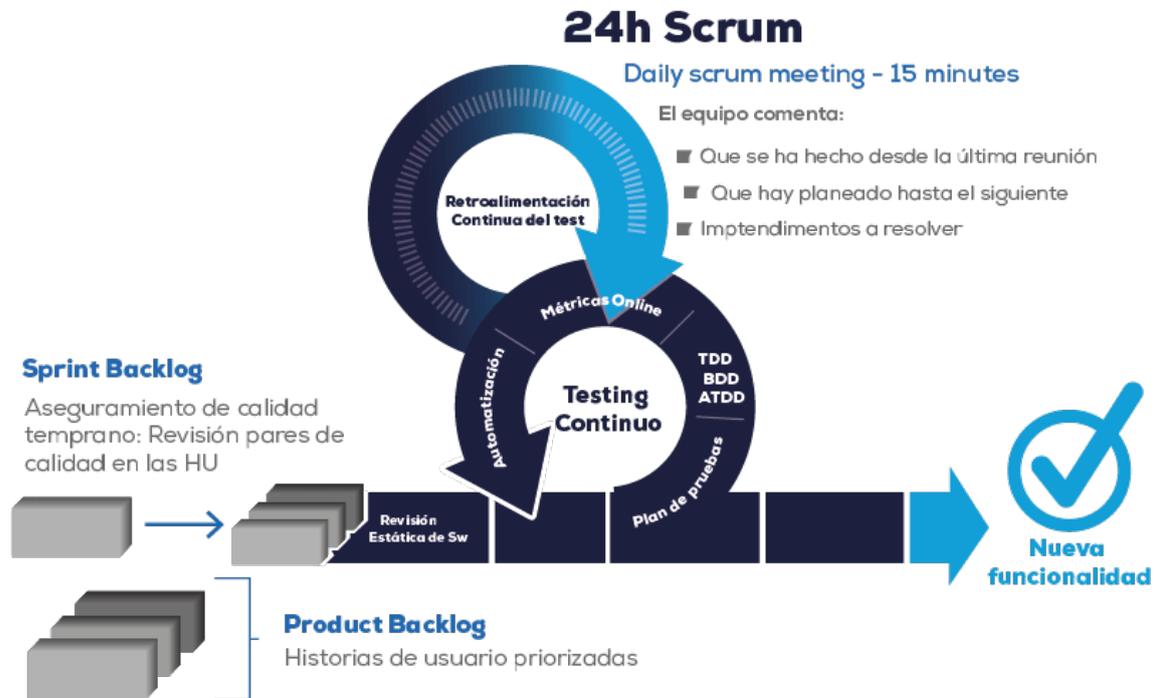


Figura 2.3: Agile Testing [24].

2.2 CICLO DE DESARROLLO ITERATIVO DEL SOFTWARE

El ciclo de desarrollo de software que se seguirá para la construcción del módulo A del sistema consta de 8 fases correspondientes a BDD (*Behavior Driven Development*) y Scrum. Las 7 fases de BDD y Scrum se dividirán en sprints como se muestra en la Tabla 2.1.

2.3 ROLES

El flujo de trabajo que se realizará para el desarrollo del proyecto consta de los roles básicos de Scrum. El primer rol es el dueño del producto (*Product Owner*) quien es el responsable de la identificación de las funcionalidades del producto del software, y la continua priorización de la lista de productos entregables a realizar [4]. El segundo rol es el de equipo de desarrollo (*Development Team*), que es el encargado de desarrollar el producto de software

Id	Nombre	Descripción
S0	Identificación y definición de requisitos	En este sprint se busca identificar y definir los requisitos funcionales, no funcionales, y las especificaciones técnicas del módulo del sistema.
S1	Diseño de escenarios	En este sprint se crean los escenarios que describen el comportamiento esperado del módulo del sistema enfocado en la perspectiva del usuario.
S2	Implementación de escenarios	En este sprint se busca convertir los escenarios diseñados en pruebas automatizadas.
S3	Desarrollo guiado por pruebas	En este sprint se busca desarrollar las pruebas unitarias correspondientes a lo que se espera que realice el código del módulo del sistema.
S4	Desarrollo del módulo	En este sprint se realiza la construcción del código correspondiente al módulo del sistema, tomando en cuenta que deben pasar las pruebas desarrolladas.
S5	Ejecución de las pruebas automatizadas	En este sprint se ejecuta las pruebas automatizadas correspondientes a los escenarios.
S6	Verificación y validación	En este sprint se valida que el comportamiento y el funcionamiento del sistema cumpla con los requisitos. Además se verifica que los resultados de las pruebas automatizadas sean satisfactorios.

Tabla 2.1: Sprints.

[4], es así que para este caso habrá un equipo de desarrollo enfocado en la construcción de los componentes de registro y autenticación de usuarios, creación y manejo grupos de investigación y notificación de la actividad de grupo, para el Sistema de Toma de Decisiones y Consenso Grupal Aplicado a los Motores de Recomendación, correspondientes al módulo A. Se debe recordar que en el marco de trabajo Scrum los integrantes son multidisciplinarios y el equipo es auto-gestionado, es decir, el equipo se encargará de comprometerse a construir el producto de software con responsabilidad y autonomía. Finalmente el tercer rol es el experto en Scrum (*Scrum Master*), quien se encargará de guiar a todo el equipo para implementar correctamente el marco de trabajo Scrum [4]. En la Tabla 2.2 se encuentra definida una lista de los encargados de cumplir con los roles del equipo Scrum.

Rol	Responsable
Product Owner	Lorena Recalde
Scrum Master	Lorena Recalde
Developer	Daniel Aimacaña

Tabla 2.2: Roles y responsables del equipo Scrum.

2.4 REQUISITOS Y ESPECIFICACIONES

Los requisitos del módulo del sistema se obtuvieron mediante varias entrevistas realizadas al *Product Owner* y en conjunto con el *Scrum Master*. Es así que, en la Tabla 2.3 se muestra una lista de los requerimientos obtenidos resultado de las entrevistas.

Id	Requisito	Complejidad
01	Definir la arquitectura del aplicativo móvil y web	Muy alta
02	Diseñar un prototipo de interfaz y experiencia de usuario	Alta
03	Establecer la estructura y funcionalidad del aplicativo móvil y web	Alta
04	Identificar la integración con servicios externos	Media
05	Implementar el aplicativo para móvil y web	Muy alta
06	Probar la funcionalidad del aplicativo móvil y web	Alta

Tabla 2.3: Requerimientos del módulo del sistema.

2.5 PRODUCT BACKLOG

El product backlog es una lista de trabajo ordenado por prioridades para el equipo de desarrollo que se obtiene de los requisitos [25]. Para el presente trabajo se utilizará los ítems mencionados a continuación:

- ❑ **Epic:** Es un conjunto de *features* que abarcan varios *releases*.
- ❑ **Feature:** Es un conjunto de *user stories* que representan un valor comercial para las partes interesadas durante y a lo largo de los sprints.
- ❑ **User stories:** Son las necesidades del trabajo que detallan con claridad lo que los desarrolladores deben implementar.

Cada uno de los ítems se encontrarán ordenados como se muestra en la Figura 2.4. Además, la nomenclatura de identidad que se utilizará tanto para los *epics*, *features* y *user stories* se lo realizará como se lo especifica a continuación:

- ❑ **EP[Id]:** Identificador de las épicas del sistema.
- ❑ **F[Id]:** Identificador de las *features* de las *epics*.

□ **US[Id]**: identificador de las *user stories* de cada una de las *features*.

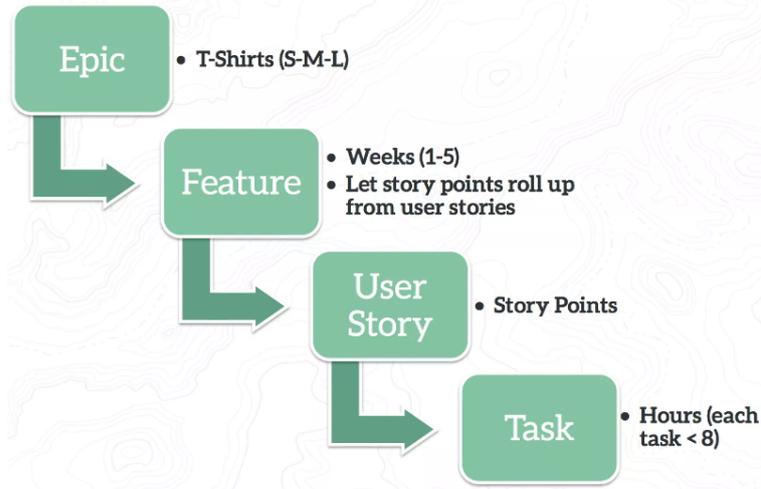


Figura 2.4: Jerarquía de los ítems del product backlog [26].

En la Tabla 2.4 se encuentran establecidas las *epics* correspondientes a los cuatro componentes, así como dos consideraciones importantes en relación con el desarrollo de las funcionalidades. Estas consideraciones abarcan la preparación del entorno de desarrollo y la validación del producto final del software en los módulos *B* y *C*.

Id (Epic)	Épica	Descripción
EP1	Levantamiento del ambiente de desarrollo y pruebas	En esta épica se pretende instalar, configurar y estabilizar el ambiente de trabajo donde se desarrollarán los componentes del módulo.
EP2	Presentación de tópicos de investigación	Esta épica se enfoca en desarrollar la funcionalidad que permitirá presentar las recomendaciones de tópicos de investigación generadas por el sistema a los usuarios del grupo de investigación.
EP3	Integración con Google Scholar	En esta épica se llevará a cabo la integración del sistema con la API de Google Scholar, lo que permitirá a los usuarios consultar directamente información relevante sobre los tópicos de investigación recomendados.
EP4	Ranking de tópicos	Esta épica se centra en la implementación de un mecanismo de evaluación y ordenamiento de los tópicos recomendados.
EP5	Presentación de tópicos aceptados y rechazados	Esta épica se encargará de mostrar los resultados de las decisiones tomadas por los usuarios del grupo de investigación con respecto a los tópicos recomendados.
EP6	Validación del producto de software final	El propósito de esta épica es asegurar la validación de que los requisitos del proyecto se han cumplido en la implementación del producto de software final.

Tabla 2.4: Épicas.

Además, en la Tabla 2.5 se han establecido las *features* relacionadas con cada épica.

Estas *features* representan las funcionalidades principales de los módulos *B* y *C*, las cuales se subdividen en diversas historias de usuario, tal como se detalla en la Tabla 2.7. Estas historias de usuario encapsulan los requisitos funcionales del módulo y se describen desde la perspectiva de las necesidades de los usuarios.

Id	Feature
EP1F1	Especificación de la arquitectura del módulo.
EP1F2	Configuración del repositorio para el proyecto.
EP1F3	Configuración del entorno de desarrollo local.
EP1F4	Configuración del entorno de pruebas automatizadas.
EP2F1	Generación de tópicos de investigación.
EP2F2	Presentación de tópicos de investigación.
EP3F1	Búsqueda de información de tópicos.
EP4F1	Reordenamiento de tópicos.
EP4F2	Evaluación de los tópicos.
EP5F1	Presentación de tópicos aceptados y rechazados.
EP6F1	Pruebas exhaustivas del producto final para cumplir con los requisitos.

Tabla 2.5: Características del sistema.

Por último, en la Figura 2.5 se llevó a cabo la priorización y estimación del esfuerzo de desarrollo para cada una de las historias de usuario identificadas. La priorización de estas historias de usuario se realizó utilizando una plantilla conocida como *Business Value x Urgency* [27], en la que las historias de usuario que se encuentran en la parte superior de cada columna tienen un mayor valor para el negocio, mientras que las historias que se ubican en la parte izquierda de la Figura 2.5 poseen una mayor urgencia para su desarrollo. Además, cada historia de usuario está etiquetada con un valor que indica el esfuerzo requerido para satisfacer las necesidades del usuario.

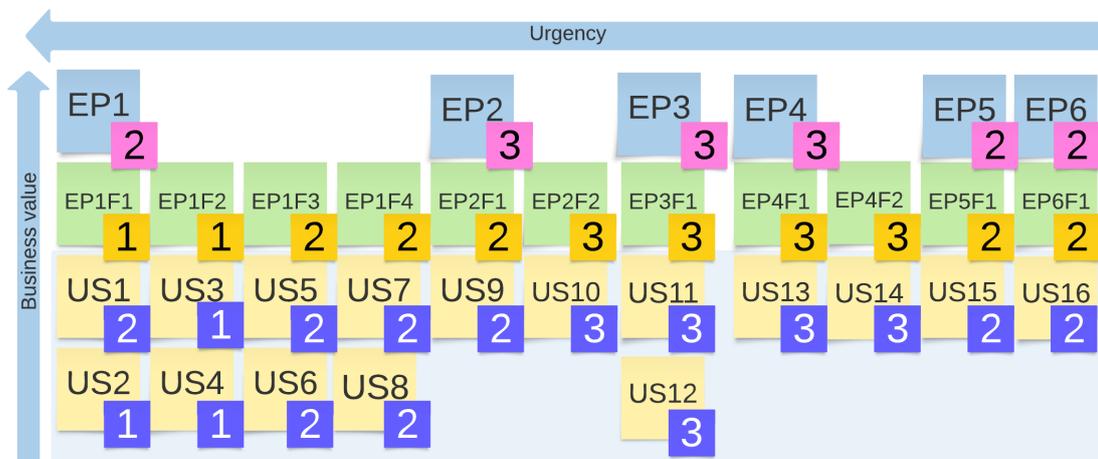


Figura 2.5: Product backlog. Elaborado por el autor.

Id	Id	Historias de usuario
EP1F1	US1	Como desarrollador, quiero definir la arquitectura general de los módulos, identificando los componentes principales y su interacción, para tener una guía clara sobre cómo se estructurará el sistema.
	US2	Como miembro del equipo, quiero documentar la arquitectura de los módulos, describiendo los patrones de diseño y las tecnologías utilizadas, para que otros miembros puedan comprender fácilmente la estructura y colaborar eficientemente.
EP1F2	US3	Como desarrollador, quiero crear un repositorio en una plataforma de control de versiones, configurando la estructura básica del proyecto, para asegurar un control adecuado del código fuente y colaboración en equipo.
	US4	Como miembro del equipo, quiero definir políticas y estándares de ramas en el repositorio, para asegurar una gestión ordenada de las ramas de desarrollo y la implementación de características.
EP1F3	US5	Como desarrollador, deseo identificar todas las herramientas de software y hardware necesarios para desarrollar los módulos.
	US6	Como desarrollador, deseo configurar un entorno de desarrollo estable para ejecutar el código del módulo sin problemas.
EP1F4	US7	Como miembro del equipo de desarrollo, deseo establecer una configuración de pruebas automatizadas en el entorno de desarrollo, permitiendo la ejecución de pruebas unitarias y de integración, como parte del proceso de desarrollo de los módulos.
	US8	Como miembro del equipo de desarrollo, quiero crear un conjunto exhaustivo de pruebas unitarias que abarque todas las funcionalidades de los módulos, con el objetivo de verificar y confirmar que el código funcione correctamente y cumpla con los requisitos establecidos.
EP2F1	US9	Como usuario, deseo que el sistema genere automáticamente tópicos de investigación precalculados, para obtener una lista inicial de temas relevantes.
EP2F2	US10	Como usuario, quiero ver una lista de los tópicos de investigación generados, ordenados de acuerdo a su relevancia.
EP3F1	US11	Como usuario, quiero poder hacer clic en cada tópico de investigación en el sistema y ser redirigido hacia Google Scholar, para obtener más detalles sobre el tópico.
	US12	Como usuario, quiero poder seleccionar varios tópicos de investigación y ser redirigido hacia Google Scholar, para obtener más detalles sobre los tópicos seleccionados.
EP4F1	US13	Como usuario, quiero poder arrastrar y soltar los tópicos de investigación en un orden de prioridad deseado y guardarlo, para facilitar el proceso de ranking y consenso grupal.
EP4F2	US14	Como usuario, quiero tener la opción de calificar cada tópico de investigación en función de su relevancia y potencial, para ayudar en el proceso de toma de decisiones.
EP5F1	US15	Como usuario miembro de un grupo, deseo poder visualizar una lista de tópicos aceptados y rechazados, junto con sus respectivas justificaciones, para tener una visión clara de los tópicos que serán considerados y descartados.
EP6F1	US16	Como miembro del equipo de calidad, quiero diseñar y ejecutar pruebas de funcionalidad para verificar que todas las características principales del producto final se comporten según los requisitos SUS.

Tabla 2.7: Historias de usuarios.

2.6 PLANIFICACIÓN DE LOS *SPRINTS*

Se determinó la duración y las tareas a realizar en cada *sprint* para la planificación. La Tabla 2.8 presenta las historias de usuario asignadas a cada *sprint*, junto con su esfuerzo estimado y la duración en días de trabajo.

No. Sprint	S0	S1	S2	S3	S4	S5	S6
US(ID)	US1, US2, US3, US4, US5, US6, US7	US8, US9, US10	US8, US11, US12	US8, US13	US8, US14	US15, U16	US16
Tiempo estimado	20 días	7 días	14 días	14 días	14 días	14 días	7 días

Tabla 2.8: *Sprint planning*.

2.7 SPRINT 0

2.7.1 Objetivos del sprint

- Establecer la arquitectura técnica de los módulos *B* y *C*.
- Definir pautas y normas para el código, la gestión de versiones y la documentación.
- Identificar los recursos necesarios para el desarrollo.
- Configurar el entorno de desarrollo y las pruebas automatizadas.
- Describir los requisitos esenciales del sistema mediante la definición de los escenarios de comportamiento del módulo *B* y *C*.

2.7.2 Ejecución del sprint

El presente *sprint* se realizó en base a la planificación (*Sección 2.6*) y a las historias de usuario correspondientes al *Sprint 0* de la Tabla 2.8.

2.7.2.1 Arquitectura de los módulos

La arquitectura de los módulos está dividida en 2 secciones: el *Back-End* y el *Front-End* (Figura 2.6). El *Back-End* usa los servicios de almacenamiento y autenticación de *Firebase* y el *Front-End* esta basado en el uso del framework *React Native* que permite la construcción de una aplicación multiplataforma. La comunicación entre el *Back-End* y el *Front-End* es transparente dado que únicamente se usa un modelo de dependencias que contiene todas las funcionalidades de *Firebase* y sus servicios listos para usar.

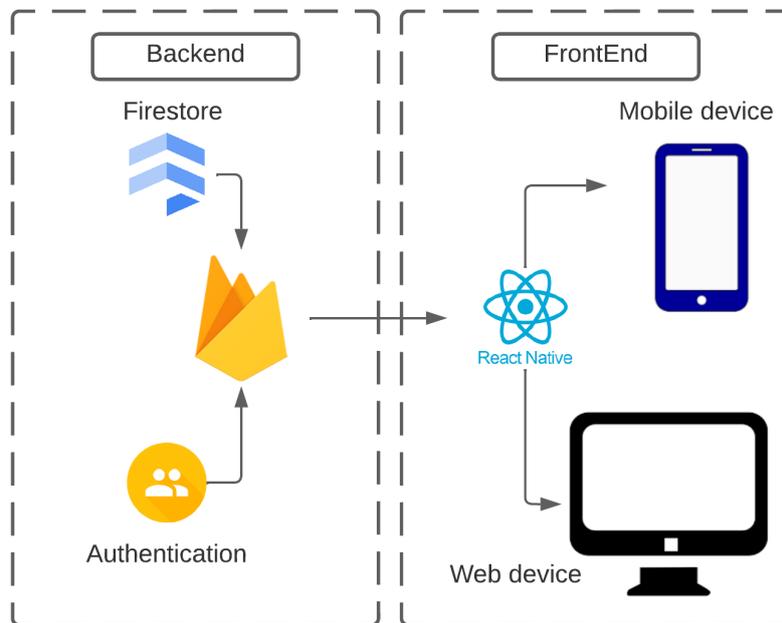


Figura 2.6: Arquitectura técnica de los módulos *B* y *C*. Elaborado por el Autor.

2.7.2.2 Normas, estándares y convenciones

Con el fin asegurar una documentación y un código legible, mantenible y correcto se especificó en la Tabla 2.9 las convenciones y estándares de codificación que se usarán en el desarrollo de los módulos *B* y *C*.

Herramienta	Estándar o convención	Descripción
Javascript y Typescript	Douglas Crockford	En la convención de Douglas Crockford se especifica el formato para nombrar archivos javascript, espacios en blanco, comentarios, declaración de variables, declaración de funciones, nombres y declaraciones [28].
Archivos y documentos	CamelCase	CamelCase es un estilo de escritura donde cada palabra debe tener su primera letra en mayúscula [29].
Git	Git commit workflow, nomenclatura de sucursales	Workflow y nomenclatura de sucursales tienen convenciones de los formatos para realizar <i>commits</i> al repositorio, nombrar ramas y una guía de trabajo para usar git correctamente [30].
React Native	<i>Feature-Based</i>	<i>Feature-Based</i> es la guía de cómo organizar el código de una aplicación de <i>React</i> basándose en sus características o módulos [31].

Tabla 2.9: Convenciones y estándares de desarrollo y documentación.

2.7.2.3 Recursos

Para llevar a cabo el desarrollo del presente trabajo, se procedió a identificar los recursos disponibles, tanto en el ámbito de software como de hardware. Los detalles específicos de estos recursos se encuentran en la Tabla 2.11.

Tipo	Recurso	Descripción
Hardware	Laptop	ASUS VivoBook 15
	Memoria RAM	Memoria física de 8.0 GB (7.88 GB usable)
	Procesador	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1800 MHz
	Almacenamiento	Disco sólido local de 256 GB
Software	Sistema Operativo	Windows 11 Home
	Base de datos	Firebase Firestore
	Lenguaje de programación	Typescript
	Sistema de control de versiones	Git y Github
	Servicios en la nube	Firebase

Tabla 2.11: Recursos de software y hardware.

2.7.2.4 Configuración del entorno de desarrollo y pruebas automatizadas

Para comenzar, se procedió a la creación de un repositorio en la plataforma de control de versiones *GitHub*. Este repositorio se identificó con el nombre *SITDECG*^[1] (Sistema de Toma de Decisiones y Consenso Grupal), en consonancia con el contexto del proyecto. Asimismo, se estableció el nombre comercial para la aplicación web y móvil como *ReSearch-Decide*, una designación que refleja directamente el propósito del aplicativo.

^[1] <https://github.com/SITDECG/ReSearchDecide-v1.0.0.git>

Además, se procedió a la configuración del entorno de desarrollo, siguiendo las directrices del lenguaje de programación mencionado en la Sección 2.7.2.3. En consecuencia, se optó por emplear el *framework* de desarrollo *React Native* para la implementación de la parte del Front-End. La Figura 2.7 exhibe el entorno de desarrollo activado y muestra el resultado de la primera compilación realizada en dicho entorno.



Figura 2.7: *React native* levantado. Elaborado por el Autor.

Adicionalmente, en cuanto a la implementación del Back-End, se tomó la decisión de emplear *Firebase*, que es una plataforma que ofrece diversos servicios de almacenamiento y autenticación en la nube. Para la lógica del negocio, se optó por aprovechar el esquema de colecciones proporcionado por *Firestore*. En la Figura 2.8, se presenta el esquema de almacenamiento que se utilizará como base para desarrollar las funcionalidades del proyecto.

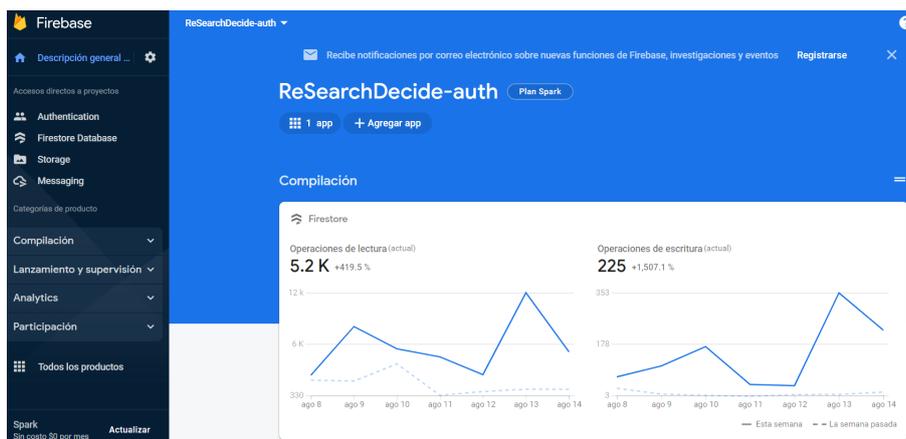
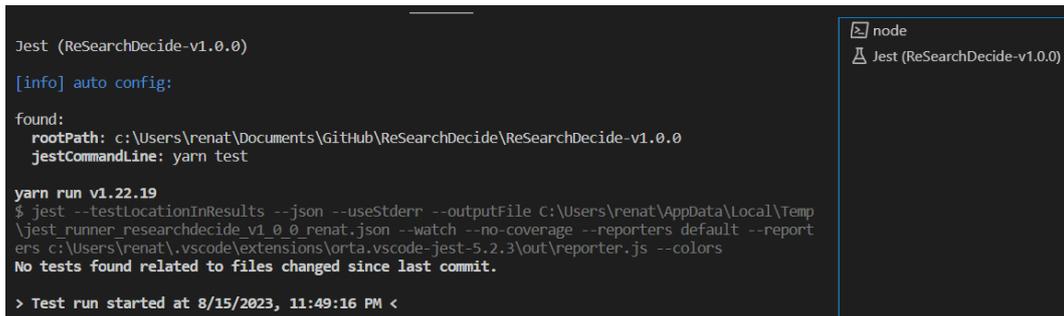


Figura 2.8: Inicio de *Firebase*. Elaborado por el Autor.

Para el entorno de pruebas, se tomó la decisión de seleccionar y configurar el *framework* de pruebas *Jest-Cucumber*, el cual posibilita la realización de pruebas automatizadas basadas en los escenarios de casos de uso del módulo, tal como lo experimentaría el usuario. En la

Figura 2.9 se presenta una representación visual del entorno de pruebas configurado.



```
Jest (ReSearchDecide-v1.0.0)
[info] auto config:
found:
  rootPath: c:\Users\renat\Documents\GitHub\ReSearchDecide\ReSearchDecide-v1.0.0
  jestCommandLine: yarn test
yarn run v1.22.19
$ jest --testLocationInResults --json --useStderr --outputFile C:\Users\renat\AppData\Local\Temp\jest_runner_researchdecide_v1_0_0_renat.json --watch --no-coverage --reporters default --reporters c:\Users\renat\.vscode\extensions\orta.vscode-jest-5.2.3\out\reporter.js --colors
No tests found related to files changed since last commit.
> Test run started at 8/15/2023, 11:49:16 PM <
```

Figura 2.9: Jest levantado. Elaborado por el Autor.

2.7.2.5 Especificación de escenarios

Con el objetivo de garantizar que la construcción de los módulos *B* y *C* satisfaga las necesidades y expectativas de los usuarios, la metodología BDD (*Behavior Driven Development*) propone la utilización de escenarios como una técnica para recopilar los requerimientos del usuario en relación al comportamiento deseado del módulo en desarrollo. Con esta premisa, se llevó a cabo una serie de entrevistas y se aplicaron técnicas de maquetación de interfaces de software para obtener un conjunto de 20 escenarios de comportamiento junto con sus respectivas interfaces^[2]. Estos escenarios servirán como base para la posterior etapa de desarrollo del módulo.

Los escenarios fueron identificados y agrupados en dos *features* principales del comportamiento de los módulos. El primer *feature* se centra en la obtención y actualización de miembros dentro de la aplicación, lo que dio lugar a la obtención de los siguientes escenarios:

```
1 Feature: Members
-
- Scenario: Getting a member with valid userId and id
-   Given a member with userId "validUserId" and id "validId"
5   When the function getMemberByUserId is called
-   Then the result should be a Member object
-
```

^[2] <https://shorturl.at/lqF58>

```
- Scenario: Getting a member with invalid userId
-   Given a member with userId "invalidUserId" and id "validId"
10  When the function getMemberByUserId is called
-   Then the result should be null
-
- Scenario: Getting a member with invalid id
-   Given a member with userId "validUserId" and id "invalidId"
15  When the function getMemberByUserId is called
-   Then the result should be null
-
- Scenario: Updating member's vote with valid uid and id
-   Given a member with uid "validUid" and id "validId"
20  When the function updateMemberVote is called with new vote true
-   Then the member's vote should be updated to true
-
- Scenario: Updating member's vote with invalid uid
-   Given a member with uid "invalidUid" and id "validId"
25  When the function updateMemberVote is called with new vote false
-   Then the member's vote should not be updated
-
- Scenario: Updating member's vote with invalid id
-   Given a member with uid "validUid" and id "invalidId"
30  When the function updateMemberVote is called with new vote true
-   Then the member's vote should not be updated
```

En las Figuras 2.12 y 2.13 se muestran los *mockups* resultado de los escenarios de comportamiento del *feature* "Member" definido y aprobado por el *Product Owner*.

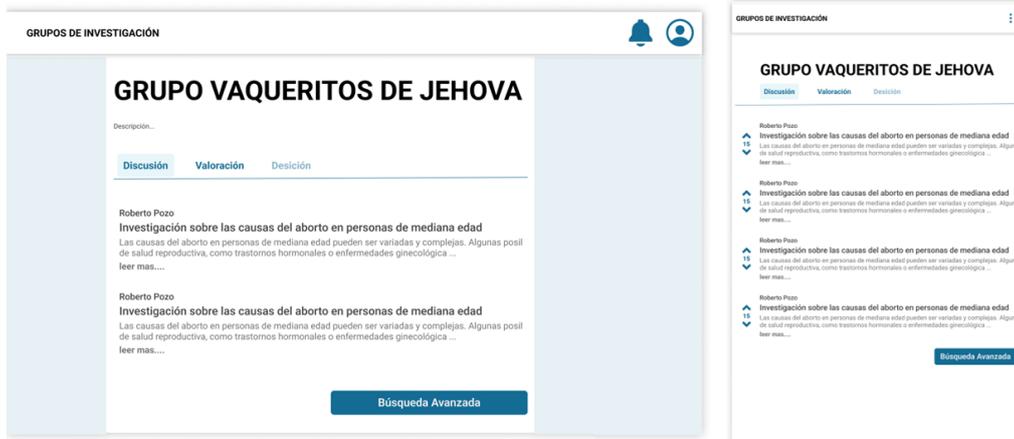


Figura 2.10: Obtención de votación de miembro. Elaborado por el Autor.



Figura 2.11: Actualización de votación de miembro. Elaborado por el Autor.

Por otro lado, el segundo *feature* se enfoca en la obtención y actualización de tópicos dentro de la aplicación, lo que conllevó a la obtención de los siguientes escenarios:

```

1 Feature: Topics
-
- Scenario: Getting topics when topics are available
- Given there are topics available
5 When the function getTopics is called

```

```

-   Then it should return a list of topics
-
-   Scenario: Getting topics when no topics are available
-   Given there are no topics available
10  When the function getTopics is called
-   Then it should return an empty list
-
-   Scenario: Getting a list of topic scores ordered by score ascending
-   Given there are topic scores available
15  When the function getTopicsScore is called
-   Then it should return a list of topic scores ordered by score
      ascending
-
-   Scenario: Updating the score of an existing topic
-   Given there is a topic with title "Topic 1" and score 10
20  When the function updateTopicScoreByTopic is called with title "
      Topic 1" and new score 5
-   Then the score of the topic "Topic 1" should be updated to 15
-
-   Scenario: Updating boolean properties of an existing topic
-   Given there is a topic with title "Topic 1" and properties
      attractive=true, novel=false
25  When the function updateBooleanProperties is called with title "
      Topic 1" and properties to update attractive=false, trend=true
-   Then the properties attractive and novel of the topic "Topic 1"
      should be updated to false and true, respectively

```

En las Figuras 2.12 y 2.13 se muestran los *mockups* resultado de los escenarios de comportamiento del *feature "Topics"* definido y aprobado por el *Product Owner*.

2.7.3 Revisión y retrospectiva del sprint

Se llevó a cabo la revisión y la retrospectiva correspondiente al *sprint 0* en línea con las historias de usuario previamente planificadas y el periodo de tiempo estipulado. Como resultado, los objetivos planteados para dicho Sprint se lograron exitosamente.

- ❑ Se estableció la arquitectura que guiará el desarrollo de los módulos *B* y *C*.



Figura 2.12: Obtención de tópicos recomendados. Elaborado por el Autor.



Figura 2.13: Actualización de tópicos recomendados. Elaborado por el Autor.

- Se definió convenciones y estándares para el código, el versionamiento y la documentación.
- Se identificó los recursos necesarios para la fase de desarrollo.
- Se procedió a la configuración del entorno de desarrollo.
- Se definió y especificó los escenarios de comportamiento para los módulos B y C.

2.8 SPRINT 1

2.8.1 Objetivos del sprint

- ❑ Definir las pruebas unitarias y de integración para la obtención y ordenamiento de tópicos de los módulos *B* y *C*.
- ❑ Desarrollar el código para el Front-end y el Back-end para la obtención de tópicos.
- ❑ Desarrollar el código para el Front-end y el Back-end para ordenar los tópicos generados de acuerdo a su relevancia.
- ❑ Verificar el funcionamiento de la obtención y ordenamiento de tópicos.

2.8.2 Ejecución del sprint

El presente *sprint* se realizó en base a la planificación (*Sección 2.6*) y a las historias de usuario correspondientes al *Sprint 1* de la Tabla 2.8.

2.8.2.1 Definición de pruebas

Durante este *sprint*, se emplearon los escenarios de comportamiento ilustrados en la Figura 2.14 como base para la definición de las pruebas unitarias. Dichas pruebas unitarias se presentan detalladamente en las Figuras 2.15 y 2.16.

```
Scenario: Getting topics when topics are available
  Given there are topics available
  When the function getTopics is called
  Then it should return a list of topics

Scenario: Getting a list of topic scores ordered by score ascending
  Given there are topic scores available
  When the function getTopicsScore is called
  Then it should return a list of topic scores ordered by score ascending
```

Figura 2.14: Escenarios de comportamiento de la obtención y ordenamiento de tópicos. Elaborado por el Autor.

```

test('Getting topics when topics are available', ({ given, when, then }) => {
  let getStub: sinon.SinonStub;

  given('there are topics available', () => {
    getStub = sinon.stub().resolves({
      docs: [
        { data: () => ({ id: 'topic1', topic: 'Topic 1' }) },
        { data: () => ({ id: 'topic2', topic: 'Topic 2' }) },
      ],
    });
    getCollectionStub.returns({
      get: getStub,
    });
  });

  when('the function getTopics is called', async () => {
    await getTopics();
  });

  then('it should return a list of topics', () => {
    expect(getStub.calledOnce).to.be.true;
  });
});

```

Figura 2.15: Prueba de la obtención de tópicos. Elaborado por el Autor.

```

test('Getting a list of topic scores ordered by score ascending', ({ given, when, then }) => {
  given('there are topic scores available', () => {
  });

  when('the function getTopicsScore is called', async () => {
    await getTopicsScore();
  });

  then('it should return a list of topic scores ordered by score ascending', () => {
    expect(orderByStub.calledOnceWith('score', 'asc')).to.be.true;
  });
});

beforeEach(() => {
  updateStub = sinon.stub().resolves();
  getStub = sinon.stub().resolves({
    docs: [{ data: () => ({ id: 'topic1', topic: 'Topic 1', score: 10 }) }],
  });
  whereStub = sinon.stub().returns({
    get: getStub,
  });
  getCollectionStub = sinon.stub().returns({
    where: whereStub,
  });
  sinon.stub(require('firebase/compat/app'), 'firestore').returns({
    collection: getCollectionStub,
  });
});

```

Figura 2.16: Prueba del ordenamiento de tópicos. Elaborado por el Autor.

2.8.2.2 Desarrollo

A partir de las pruebas previamente definidas, se procedió a la codificación de las funciones necesarias para cumplir con dichas pruebas. En este proceso, fue necesario establecer el modelo de la base de datos (como se muestra en la Figura 2.17), implementar las interfaces de usuario (representadas en las Figuras 2.19 y 2.20, y desarrollar los servicios (ilustrados en la Figura 2.18) con el fin de asegurar la correcta operación de las funcionalidades propuestas.

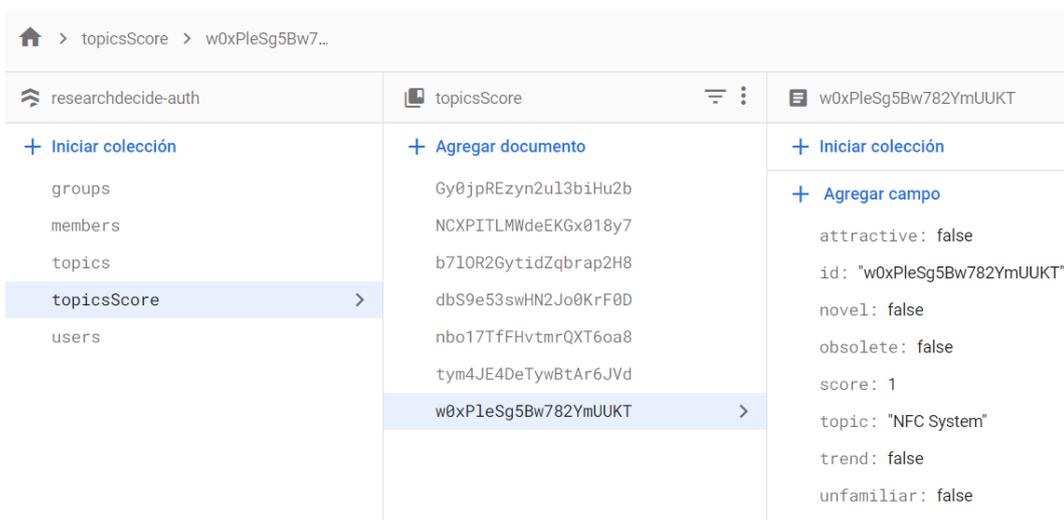


Figura 2.17: Colecciones para la obtención de tópicos. Elaborado por el autor.

```
export const getTopicsScore = async (): Promise<TopicScore[]> => {
  try {
    const auxTopics: TopicScore[] = [];
    const snapshot = await firebase.firestore().collection('topicsScore').orderBy("score", "asc").get();
    const topics = snapshot.docs.map((doc) => doc.data());
    topics.forEach((topic) => {
      const auxTopic: TopicScore = {
        id: topic.id,
        topic: topic.topic,
        score: topic.score,
        attractive: topic.attractive,
        novel: topic.novel,
        trend: topic.trend,
        obsolete: topic.obsolete,
        unfamiliar: topic.unfamiliar,
      };
      auxTopics.push(auxTopic);
    });
    return auxTopics;
  } catch (error) {
    console.log(error);
    return [];
  }
};
```

Figura 2.18: Servicios de obtención y ordenamiento de tópicos. Elaborado por el autor.

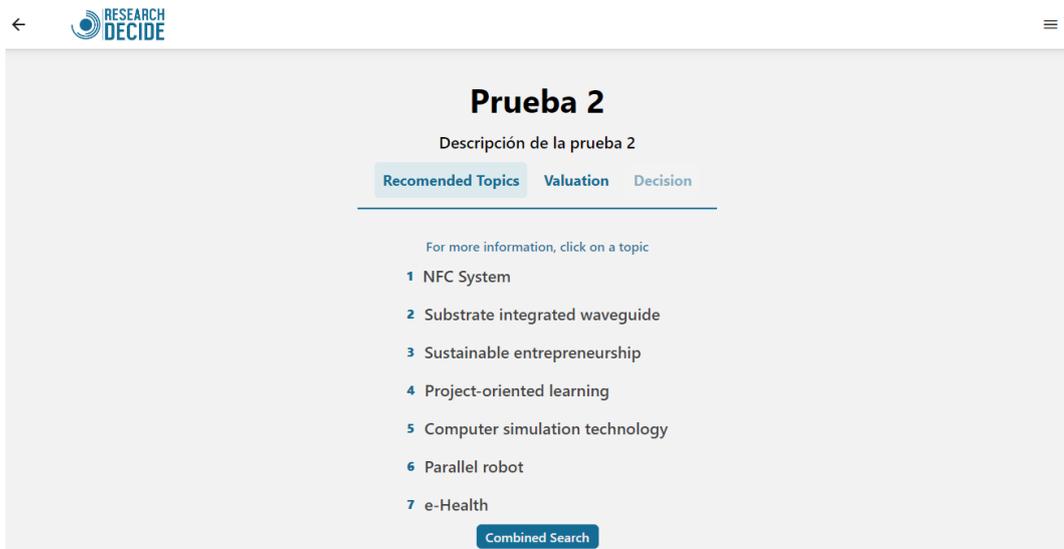


Figura 2.19: Interfaz de obtención y ordenamiento de tópicos web. Elaborado por el autor.

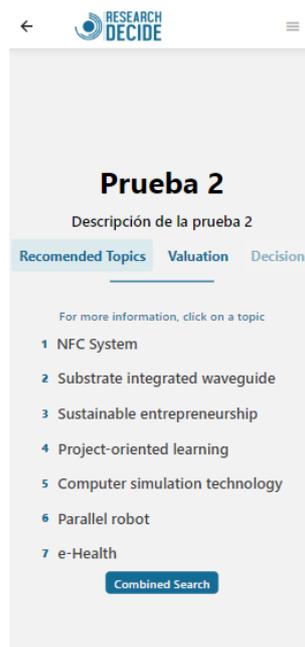


Figura 2.20: Interfaz de obtención y ordenamiento de tópicos móvil. Elaborado por el autor.

2.8.2.3 Verificación del desarrollo

Finalmente, se procedió a ejecutar las pruebas que habían sido definidas previamente, con el objetivo de verificar que el desarrollo realizado cumple con los criterios de aceptación establecidos para las historias de usuario. En la Figura 2.21 se presenta el resultado de estas pruebas sobre el desarrollo implementado.

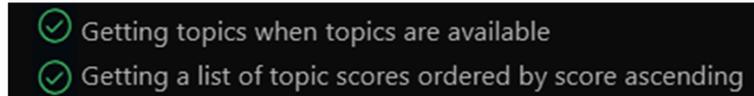


Figura 2.21: Resultado de la ejecución de las pruebas de las funciones de obtención y ordenamiento de tópicos. Elaborado por el autor.

2.8.3 Revisión y retrospectiva del sprint

Se llevó a cabo la revisión y la retrospectiva correspondiente al *sprint 1* en consonancia con las historias de usuario previamente planificadas y el periodo de tiempo establecido. Como resultado, los objetivos establecidos para dicho Sprint fueron logrados con éxito.

- Se definió las pruebas unitarias y de integración para la obtención de tópicos de los módulos *B* y *C*.
- Se desarrolló el código para el Front-end y el Back-end para la obtención de tópicos.
- Se desarrolló el código para el Front-end y el Back-end para ordenar los tópicos generados de acuerdo a su relevancia.
- Se verificó la obtención y ordenamiento de tópicos.

2.9 SPRINT 2

2.9.1 Objetivos del sprint

- Definir las pruebas unitarias y de integración para el redireccionamiento hacia Google Scholar y realizar una búsqueda simple y combinada.
- Desarrollar el código para el Front-end y el Back-end para realizar una búsqueda simple en Google Scholar.
- Desarrollar el código para el Front-end y el Back-end para realizar una búsqueda combinada en Google Scholar.
- Verificar el funcionamiento del redireccionamiento hacia Google Scholar.

2.9.2 Ejecución del sprint

El presente *sprint* se realizó en base a la planificación (Sección 2.6) y a las historias de usuario correspondientes al *Sprint 2* de la Tabla 2.8.

2.9.2.1 Definición de pruebas

Durante este *sprint*, se emplearon los escenarios de comportamiento ilustrados en la Figura 2.22 como base para la definición de las pruebas unitarias. Dichas pruebas unitarias se presentan detalladamente en las Figuras 2.23 y 2.24.

```
Scenario: Clicking on a research topic should redirect to Google Scholar
  Given a user is viewing a research topic "Topic 1"
  When the user clicks on the research topic
  Then the user should be redirected to Google Scholar with the search query "Topic 1"

Scenario: Clicking on multiple research topics should redirect to Google Scholar
  Given a user is viewing the research topics "Topic 1", "Topic 2", and "Topic 3"
  When the user selects the research topics "Topic 1" and "Topic 3"
  And the user clicks on the selected research topics
  Then the user should be redirected to Google Scholar with the search queries "Topic 1" and "Topic 3"
```

Figura 2.22: Escenarios de comportamiento del redireccionamiento hacia Google Scholar de forma simple y combinada. Elaborado por el Autor.

```
test('Clicking on a research topic should redirect to Google Scholar', ({ given, when, then }) => {
  let researchTopic: string;

  given('a user is viewing a research topic "Topic 1"', () => {
    researchTopic = 'Topic 1';
  });

  when('the user clicks on the research topic', () => {
    handleTitlePress(researchTopic);
  });

  then('the user should be redirected to Google Scholar with the search query "Topic 1"', () => {
    expect(openURLStub.calledOnceWith('https://scholar.google.com/scholar?q=Topic%201')).to.be.true;
  });
});
```

Figura 2.23: Prueba de búsqueda simple en Google Scholar. Elaborado por el Autor.

2.9.2.2 Desarrollo

A partir de las pruebas previamente definidas, se procedió a la codificación de las funciones necesarias para cumplir con dichas pruebas. En este proceso, fue necesario implementar las interfaces de usuario (representadas en las Figuras 2.25 y 2.26, y desarrollar funcionalidad (ilustrada en la Figura 2.27) con el fin de asegurar la correcta operación de las funcionalidades propuestas.

```

test('Clicking on multiple research topics should redirect to Google Scholar', ({ given, when, then }) => {
  let selectedTopics: string[];

  given('a user is viewing the research topics "Topic 1", "Topic 2", and "Topic 3"', () => {
    selectedTopics = ['Topic 1', 'Topic 2', 'Topic 3'];
  });

  when('the user selects the research topics "Topic 1" and "Topic 3"', () => {
    selectedTopics = ['Topic 1', 'Topic 3'];
  });

  when('the user clicks on the selected research topics', () => {
    selectedTopics.forEach((topic) => {
      handleTitlePress(topic);
    });
  });

  then('the user should be redirected to Google Scholar with the search queries "Topic 1" and "Topic 3"', () => {
    selectedTopics.forEach((topic) => {
      expect(openURLStub.calledWith(`https://scholar.google.com/scholar?q=${encodeURIComponent(topic)}`).to.be.true;
    });
  });
});

```

Figura 2.24: Prueba de búsqueda combinada en Google Scholar. Elaborado por el Autor.

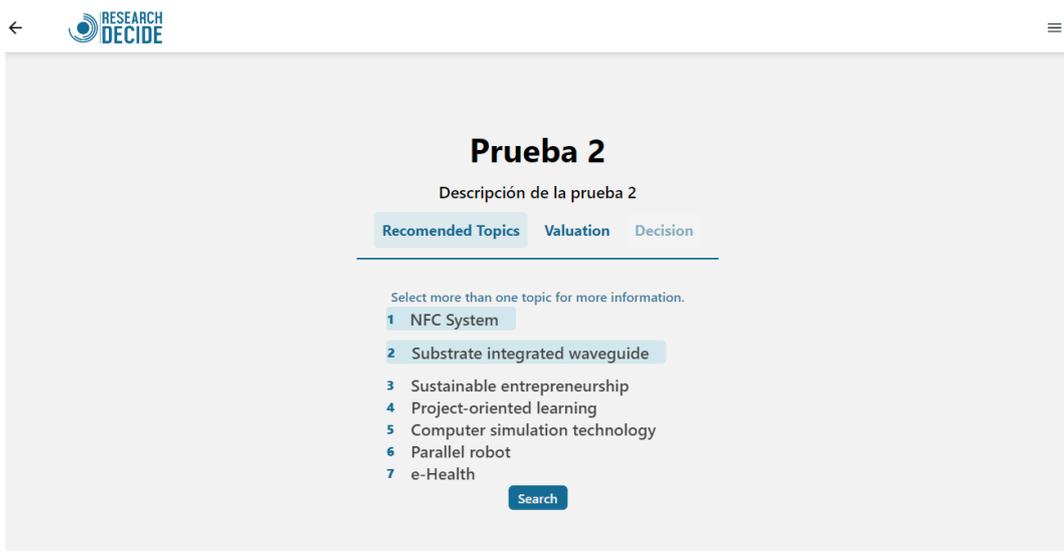


Figura 2.25: Interfaz de búsqueda en Google Scholar web. Elaborado por el autor.

2.9.2.3 Verificación del desarrollo

Finalmente, se procedió a ejecutar las pruebas que habían sido definidas previamente, con el objetivo de verificar que el desarrollo realizado cumple con los criterios de aceptación establecidos para las historias de usuario. En la Figura 2.28 se presenta el resultado de estas pruebas sobre el desarrollo implementado.

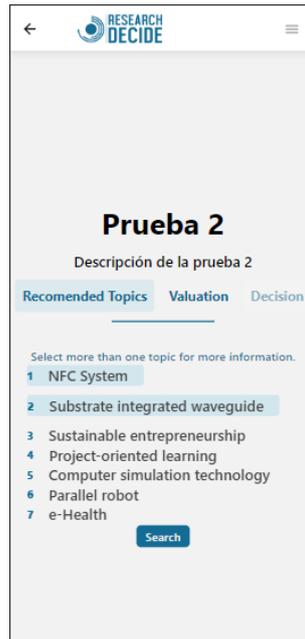


Figura 2.26: Interfaz de búsqueda en Google Scholar móvil. Elaborado por el autor.

```
const handleTitlePress = (text: string) => {
  const query = encodeURIComponent(text)
  const url = `https://scholar.google.com/scholar?q=${query}`
  Linking.openURL(url)
}

const handleSearch = () => {
  const selectedItems = data.filter((item) => item.checked);
  const query = selectedItems.map((item) => item.title).join('+');
  const url = `https://scholar.google.com/scholar?q=${query}`
  Linking.openURL(url)
};
```

Figura 2.27: Función de búsqueda en Google Scholar. Elaborado por el autor.

- ✓ Clicking on a research topic should redirect to Google Scholar
- ✓ Clicking on multiple research topics should redirect to Google Scholar

Figura 2.28: Resultado de las pruebas ejecutadas sobre las funciones búsqueda simple y combinada de tópicos. Elaborado por el autor.

2.9.3 Revisión y retrospectiva del sprint

Se llevó a cabo la revisión y la retrospectiva correspondiente al *sprint 2* en consonancia con las historias de usuario previamente planificadas y el periodo de tiempo establecido. Como resultado, los objetivos establecidos para dicho Sprint fueron logrados con éxito.

- ❑ Se definió las pruebas unitarias y de integración para el redireccionamiento hacia Google Scholar y realizar una búsqueda simple y combinada.
- ❑ Se desarrolló el código para el Front-end y el Back-end para realizar una búsqueda simple en Google Scholar.
- ❑ Se desarrolló el código para el Front-end y el Back-end para realizar una búsqueda combinada en Google Scholar.
- ❑ Se verificó el funcionamiento del redireccionamiento hacia Google Scholar.

2.10 SPRINT 3

2.10.1 Objetivos del sprint

- ❑ Definir las pruebas unitarias y de integración para el reordenamiento de tópicos mediante la funcionalidad de arrastrar y soltar, además guardar el nuevo orden.
- ❑ Desarrollar el código para el Front-end y el Back-end para el reordenamiento de tópicos mediante la funcionalidad de arrastrar y soltar.
- ❑ Desarrollar el código para el Front-end y el Back-end para guardar el nuevo orden de tópicos.
- ❑ Verificar el funcionamiento del reordenamiento de tópicos mediante la funcionalidad de arrastrar y soltar, así como asegurarse de guardar el nuevo orden.

2.10.2 Ejecución del sprint

El presente *sprint* se realizó en base a la planificación (*Sección 2.6*) y a las historias de usuario correspondientes al *Sprint 3* de la Tabla 2.8.

2.10.2.1 Definición de pruebas

Durante este *sprint*, se emplearon los escenarios de comportamiento ilustrados en las Figuras 2.29 y 2.30 como base para la definición de las pruebas unitarias. Dichas pruebas unitarias se presentan detalladamente en las Figuras 2.31 y 2.32.

```

Feature: Drag and Drop Research Topics

Scenario: User can reorder research topics using drag and drop
  Given a user is on the research topics ranking page
  When the user arranges the research topics in the desired order
  And the user saves the new order
  Then the research topics should be reordered as per user's preference

```

Figura 2.29: Escenario de comportamiento para el reordenamiento de tópicos. Elaborado por el Autor.

```

Scenario: Updating the score of an existing topic
  Given there is a topic with title "Topic 1" and score 10
  When the function updateTopicScoreByTopic is called with title "Topic 1" and new score 5
  Then the score of the topic "Topic 1" should be updated to 15

```

Figura 2.30: Escenario de comportamiento para el guardado de tópicos. Elaborado por el Autor.

```

test('User can reorder research topics using drag and drop', ({ given, when, and, then }) => {
  given('a user is on the research topics ranking page', () => {
    // Nada que hacer aquí, es la configuración inicial del escenario
  });

  when('the user arranges the research topics in the desired order', () => {
    // Simular el proceso de arrastrar y soltar los tópicos en el orden deseado
  });

  and('the user saves the new order', () => {
    // Simular el proceso de guardar el nuevo orden de prioridad
    reorderStub.returns(true); // Simular éxito al reordenar
  });

  then('the research topics should be reordered as per user\'s preference', () => {
    expect(reorderStub.calledOnce).to.be.true;
  });
});

```

Figura 2.31: Prueba de reordenamiento de tópicos. Elaborado por el Autor.

```

test('Updating the score of an existing topic', ({ given, when, then }) => {
  let currentScore = number;

  given('there is a topic with title "Topic 1" and score 10', () => {
    currentScore = 10;
  });

  when('the function updateTopicScoreByTopic is called with title "Topic 1" and new score 5', async () => {
    await updateTopicScoreByTopic('Topic 1', 5);
  });

  then('the score of the topic "Topic 1" should be updated to 15', async () => {
    expect(whenStub.calledOnceWith('topic', '=', 'Topic 1')).to.be.true;
    expect(updateStub.calledOnceWith({ score: currentScore + 5 })).to.be.true;
  });
});

```

Figura 2.32: Prueba de guardado de tópicos. Elaborado por el Autor.

2.10.2.2 Desarrollo

A partir de las pruebas previamente definidas, se procedió a la codificación de las funciones necesarias para cumplir con dichas pruebas. En este proceso, fue necesario implementar las interfaces de usuario (representadas en las Figuras 2.33 y 2.34, y desarrollar funcionalidad y el servicio (ilustrados en la Figuras 2.35 y 2.36) con el fin de asegurar la correcta

operación de las funcionalidades propuestas.

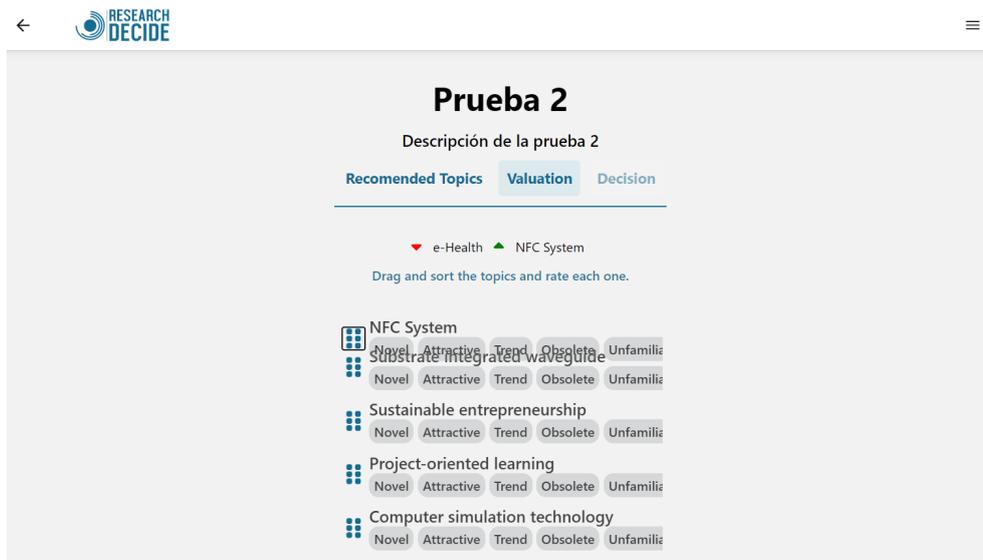


Figura 2.33: Interfaz de reordenamiento de tópicos web. Elaborado por el autor.

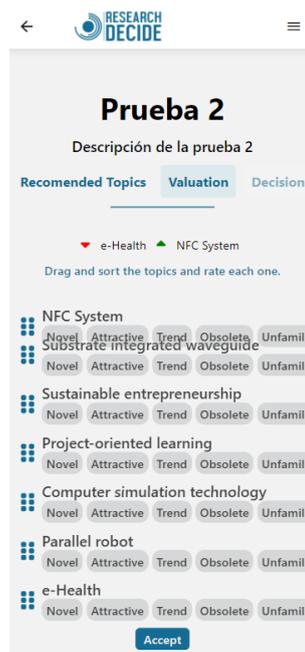


Figura 2.34: Interfaz de reordenamiento de tópicos móvil. Elaborado por el autor.

2.10.2.3 Verificación del desarrollo

Finalmente, se procedió a ejecutar las pruebas que habían sido definidas previamente, con el objetivo de verificar que el desarrollo realizado cumple con los criterios de aceptación establecidos para las historias de usuario. En la Figura 2.37 se presenta el resultado de estas pruebas sobre el desarrollo implementado.

```

function renderItem(info: DragListRenderItemInfo<string>) {
  const {item, onDragStart, isActive} = info;
  return (
    <TouchableOpacity
      key={item}
      onPressIn={onDragStart}
    >
    <ElementValuation title={item} />
    </TouchableOpacity>
  );
}

async function onReordered(fromIndex: number, toIndex: number) {
  const finalIndex = fromIndex < toIndex ? toIndex - 1 : toIndex;
  const copy = [...data]; // Don't modify react data in-place
  const removed = copy.splice(fromIndex, 1);

  copy.splice(finalIndex, 0, removed[0]); // Now insert at the new pos
  setData(copy);
}

```

Figura 2.35: Función de reordenamiento de tópicos. Elaborado por el autor.

```

export const updateTopicScoreByTopic = async (title: string, newScore: number): Promise<void> => {
  try {
    const topicRef = firebase.firestore().collection('topicsScore');
    const querySnapshot = await topicRef.where('topic', '==', title).get();

    if (!querySnapshot.empty) {
      const topicDoc = querySnapshot.docs[0];
      const currentScore = topicDoc.data().score || 0;
      const updatedScore = currentScore + newScore;
      await topicDoc.ref.update({ score: updatedScore });
    }
  } catch (error) {
    console.log('Error updating topic score:', error);
  }
};

```

Figura 2.36: Servicio de guardado de tópicos. Elaborado por el autor.

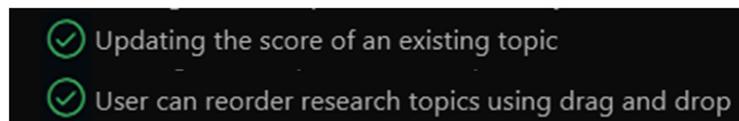


Figura 2.37: Pruebas ejecutadas del reordenamiento y guardado de tópicos. Elaborado por el autor.

2.10.3 Revisión y retrospectiva del sprint

Se llevó a cabo la revisión y la retrospectiva correspondiente al *sprint 3* en consonancia con las historias de usuario previamente planificadas y el periodo de tiempo establecido. Como resultado, los objetivos establecidos para dicho Sprint fueron logrados con éxito.

- ❑ Se definió las pruebas unitarias y de integración para el reordenamiento de tópicos mediante la funcionalidad de arrastrar y soltar, además guardar el nuevo orden.

- ❑ Se desarrolló el código para el Front-end y el Back-end para el reordenamiento de tópicos mediante la funcionalidad de arrastrar y soltar.
- ❑ Se desarrolló el código para el Front-end y el Back-end para guardar el nuevo orden de tópicos.
- ❑ Se verificó el funcionamiento del reordenamiento de tópicos mediante la funcionalidad de arrastrar y soltar, además de guardar el nuevo orden.

2.11 SPRINT 4

2.11.1 Objetivos del sprint

- ❑ Definir las pruebas unitarias y de integración para la calificación de tópicos.
- ❑ Desarrollar el código para el Front-end y el Back-end para la calificación de tópicos.
- ❑ Verificar el funcionamiento de la calificación de tópicos.

2.11.2 Ejecución del sprint

El presente *sprint* se realizó en base a la planificación (*Sección 2.6*) y a las historias de usuario correspondientes al *Sprint 4* de la Tabla 2.8.

2.11.2.1 Definición de pruebas

Durante este *sprint*, se emplearon los escenarios de comportamiento ilustrados en la Figura 2.38 como base para la definición de las pruebas unitarias. Dichas pruebas unitarias se presentan detalladamente en la Figura 2.39.

```
Scenario: Updating boolean properties of an existing topic
  Given there is a topic with title "Topic 1" and properties attractive=true, novel=false
  When the function updateBooleanProperties is called with title "Topic 1" and properties to update attractive=false, trend=true
  Then the properties attractive and novel of the topic "Topic 1" should be updated to false and true, respectively
```

Figura 2.38: Escenario de comportamiento para la calificación de tópicos. Elaborado por el Autor.

```

test('Updating boolean properties of an existing topic', ({ given, when, then }) => {
  let initialProperties: Partial<TopicScore>;

  given('there is a topic with title "Topic 1" and properties attractive=true, novel=false', () => {
    initialProperties = { attractive: true, novel: false };
  });

  when('the function updateBooleanProperties is called with title "Topic 1" and properties to update attractive=false, trend=true', async () => {
    await updateBooleanProperties('Topic 1', { attractive: false, trend: true });
  });

  then('the properties attractive and novel of the topic "Topic 1" should be updated to false and true, respectively', async () => {
    expect(wheresStub.calledOnceWith('topic', '=', 'Topic 1')).to.be.true;
    expect(updateStub.calledOnceWith({ attractive: false, trend: true })).to.be.true;
  });
});

```

Figura 2.39: Prueba de calificación de tópicos. Elaborado por el Autor.

2.11.2.2 Desarrollo

A partir de la prueba previamente definida, se procedió a la codificación de las funciones necesarias para cumplir con dichas pruebas. En este proceso, fue necesario implementar las interfaces de usuario (representadas en las Figuras 2.40 y 2.41, y desarrollar el servicio y la funcionalidad (ilustrados en la Figuras 2.43 y 2.42) con el fin de asegurar la correcta operación de las funcionalidades propuestas.

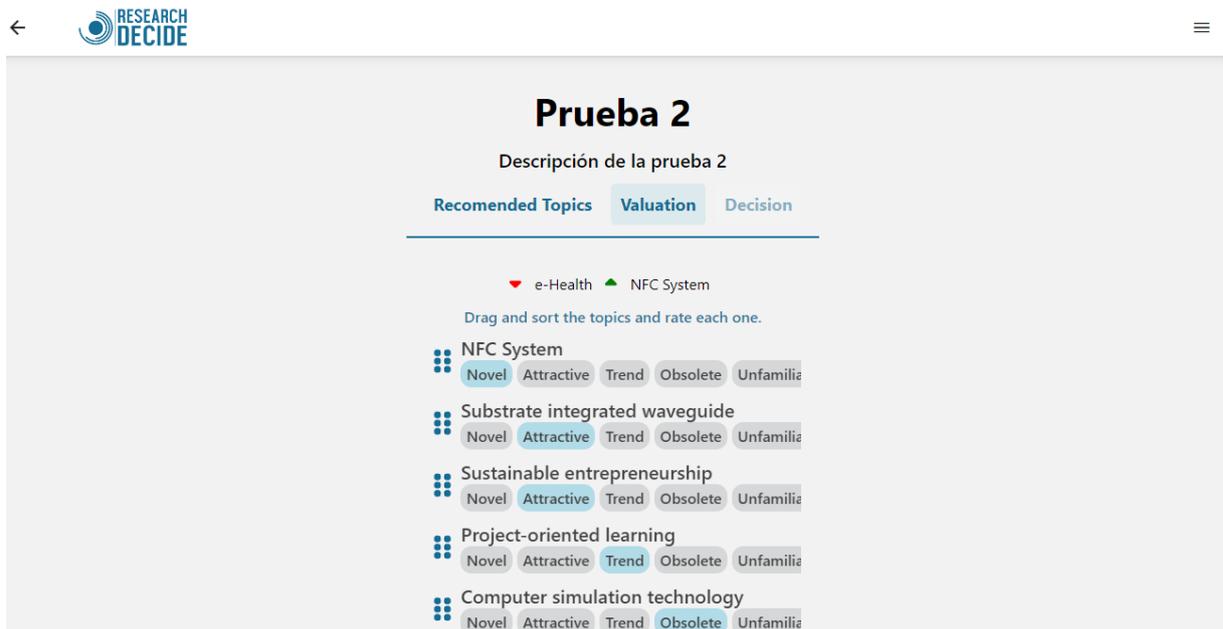


Figura 2.40: Interfaz de calificación de tópicos web. Elaborado por el autor.

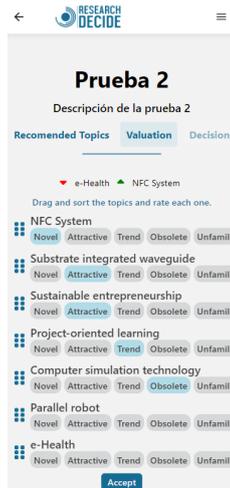


Figura 2.41: Interfaz de calificación de tópicos móvil. Elaborado por el autor.

```
export const updateBooleanProperties = async (title: string, propertiesToUpdate: Partial<TopicScore>): Promise<void> => {
  try {
    const topicRef = firebase.firestore().collection('topicsScore');
    const querySnapshot = await topicRef.where('topic', '==', title).get();

    if (!querySnapshot.empty) {
      const topicDoc = querySnapshot.docs[0];
      await topicDoc.ref.update(propertiesToUpdate);
    }
  } catch (error) {
    console.log('Error updating boolean properties:', error);
  }
};
```

Figura 2.42: Servicio de calificación de tópicos. Elaborado por el autor.

```
const { updateProperties } = useUpdateBooleanProperties();

const handleClick = (itemId: any) => {
  setItems((prevItems) =>
    prevItems.map((item) =>
      item.id === itemId.id ? { ...item, selected: !item.selected } : item
    )
  )
}

useEffect(() => {
  const selectedItem = items.find(item => item.selected);
  if (selectedItem) {
    const propertiesToUpdate: { [key: string]: boolean } = {};
    if (selectedItem.id === '1') {
      propertiesToUpdate.novel = selectedItem.selected;
    } else if (selectedItem.id === '2') {
      propertiesToUpdate.attractive = selectedItem.selected;
    } else if (selectedItem.id === '3') {
      propertiesToUpdate.trend = selectedItem.selected;
    } else if (selectedItem.id === '4') {
      propertiesToUpdate.obsolete = selectedItem.selected;
    } else if (selectedItem.id === '5') {
      propertiesToUpdate.unfamiliar = selectedItem.selected;
    }
    updateProperties(title, propertiesToUpdate);
  }
}, [items]);
```

Figura 2.43: Función de calificación de tópicos. Elaborado por el autor.

2.11.2.3 Verificación del desarrollo

Finalmente, se procedió a ejecutar las pruebas que habían sido definidas previamente, con el objetivo de verificar que el desarrollo realizado cumple con los criterios de aceptación establecidos para las historias de usuario. En la Figura 2.44 se presenta el resultado de estas pruebas sobre el desarrollo implementado.



Figura 2.44: Prueba ejecutada de la calificación de tópicos. Elaborado por el autor.

2.11.3 Revisión y retrospectiva del sprint

Se llevó a cabo la revisión y la retrospectiva correspondiente al *sprint 4* en consonancia con las historias de usuario previamente planificadas y el periodo de tiempo establecido. Como resultado, los objetivos establecidos para dicho Sprint fueron logrados con éxito.

- Se definió las pruebas unitarias y de integración para la calificación de tópicos.
- Se desarrolló el código para el Front-end y el Back-end para la calificación de tópicos.
- Se verificó el funcionamiento de la calificación de tópicos.

2.12 SPRINT 5

2.12.1 Objetivos del sprint

- Definir las pruebas unitarias y de integración para la obtención de tópicos aceptados y descartados, junto con sus justificaciones.
- Desarrollar el código para el Front-end y el Back-end para la obtención de tópicos aceptados y descartados, junto con sus justificaciones.
- Verificar el funcionamiento de la obtención de tópicos aceptados y descartados, junto con sus justificaciones.

2.12.2 Ejecución del sprint

El presente *sprint* se realizó en base a la planificación (*Sección 2.6*) y a las historias de usuario correspondientes al *Sprint 5* de la *Tabla 2.8*.

2.12.2.1 Definición de pruebas

Durante este *sprint*, se emplearon los escenarios de comportamiento ilustrados en la *Figura 2.45* como base para la definición de las pruebas unitarias. Dichas pruebas unitarias se presentan detalladamente en la *Figura 2.46*.

```
Scenario: Getting a list of topic scores ordered by score ascending
  Given there are topic scores available
  When the function getTopicsScore is called
  Then it should return a list of topic scores ordered by score ascending
```

Figura 2.45: Escenario de comportamiento para la obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el Autor.

```
test('Getting a list of topic scores ordered by score ascending', ({ given, when, then }) => {
  given('there are topic scores available', () => {
  });

  when('the function getTopicsScore is called', async () => {
    await getTopicsScore();
  });

  then('it should return a list of topic scores ordered by score ascending', () => {
    expect(orderByStub.calledOnceWith('score', 'asc')).to.be.true;
  });
});
```

Figura 2.46: Prueba de obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el Autor.

2.12.2.2 Desarrollo

A partir de la prueba previamente definida, se procedió a la codificación de las funciones necesarias para cumplir con dichas pruebas. En este proceso, fue necesario implementar las interfaces de usuario (representadas en las *Figuras 2.47* y *2.48*, y desarrollar el servicio y la funcionalidad (ilustrados en la *Figuras 2.49* y *2.50*) con el fin de asegurar la correcta operación de las funcionalidades propuestas.

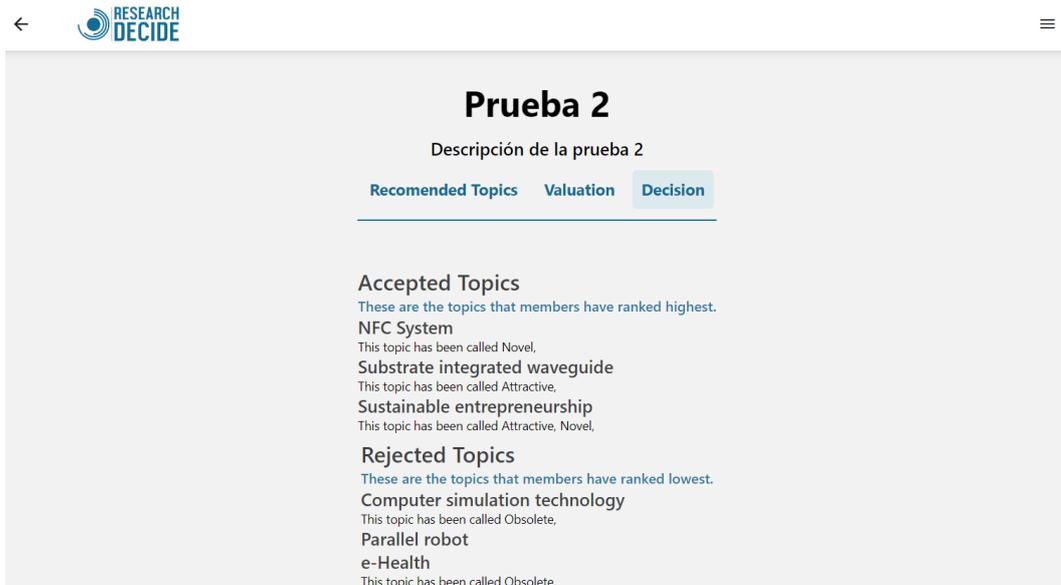


Figura 2.47: Interfaz de obtención de tópicos aceptados y descartados, junto con sus justificaciones web. Elaborado por el autor.

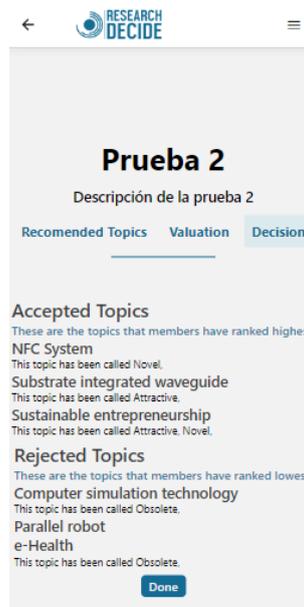


Figura 2.48: Interfaz de obtención de tópicos aceptados y descartados, junto con sus justificaciones móvil. Elaborado por el autor.

2.12.2.3 Verificación del desarrollo

Finalmente, se procedió a ejecutar las pruebas que habían sido definidas previamente, con el objetivo de verificar que el desarrollo realizado cumple con los criterios de aceptación establecidos para las historias de usuario. En la Figura 2.51 se presenta el resultado de estas pruebas sobre el desarrollo implementado.

```

export const getTopicsScore = async (): Promise<TopicScore[]> => {
  try {
    const auxTopics: TopicScore[] = [];
    const snapshot = await firebase.firestore().collection('topicsScore').orderBy("score","asc").get();
    const topics = snapshot.docs.map((doc) => doc.data());
    topics.forEach((topic) => {
      const auxTopic: TopicScore = {
        id: topic.id,
        topic: topic.topic,
        score: topic.score,
        attractive: topic.attractive,
        novel: topic.novel,
        trend: topic.trend,
        obsolete: topic.obsolete,
        unfamiliar: topic.unfamiliar,
      };
      auxTopics.push(auxTopic);
    });
    return auxTopics;
  } catch (error) {
    console.log(error);
    return [];
  }
};

```

Figura 2.49: Servicio de obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.

```

{topics.slice(0, 3).map((topic, index) => (
  <View key={topic.id}>
    <Text style={styles.titleText}>{topic.topic}</Text>
    <View style={styles.containerSectionB}>
      {topic.attractive || topic.novel || topic.trend || topic.obsolete || topic.unfamiliar ? (
        <Text>This topic has been called </Text>
      ) : null}
      {topic.attractive && <Text>Attractive, </Text>}
      {topic.novel && <Text>Novel, </Text>}
      {topic.trend && <Text>Trend, </Text>}
      {topic.obsolete && <Text>Obsolete, </Text>}
      {topic.unfamiliar && <Text>Unfamiliar</Text>}
    </View>
  </View>
))}

</View>
<View style={styles.containerSectionA}>
  <Text style={styles.title}>Rejected Topics</Text>
  <Text style={styles.text}>These are the topics that members have ranked lowest.</Text>
  {topics.slice(-3).map((topic, index) => (
    <View key={topic.id}>
      <Text style={styles.titleText}>{topic.topic}</Text>
      <View style={styles.containerSectionB}>
        {topic.attractive || topic.novel || topic.trend || topic.obsolete || topic.unfamiliar ? (
          <Text>This topic has been called </Text>
        ) : null}
        {topic.attractive && <Text>Attractive, </Text>}
        {topic.novel && <Text>Novel, </Text>}
        {topic.trend && <Text>Trend, </Text>}
        {topic.obsolete && <Text>Obsolete, </Text>}
        {topic.unfamiliar && <Text>Unfamiliar</Text>}
      </View>
    </View>
  ))}
</View>
))}

```

Figura 2.50: Función de presentación de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.

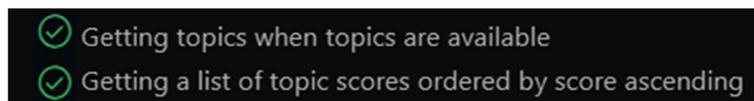


Figura 2.51: Prueba ejecutada de la obtención de tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.

2.12.3 Revisión y retrospectiva del sprint

Se llevó a cabo la revisión y la retrospectiva correspondiente al *sprint 5* en consonancia con las historias de usuario previamente planificadas y el periodo de tiempo establecido. Como resultado, los objetivos establecidos para dicho Sprint fueron logrados con éxito.

- ❑ Se definió las pruebas unitarias y de integración para la obtención de tópicos aceptados y descartados, junto con sus justificaciones.
- ❑ Se desarrolló el código para el Front-end y el Back-end para la obtención de tópicos aceptados y descartados, junto con sus justificaciones.
- ❑ Se verificó el funcionamiento de la obtención de tópicos aceptados y descartados, junto con sus justificaciones.

2.13 SPRINT 6

2.13.1 Objetivos del sprint

- ❑ Establecer pruebas de usabilidad con el objetivo de evaluar la adecuada funcionalidad de los módulos *B* y *C* que han sido desarrollados.

2.13.2 Ejecución del sprint

El presente *sprint* se realizó en base a la planificación (*Sección 2.6*) y a las historias de usuario correspondientes al *Sprint 6* de la Tabla 2.8.

2.13.2.1 Definición de pruebas

Para evaluar la usabilidad de los módulos *B* y *C*, se empleó el sistema de escalas de usabilidad (*SUS*), la cual es una herramienta eficaz y ágil que captura el nivel de aceptación de dichos módulos. Esto se logra mediante un cuestionario compuesto por diez preguntas, cada una con cinco opciones de respuesta, que varían desde *Totalmente en desacuerdo* hasta *Totalmente de acuerdo*.

Para llevar a cabo el cuestionario, se estableció varios aspectos fundamentales para lograr el objetivo de evaluar la usabilidad de los módulos *B* y *C*. Estos aspectos se presentan en la Tabla 2.12. El primer aspecto se centra en la identificación de los usuarios que participarán en la prueba del sistema. El segundo aspecto se relaciona con el tipo de pruebas que se llevarán a cabo, en este caso, *Pruebas de usabilidad moderadas*, en las cuales se pidió a los participantes que realizaran diversas tareas relacionadas con el módulo durante sesiones presenciales. Por último, se definió las métricas a ser evaluadas en relación al módulo.

Requisito	Descripción
Usuarios	Investigadores o estudiantes con interés en trabajos de investigación
Tipos de pruebas de usabilidad	Pruebas de usabilidad moderadas y guiadas.
Métricas de evaluación	Tiempo promedio para comprender e investigar los tópicos.
	Tiempo promedio para la toma de la decisión final sobre los tópicos.

Tabla 2.12: Criterios para la ejecución de las pruebas de usabilidad.

En la Tabla 2.13 se encuentran las 10 preguntas basadas en los principios de usabilidad de *Jakob Nielsen* [32]. Cada una de ellas debe ser respondida por parte de los usuarios seleccionados.

N°	Pregunta
1	¿Creo que me gustaría utilizar este sistema con frecuencia [32]?
2	¿Encontré el sistema innecesariamente complejo [32]?
3	¿Pensé que el sistema era fácil de usar [32]?
4	¿Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema [32]?
5	¿Encontré que las diversas funciones de este sistema estaban bien integradas [32]?
6	¿Pensé que había demasiada inconsistencia en este sistema [32]?
7	¿Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente [32]?
8	¿Encontré el sistema muy complicado de usar [32]?
9	¿Me sentí muy seguro usando el sistema [32]?
10	¿Necesitaba aprender muchas cosas antes de empezar con este sistema [32]?

Tabla 2.13: Preguntas de usabilidad [32].

2.13.2.2 Desarrollo

Basándose en las pruebas establecidas, se llevó a cabo las evaluaciones de usabilidad con la participación de 10 usuarios, entre investigadores y estudiantes con afinidad por la investigación. La Figura 2.52 proporciona evidencia visual de las sesiones realizadas durante las pruebas.

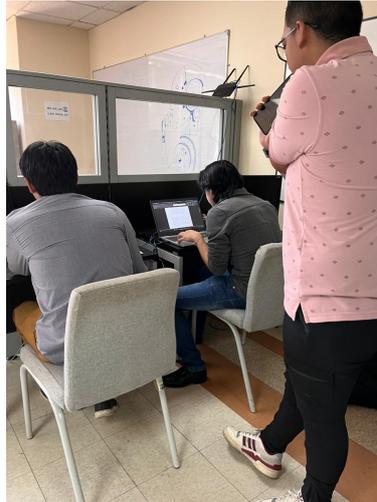


Figura 2.52: Sesiones presenciales de prueba con los usuarios. Elaborado por el autor.

2.13.3 Revisión y retrospectiva del sprint

Se llevó a cabo la revisión y la retrospectiva correspondiente al *sprint 6* en consonancia con las historias de usuario previamente planificadas y el periodo de tiempo establecido. Como resultado, los objetivos establecidos para dicho Sprint fueron logrados con éxito.

- ❑ Se estableció las pruebas de usabilidad para evaluar el correcto funcionamiento de los módulos *B* y *C* desarrollados.

3 RESULTADOS Y DISCUSIÓN

3.1 PRODUCTO FINAL

Los módulos *B* y *C* están compuestos por un total de 4 pantallas funcionales diseñadas para ser utilizadas en las plataformas web, iOS y Android. Estas 4 pantallas han sido desarrolladas de acuerdo a los requisitos específicos de los módulos, y a continuación se detalla en qué consisten cada una de ellas:

- Presentación y búsqueda simple de tópicos
- Búsqueda combinada de tópicos
- Reordenamiento y calificación de tópicos
- Presentación de tópicos aceptados y descartados, junto con sus justificaciones

3.1.1 Presentación y búsqueda simple de tópicos

La primera pantalla perteneciente a los módulos *B* y *C* es la de presentación de tópicos. Esta pantalla se vuelve accesible una vez que se ha creado el grupo correspondiente. La función principal de esta pantalla es permitir a los usuarios hacer clic sobre un tópico específico, lo que los redirigirá a Google Scholar para llevar a cabo una búsqueda relacionada con el tópico seleccionado como se puede observar en las Figuras 3.1 y 3.2.

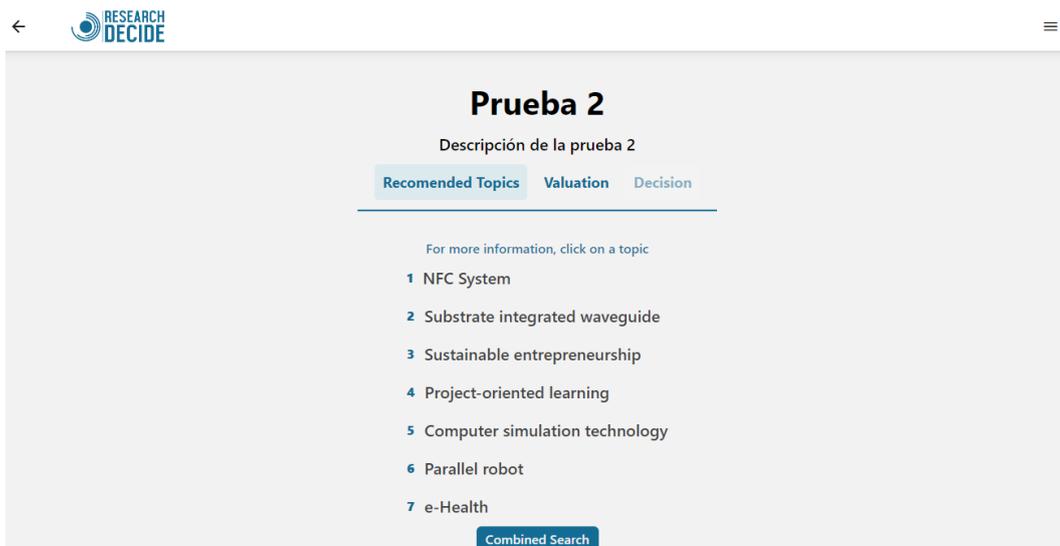


Figura 3.1: Presentación de tópicos. Elaborado por el autor.

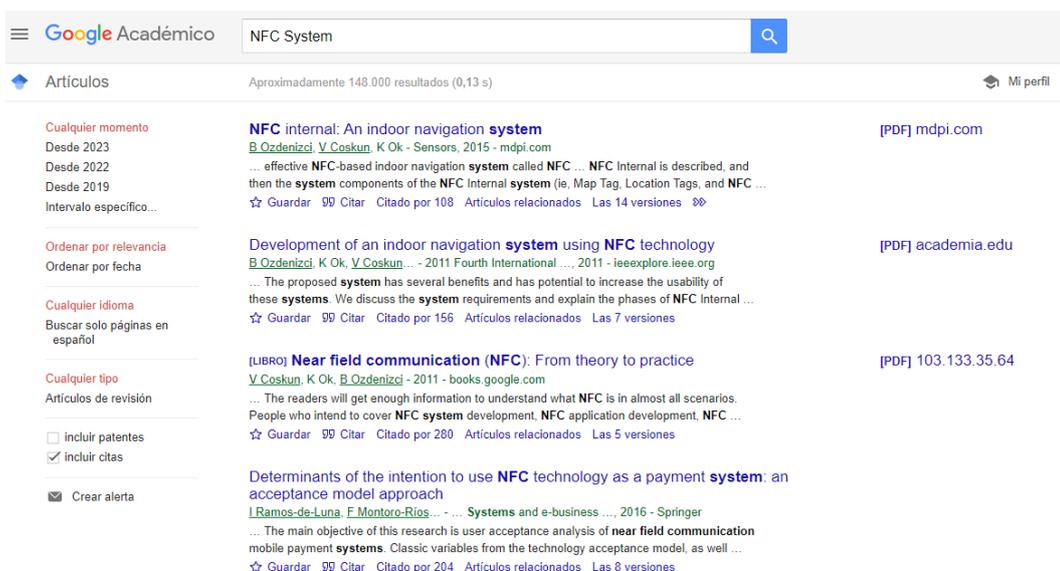


Figura 3.2: Búsqueda simple de tópicos. Elaborado por el autor.

3.1.2 Búsqueda combinada de tópicos

Para acceder a esta pantalla, es necesario hacer clic en el botón *Combined Search*, que es visible en la pantalla de presentación de tópicos (consulte la Sección 3.1.1). La función principal de esta pantalla es permitir a los usuarios seleccionar más de un tópico para realizar una búsqueda combinada en Google Scholar. Esto se ilustra en las Figuras 3.3 y 3.4.

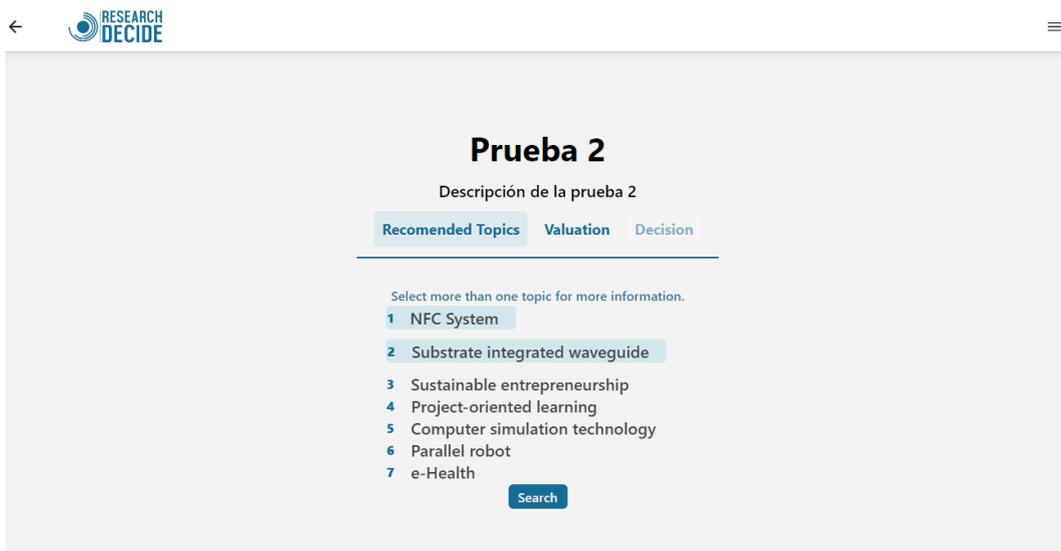


Figura 3.3: Selección de tópicos. Elaborado por el autor.

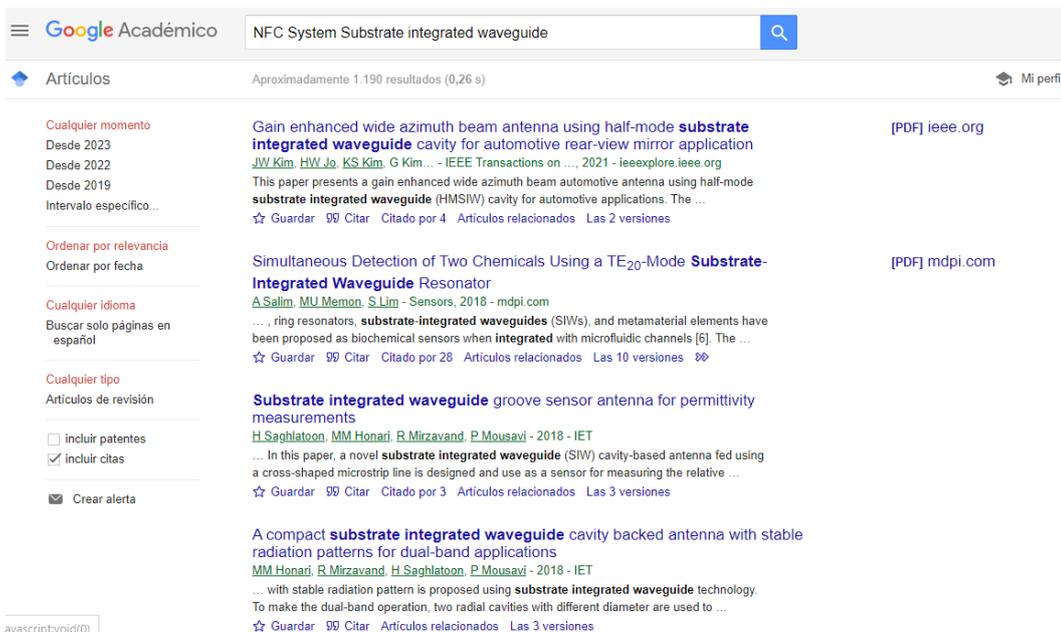


Figura 3.4: Búsqueda combinada de tópicos. Elaborado por el autor.

3.1.3 Reordenamiento y calificación de tópicos

La Figura 3.5 ilustra la pantalla de reordenamiento y calificación de tópicos. En esta pantalla, los usuarios tienen la capacidad de arrastrar los tópicos para cambiar su orden y también tienen la opción de asignar calificaciones a cada uno de los tópicos, según corresponda. Una vez que se ha realizado el reordenamiento y la calificación, es importante mencionar que los usuarios deben confirmar en el mensaje de advertencia que se presenta para guardar la votación, como se observa en la Figura 3.6. Es necesario tener en cuenta

que este proceso de confirmación es irreversible y no podrá ser deshecho.

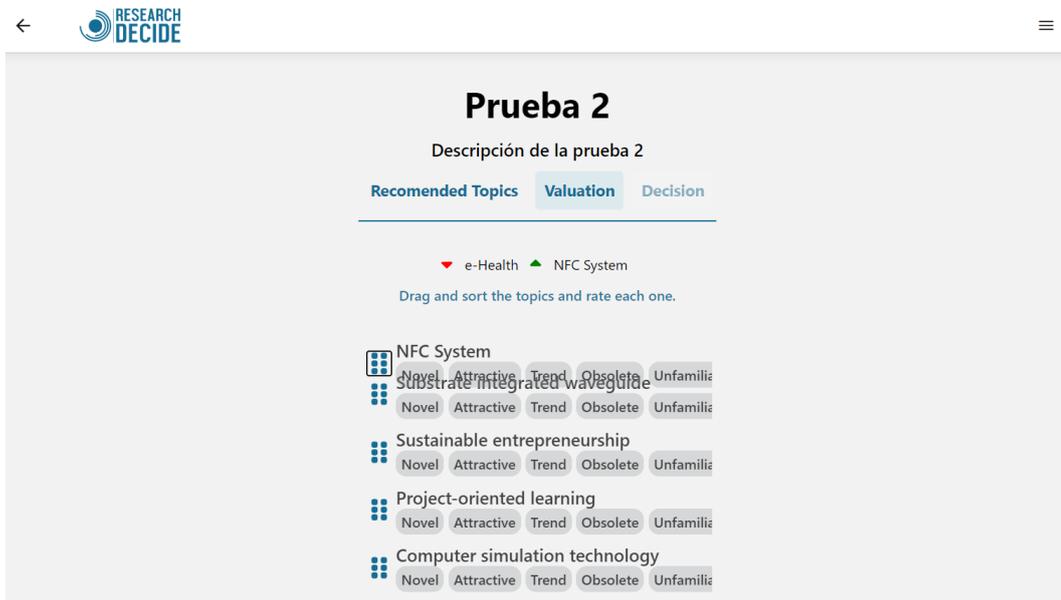


Figura 3.5: Reordenamiento y calificación de tópicos. Elaborado por el autor.

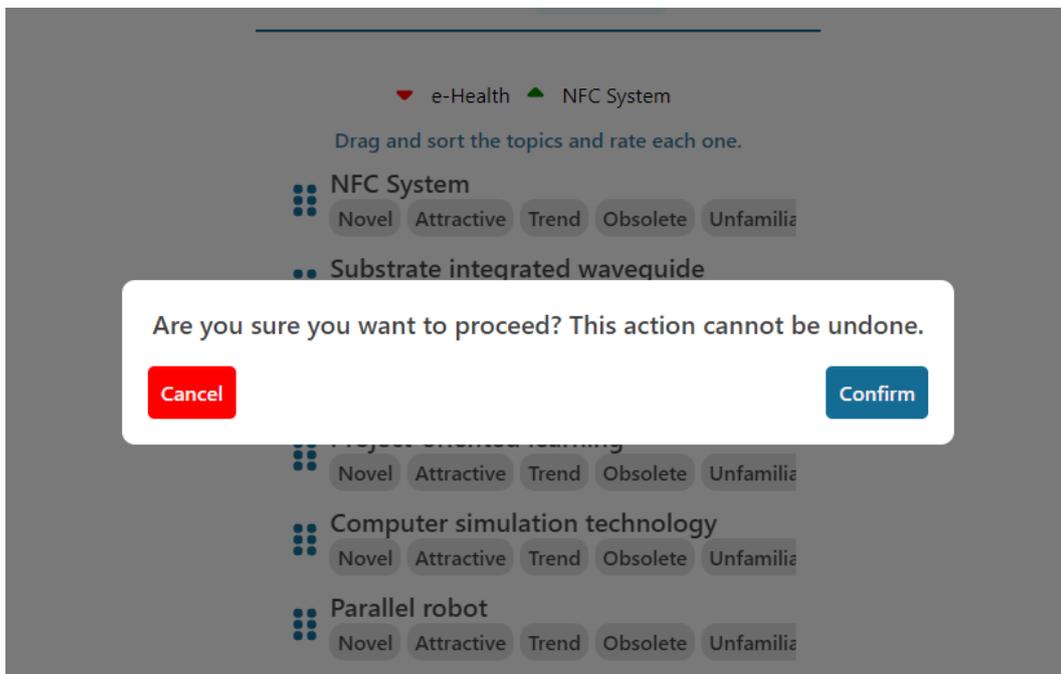


Figura 3.6: Mensaje de advertencia. Elaborado por el autor.

3.1.4 Presentación de tópicos aceptados y descartados, junto con sus justificaciones

Después de que el usuario haya completado su votación, se presentan los tres tópicos con el menor *score* (escala de tipo ranking), los cuales son los aceptados para el grupo, como se muestra en la Figura 3.7. Del mismo modo, se muestran los tres tópicos con el mayor *score*, que son los rechazados para el grupo. Además, en esta pantalla se visualizan los calificativos asignados a cada tópico, en caso de que se haya realizado una calificación.

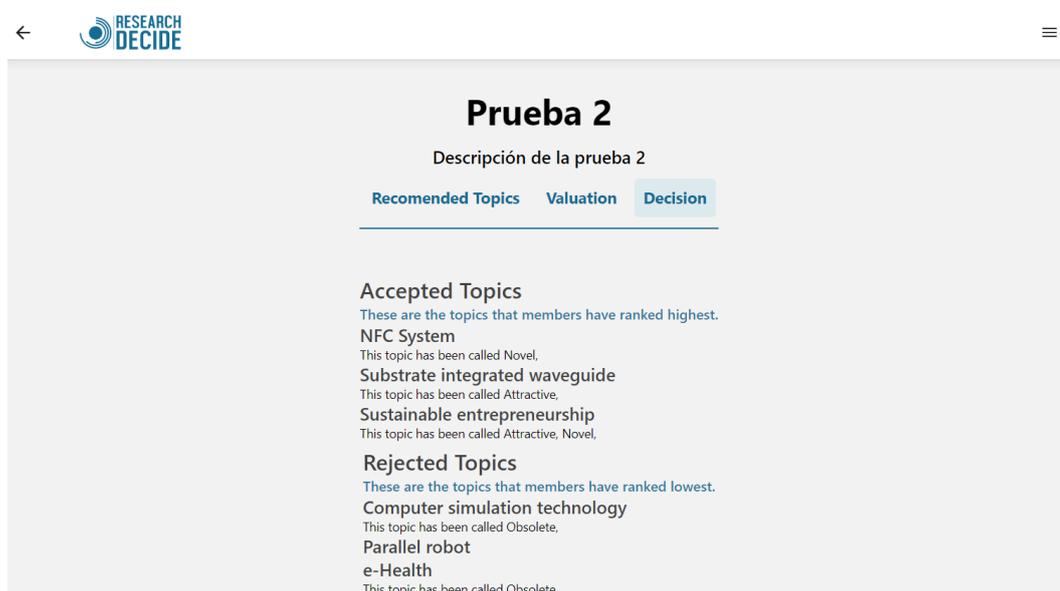


Figura 3.7: Tópicos aceptados y descartados, junto con sus justificaciones. Elaborado por el autor.

3.2 NIVEL DE USABILIDAD

En la Tabla 3.1 se presentan los resultados obtenidos a partir del formulario SUS (System Usability Scale) por pregunta, derivados de la ejecución de las pruebas de usabilidad realizadas a usuarios investigadores o estudiantes con interés en la investigación.

El promedio de todas las preguntas del cuestionario de usabilidad *SUS* fue de 77.5, lo que se considera un puntaje aceptable para evaluar la facilidad de uso percibida por los usuarios del módulo. Esta evaluación se basa en la representación gráfica de calor del *SUS*, donde los puntajes por debajo de 68 indican un nivel marginal de facilidad de uso, mientras que los

puntajes por encima de 68 son considerados aceptables [33], como se observa en la Figura 3.8.

N° Pregunta	Suma respuesta pares	Suma respuesta impares	Total	Puntos
1	19	13	32	80
2	20	18	38	95
3	11	12	23	57.5
4	21	11	32	80
5	16	15	31	77.5
6	10	9	19	47.5
7	18	14	32	80
8	20	16	36	90
9	21	15	36	90
			Promedio	77.5

Tabla 3.1: Evaluación SUS.

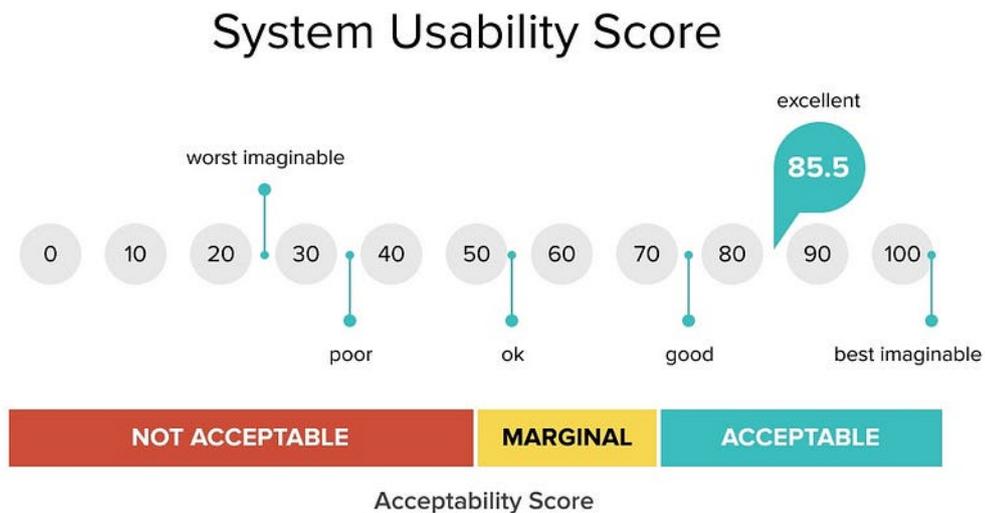


Figura 3.8: Puntaje SUS [33].

El puntaje promedio refleja que los usuarios valoran positivamente la capacidad del módulo para cumplir con su propósito, como generación y presentación de tópicos recomendados, la posibilidad de realizar una búsqueda simple de cada tópico y la búsqueda de forma combinada de tópicos, el reordenamiento y calificación de tópicos, además, la visualización de los tópicos aceptados y rechazados.

Además, varios participantes manifestaron el deseo de incluir imágenes ilustrativas que brinden una comprensión inicial del propósito del módulo y del sistema. Dado que la mayoría de los usuarios expresaron el mismo deseo, la implementación de esta característica podría contribuir a un aumento en el promedio del puntaje SUS y a una mayor percepción de utilidad por parte de los usuarios.

3.3 MÉTRICAS DE EVALUACIÓN

En la Figura 3.9 se muestra el promedio del tiempo que los usuarios tomaron para comprender e investigar los tópicos presentados. La mayoría de los usuarios requirió entre 3 y 5 minutos para realizar búsquedas en Google Scholar y obtener una comprensión más profunda de cada tópico. Estos resultados confirman la eficacia de las funcionalidades de los módulos *B* y *C* en permitir a los investigadores comprender rápidamente temas que desconocían.

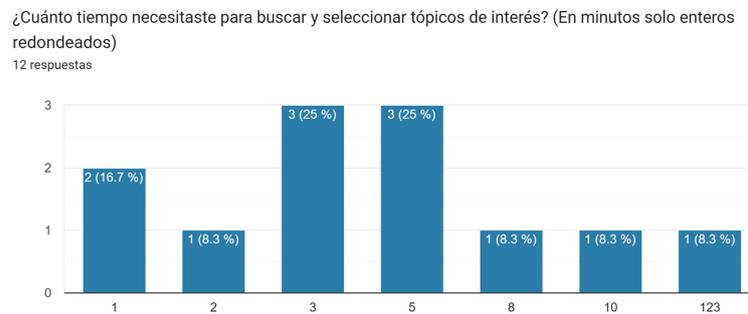


Figura 3.9: Tiempo promedio para comprender e investigar los tópicos.

Finalmente, la Figura 3.10 presenta el promedio del tiempo que los usuarios necesitaron para tomar una decisión final sobre el reordenamiento de los tópicos. La mayoría de los usuarios empleó alrededor de 5 minutos para realizar la votación después de reorganizar los tópicos presentados. Estos resultados refuerzan la efectividad de las funcionalidades de los módulos *B* y *C* al permitir que los investigadores obtengan una clasificación mejorada de los tópicos de interés. ^[1]

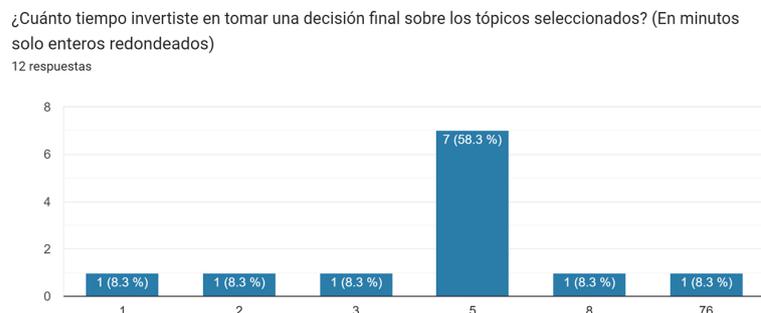


Figura 3.10: Tiempo promedio para la toma de la decisión final sobre los tópicos.

^[1] **Nota:** En la sección 6, se proporcionan enlaces a repositorios que contienen todos los activos generados en el proyecto, incluyendo prototipos, documentación, código fuente, manuales de instalación y resultados de evaluaciones de usabilidad y calidad.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- ❑ En este proyecto, se abordó la problemática de la toma de decisiones y consenso en el ámbito académico a través del desarrollo de un aplicativo móvil y web. Se identificó que la colaboración y el proceso de consenso entre investigadores presentaban desafíos relacionados con la autoridad, la comunicación y las diferencias de habilidades.
- ❑ La implementación de módulos que permiten la generación de recomendaciones grupales, la presentación y reordenamiento de los mismos, han demostrado ser una solución prometedora para mejorar la toma de decisiones grupales y lograr un consenso equilibrado.
- ❑ La metodología BDD utilizada en el desarrollo permitió enfocarse en el comportamiento esperado del software y garantizar una entrega de alta calidad. El enfoque ágil y la iteración constante en la evolución de los requerimientos demostraron ser adecuados para las características cambiantes del sistema.
- ❑ Además, los resultados de las pruebas de usabilidad y las evaluaciones de los usuarios reflejan la efectividad y utilidad de los módulos desarrollados. La puntuación obtenida en la escala de usabilidad SUS indica que los usuarios perciben la facilidad de uso y la eficacia de las funcionalidades implementadas.
- ❑ En términos de aplicabilidad, este proyecto tiene el potencial de impactar positivamente en la colaboración entre grupos de investigadores, facilitando la toma de decisiones, el consenso y la optimización de la experiencia del usuario en el ámbito académico.
- ❑ Las lecciones aprendidas en este trabajo pueden servir de base para futuras investigaciones y desarrollos en la mejora de la colaboración y la toma de decisiones en otros contextos similares.

4.2 RECOMENDACIONES

- ❑ A partir de las pruebas de usabilidad, se identificó la necesidad de incorporar imágenes ilustrativas y elementos visuales que proporcionen una comprensión rápida y clara de las funcionalidades del aplicativo. Se recomienda trabajar en la mejora de la interfaz de usuario para garantizar una experiencia agradable y efectiva para los usuarios.
- ❑ Aunque no se contempló en este proyecto, sería beneficioso explorar la incorporación de algoritmos de recomendación para generar recomendaciones más precisas y personalizadas para los grupos de investigadores. Esto podría mejorar aún más la calidad de las recomendaciones y la experiencia del usuario.
- ❑ Para obtener una perspectiva más amplia sobre la usabilidad y la experiencia del usuario, se recomienda realizar pruebas con un grupo más diverso de usuarios, incluyendo investigadores de diferentes áreas y niveles de experiencia.
- ❑ Dado que el campo de la investigación y la tecnología están en constante evolución, se sugiere mantener el aplicativo actualizado y seguir recopilando retroalimentación de los usuarios para realizar mejoras continuas y ajustes necesarios en el futuro.
- ❑ Finalmente, para trabajos futuros se sugiere la implementación de un chatbot que actúe como guía en el proceso de toma de decisiones y consenso en grupos de investigación. Al agilizar la toma de decisiones y mejorar la comunicación, el chatbot podría contribuir a consensos más rápidos y a una dinámica de grupo mejorada, especialmente en contextos dispersos o con horarios ocupados.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] M. M. Pérez, «El trabajo colaborativo en el aula universitaria,» *Laurus*, vol. 13, n.º 23, págs. 263-278, 2007.
- [2] I. P. Quintana-Moreno, «Toma de decisiones grupales:¿Dos cabezas piensan mejor que una?,» 2021.
- [3] A. Delic, H. Emamgholizadeh y F. Ricci, «CHARM: A Group Recommender ChatBot,» en *Adjunct Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, 2023, págs. 275-282.
- [4] P. Deemer, G. Benefield, C. Larman y B. Vodde, «Información básica de SCRUM,» *California: Scrum Training Institute*, 2009.
- [5] A. E. Soraluz Soraluz, M. Á. Valles Coral y D. Lévano Rodríguez, «Desarrollo guiado por comportamiento: buenas prácticas para la calidad de software,» *Ingeniería y Desarrollo*, vol. 39, n.º 1, págs. 190-204, 2021.
- [6] S. Dara, C. R. Chowdary y C. Kumar, «A survey on group recommender systems,» *Journal of Intelligent Information Systems*, vol. 54, n.º 2, págs. 271-295, 2020.
- [7] A. Felfernig, L. Boratto, M. Stettinger, M. Tkalčič et al., *Group recommender systems: An introduction*. Springer, 2018.
- [8] A. Torres Hidalgo y R. Ramos Rayski, «Dinamica de grupo,» 2017.
- [9] J. L. Escorihuela, «Taller de habilidades del facilitador,» *Acciones e investigaciones sociales*, n.º 22, págs. 157-226, 2006.
- [10] F. Ricci, L. Rokach y B. Shapira, «Recommender systems: Techniques, applications, and challenges,» *Recommender Systems Handbook*, págs. 1-35, 2021.
- [11] T. Nguyen, «Developing geolocation chat base application with ionic framework,» 2016.

- [12] Microsoft, *React Native for Windows*, <https://learn.microsoft.com/es-es/windows/dev-environment/javascript/react-native-for-windows>, Accedido el 10-07-2023.
- [13] Mozilla Developer Network, *What is JavaScript?* https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript, Accedido el 10-07-2023.
- [14] Node.js, *Acerca de Node.js*, <https://nodejs.org/es/about>, Accedido el 10-07-2023.
- [15] Microsoft, *What is Git?* <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>, Accedido el 10-07-2023.
- [16] Google, *Firebase*, <https://firebase.google.com/?hl=es>, Accedido el 10-07-2023.
- [17] UCI Blogs, *How to Use Font Awesome on Your Site*, <https://sites.uci.edu/blog/tips-tricks/how-to-use-font-awesome-on-your-site/>, Accedido el 10-08-2023.
- [18] J. Biezemans, *introduction - cucumber documentation*₂₀₁₉, 2019. dirección: <https://cucumber.io/docs/guides/overview/>.
- [19] C. Nakazawa, *Jest*, 2017. dirección: <https://archive.jestjs.io/en/>.
- [20] UNIR, *¿Qué es TypeScript?* <https://www.unir.net/ingenieria/revista/que-es-typescript/>, Accedido el 10-07-2023.
- [21] E. H. Uribe y L. E. V. Ayala, «Del manifiesto ágil sus valores y principios,» *Scientia et technica*, vol. 13, n.º 34, págs. 381-386, 2007.
- [22] J. F. Smart y J. Molak, *BDD in Action: Behavior-driven development for the whole software lifecycle*. Simon y Schuster, 2023.
- [23] C. B. Jurado, *Diseño Ágil con TDD*. Lulu. com, 2010.
- [24] S. Group, *La conquista de la calidad: pruebas de software con metodologías ágiles - Stefanini Group*, mar. de 2021. dirección: <https://stefanini.com/es/tendencias/articulos/la-conquista-de-la-calidad-pruebas-de-software-con-metodologias-agiles>.
- [25] Atlassian, *Backlogs*, <https://www.atlassian.com/es/agile/scrum/backlogs>, Accedido el 10-08-2023.
- [26] M. Bradley, *Agile Estimation Guidance*, ago. de 2016. dirección: <https://www.leadingagile.com/2016/08/agile-estimation-guidance/>.
- [27] *técnicas de priorización ágiles*₂₀₁₉, 2019. dirección: <https://es.slideshare.net/GiovannyAndresCifuen/tcnicas-de-priorizacin-agiles>.
- [28] D. Crockford, *Code Conventions for the JavaScript Programming Language*, 2023. dirección: <https://www.crockford.com/code.html>.

- [29] A. Alonso, *Convención de nombres: desde el CamelCase hasta el kebab-case*, jun. de 2017. dirección: <https://adrianalonsodev.medium.com/convenci%C3%B3n-de-nombres-desde-el-camelcase-hasta-el-kebab-case-787e56d6d023>.
- [30] D. Verma, *Convención de nomenclatura de sucursales de Git: 7 mejores prácticas a seguir*, mar. de 2021. dirección: <https://hackernoon.com/es/git-branch-naming-convention-7-best-practices-to-follow-1c2l33g2>.
- [31] V. Shevchuk, *Top 6 Best Folder Structures for React: Ultimate Comparison*, mayo de 2023. dirección: <https://itnext.io/top-6-best-folder-structures-for-react-ultimate-comparison-efc29ae5045>.
- [32] J. Nielsen, «Jakob Nielsen,» *Sort*, vol. 50, n.º 100, pág. 500, 1998.
- [33] A. Smyk, *The System Usability Scale How it's Used in UX - Thinking Design - Medium*, mar. de 2020. dirección: <https://medium.com/thinking-design/the-system-usability-scale-how-its-used-in-ux-b823045270b7>.

6 ANEXOS

Anexo 1: Prototipos del proyecto en *Figma*

Enlace del prototipo:

[https://www.figma.com/file/5T0avhsKPha6rml0fi2t60/Sistema-Grupos-Investigaci%C3%B3n-\(Copy\)?type=design&node-id=0%3A1&mode=design&t=BUmZAYSuYRjFxEpW-1](https://www.figma.com/file/5T0avhsKPha6rml0fi2t60/Sistema-Grupos-Investigaci%C3%B3n-(Copy)?type=design&node-id=0%3A1&mode=design&t=BUmZAYSuYRjFxEpW-1)

Anexo 2: Repositorio del proyecto desarrollado en *React Native*

Enlace del repositorio:

<https://github.com/SITDECG/ReSearchDecide-v1.0.0.git>

Anexo 3: Resultados evaluación SUS

Enlace del documento:

https://docs.google.com/spreadsheets/d/1ifBNywbdkSLdkEE4MLO7-CjyiQC_s78WM22a0SqsNg/edit?usp=sharing

Anexo 4: Resultados métricas de evaluación del sistema

Enlace del documento:

https://docs.google.com/spreadsheets/d/1NW_Xb38u2z-eVZ-HLtaEc2CKaK5BPDT_F3D86zAYA3M/edit?usp=sharing