



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

TUTOR COGNITIVO DE LÓGICA Y TEORÍA DE CONJUNTOS

Proyecto para optar al título de
**Ingeniero Civil en
Informática**

PROFESOR PATROCINANTE:
JULIO DANIEL GUERRA HOLLSTEIN
DOCTOR OF PHILOSOPHY IN INFORMATION
SCIENCES
INGENIERO CIVIL EN INFORMÁTICA
LICENCIADO EN CIENCIAS DE LA INGENIERÍA

PROFESOR CO-PATROCINANTE
JUAN FRANCISCO HERRERA TOBAR
MAGISTER EN MATEMÁTICAS
LICENCIADO EN MATEMÁTICAS

FELIPE IGNACIO VIDAL DÍAZ

VALDIVIA – CHILE
AÑO 2023

ÍNDICE

ÍNDICE TABLAS	I
ÍNDICE FIGURAS	II
RESUMEN.....	III
ABSTRACT	IV
1. INTRODUCCIÓN	1
1.1 Contexto y motivación.....	1
1.3 Propuesta.....	1
1.4 Objetivos.....	2
2. MARCO TEÓRICO.....	4
2.1 Modelado Cognitivo	4
2.2 Sistemas de tutoría inteligente	4
2.2.1 Módulo del tutor	5
2.2.2 Módulo del Dominio	5
2.2.3 Módulo del Estudiante.....	5
2.2.4 Módulo Interfaz de Usuario	6
2.3 Tutores Cognitivos.....	6
2.4 Lógica y teoría de conjuntos.....	7
3. TUTORES INTELIGENTES PARA LÓGICA Y TEORÍA DE CONJUNTOS.....	8
3.1 Pregunta de revisión.....	8
3.2 Palabras claves seleccionadas	8
3.3 Fuente de búsqueda.....	9
3.4 Criterios de Inclusión / Exclusión.....	9
3.5 Extracción de información.....	9
3.6 Estudios incluidos y excluidos.....	10
3.7 Resultados	11
3.8 Conclusiones	13
4. METODOLOGÍA DE DESARROLLO	14
4.1 Modelado Cognitivo de los ejercicios	14
4.2 Metodología de desarrollo	14
4.3 TutorCamp	14
4.4 Plataforma LearnerModelGQL.....	15
4.4.1 Lenguaje de programación	15
4.4.2 Auth0	16
4.4.3 React	16
4.4.4 GraphQL.....	16
4.4.5 Framework.....	16
4.4.6 Componentes	16
4.4.7 Control de versiones	16
4.4.8 Despliegue	16
4.5 Validación.....	17
4.6 Supervisión de la creación de nuevo contenido.....	17
5. MODELAMIENTO DEL CONTENIDO	18
5.1 Modelado Cognitivo	18
5.2 Tópicos de contenido seleccionados.....	18

5.2.1	Proposiciones.....	19
5.2.3	Intervalos	19
5.3	Definición de ejercicios tipo	21
5.4	Mock-ups Ejercicios	22
5.4.1	Representación de intervalos	22
5.4.2	Resolución de Sucesiones	24
5.4.3	Intersección y Unión de conjuntos	25
5.4.4	Comprobación de Equivalencias Lógicas utilizando tablas de verdad	26
5.4.5	Reconocimiento de proposiciones	27
5.4.6	Utilización de conectivos lógicos para conjunciones.....	28
5.5	Requisitos.....	29
6.	IMPLEMENTACIÓN.....	31
6.1	Definición JSON.....	31
6.2	Definición de Tipos	33
6.3	Primera Iteración.....	34
6.3.1	<i>Exercise</i>	34
6.3.2	<i>StepComponent</i>	34
6.3.3	<i>DisplayComponent</i>	34
6.3.4	<i>DualInputs</i>	35
6.3.5	<i>TrueFalse</i>	36
6.3.6	<i>Blank</i>	36
6.3.7	<i>InputButtons</i>	37
6.4	Segunda Iteración	38
6.4.1	<i>showSteps</i>	39
6.4.2	<i>multiplePlaceholders</i>	40
6.4.3	<i>singlePlaceholder</i>	41
6.4.4	<i>tableStep</i>	42
7.	<i>Responsividad en la Aplicación</i>	43
7.1	<i>multiplePlaceholders</i>	43
7.2	<i>singlePlaceholders</i>	44
7.3	<i>TrueFalse</i>	45
7.4	<i>Alternatives</i>	46
7.5	<i>TableStep</i>	47
7.6	<i>Blank</i>	48
8.	EVALUACIÓN	50
9.	CONCLUSIONES	53
10.	REFERENCIAS.....	55

ÍNDICE TABLAS

Tabla	Página
Tabla 1. Información por artículo	10
Tabla 2. Selección de Tópicos	22
Tabla 3. Primera iteración Representación de intervalos.....	22
Tabla 4. Segunda iteración Representación de intervalos.....	23
Tabla 5. Representación de sucesión	25
Tabla 6. Representación de Intersección.....	26
Tabla 7. Representación Tablas de Verdad.....	27
Tabla 8. Reconocimiento de Proposiciones	28
Tabla 9. Utilización de conectivos lógicos para conjunciones	29
Tabla 10. Definición de los atributos de la estructura para los ejercicios.....	31
Tabla 11. Estructura del contenido.....	32
Tabla 12. Validación	50
Tabla 13. Resultados	51

ÍNDICE FIGURAS

Figura	Página
Figura 1. Arquitectura de Sistema Inteligente de Enseñanza.....	5
Figura 2. Estructura general de un ejercicio.....	31
Figura 3. Estructura de los pasos.....	32
Figura 4. Definición de Tipos	33
Figura 5. Código <i>StepComponent</i>	34
Figura 6. Código de <i>DisplayComponent</i>	35
Figura 7. Despliegue <i>DualInputs</i>	36
Figura 8. Despliegue <i>TrueFalse</i>	36
Figura 9. Despliegue Blank.....	37
Figura 10. Despliegue <i>InputButtons</i>	38
Figura 11. Diagrama de componentes.....	38
Figura 12. Extracto de <i>showSteps</i>	39
Figura 13. Despliegue para sucesión.....	41
Figura 14. Despliegue para reconocimiento de extremos.	41
Figura 15. Despliegue en intervalo por comprensión.	41
Figura 16. Despliegue <i>singlePlaceholder</i>	42
Figura 17. Despliegue de tablas.	42
Figura 18. Representación <i>multiplePlaceholders</i> en diferentes resoluciones.....	44
Figura 19. Representación <i>singlePlaceholders</i> en diferentes resoluciones	45
Figura 20. Representación <i>trueFalse</i> en diferentes resoluciones	46
Figura 21. Representación <i>Alternatives</i> en diferentes resoluciones.....	47
Figura 22. Representación <i>tableStep</i> en diferentes resoluciones	48
Figura 23. Representación <i>Blank</i> en diferentes resoluciones.....	49
Figura 24. Rangos de la puntuación SUS.	51

RESUMEN

Muchos estudiantes ingresan a la Facultad de Ciencias de la Ingeniería cada año, pero gran parte de los estudiantes tienen problemas en los primeros cursos. Su formación profesional comienza con las asignaturas de Álgebra y Geometría para Ingeniería, las cuales poseen unas tasas de reprobación elevadas, debido, entre otros factores, a la falta de conocimientos previos y la brecha existente entre la educación secundaria y la universitaria.

Por este motivo es que ha surgido la necesidad de realizar nivelaciones y cursos de preparación. Para complementar los cursos de nivelación se ha puesto en marcha un proyecto financiado por Fondecyt que tiene el objetivo de desarrollar tutores inteligentes para nivelación de contenidos de matemática necesarios para los cursos iniciales. Este trabajo de título tiene como objetivo ampliar la cobertura de los tópicos a la primera unidad del curso de Álgebra para la Ingeniería correspondiente a Lógica y Teoría de Conjuntos.

Se realizaron sesiones de Modelado Cognitivo en conjunto a un profesor co-patrocinante de Ciencias básicas para realizar el diseño y modelado de los ejercicios, obteniendo un set de 6 ejercicios tipo que cubren los tópicos de representación de conjuntos, sucesiones matemáticas, unión e intersección de intervalos, proposiciones y conjunciones y tablas de verdad. La metodología de desarrollo utilizada fue iterativa incremental, donde se llevaron a cabo reuniones semanales entre el estudiante tesista y el profesor patrocinante para definir nuevos requerimientos y/o validar las nuevas funcionalidades implementadas.

La implementación se realizó en base a plantillas de desarrollo front-end de la plataforma LearnerModelGQL para el desarrollo de tutores inteligentes sobre NextJS, React y GraphQL. Los requerimientos de los nuevos ejercicios demandaron el desarrollo de innovaciones sobre los componentes existentes incluyendo nuevos tipos de ingreso de respuesta, como alternativas, verdadero/falso, tablas de verdad e ingresos de expresiones parciales mediante placeholders.

El tutor fue probado en términos de usabilidad mediante una sesión de pruebas con once estudiantes de primer año de la universidad austral de Chile, quienes luego de hacer uso del tutor debieron rellenar un cuestionario SUS, obteniendo buenos resultados, pero fueron requeridos cambios para mejorar la accesibilidad de algunos símbolos matemáticos en el ingreso de respuestas.

ABSTRACT

Many students enroll in the Facultad de Ciencias de la Ingenieria each year, but many students have difficulties in their first courses. Their professional training begins with the subjects of Algebra and Geometry for Engineering, which have high failure rates due, among other factors, to a lack of prior knowledge and the gap between high school and university education.

For this reason, the need has arisen to offer preparatory and leveling courses. To complement these leveling courses, a Fondecyt-funded project has been launched with the goal of developing intelligent tutors to reinforce the mathematical concepts required for the initial courses. This thesis project aims to expand the coverage of topics to include the first unit of the Algebra for Engineering course, which focuses on Logic and Set Theory. Cognitive Modeling sessions were conducted in collaboration with a co-sponsoring professor from the mathematics department to design and model the exercises.

This resulted in a set of six sample exercises covering topics such as set representation, mathematical sequences, union and intersection of intervals, propositions and conjunctions, and truth tables. The development methodology used was iterative and incremental, involving weekly meetings between the thesis student and the supervising professor to define new requirements and/or validate implemented functionalities.

The implementation was based on front-end development templates from the LearnerModelGQL platform for building intelligent tutors using NextJS, React, and GraphQL. The new exercise requirements demanded innovations to existing components, including new types of answer inputs, such as multiple-choice, true/false, truth tables, and partial expressions through placeholders.

The tutor's usability was tested during a session with eleven first-year students from the Universidad Austral de Chile. After using the tutor, they completed a System Usability Scale (SUS) questionnaire, yielding positive results. However, changes were needed to improve the accessibility of certain mathematical symbols in the answer input process.

1. INTRODUCCIÓN

1.1 Contexto y motivación

En la Facultad de Ciencias de la Ingeniería (FCI), el año 2023 se matricularon más de 400 estudiantes (Uribe, 2023). Estos estudiantes comienzan su formación con las asignaturas de *Álgebra para Ingeniería* y *Geometría para la Ingeniería*, que tienen una alta tasa de reprobación, debido, entre otros factores, a la falta de conocimientos previos y la brecha existente entre la educación secundaria y la universitaria.

Con el fin de abordar esta situación, la Facultad de Ciencias de la Ingeniería de la Universidad Austral de Chile (FCI) ha buscado abordar la problemática mediante el desarrollo de tutorías y un programa de nivelación para estudiantes de primer año. Estos tienen el objetivo de reforzar las habilidades matemáticas fundamentales para enfrentar los retos académicos que se presentan durante el primer semestre. Junto a este programa, se ha puesto en marcha un proyecto financiado por Fondecyt que tiene el objetivo de desarrollar tutores inteligentes para nivelación de contenidos de matemática necesarios para los cursos iniciales. Actualmente están en desarrollo tutores enfocados en la nivelación de los estudiantes en tópicos como factorización, fracción algebraica, potencias y raíces, entre otros. En este trabajo se busca ampliar la cobertura de los tutores para incluir contenidos de lógica y conjuntos, los que corresponden a la primera unidad del curso de *Álgebra para la Ingeniería*.

Los tutores cognitivos pueden evaluar las habilidades y conocimientos de cada estudiante al monitorear y analizar sus interacciones, respuestas y avances en el sistema. Esto les permite identificar problemas que aborden de manera efectiva las áreas menos desarrolladas del estudiante. En consecuencia, se logra una enseñanza adaptativa y personalizada que se ajusta a las necesidades individuales de cada persona. Además, el tutor busca realizar una enseñanza lo más parecida a un profesor siguiendo sus metodologías, emulando las mejores prácticas y estrategias pedagógicas para garantizar una experiencia de aprendizaje efectiva.

1.3 Propuesta

El presente trabajo propone el desarrollo e implementación de un tutor inteligente en el ámbito de álgebra enfocado en lógica y conjuntos, con el objetivo de continuar apoyando a los estudiantes posterior a los tópicos de nivelación.

La lógica en las matemáticas se refiere al estudio y la aplicación de las reglas y principios que gobiernan el razonamiento válido y la inferencia. Es la base fundamental de la matemática, ya que proporciona un marco para la formulación precisa y la resolución de problemas matemáticos. La lógica se utiliza para demostrar teoremas, validar argumentos

y establecer conexiones coherentes entre diferentes conceptos matemáticos. La lógica matemática se divide en varias ramas, como la lógica proposicional, la lógica de primer orden y la teoría de modelos, que se utilizan para analizar y estructurar el razonamiento en matemáticas.

La teoría de conjuntos es una rama fundamental de las matemáticas que se ocupa del estudio de los conjuntos, que son colecciones bien definidas de objetos. En la teoría de conjuntos, se definen conceptos fundamentales como la unión, la intersección, la diferencia y el conjunto vacío. Además, se establecen axiomas que rigen las operaciones y relaciones entre conjuntos, como el axioma de extensión y el axioma de regularidad. La teoría de conjuntos se utiliza para construir números naturales, enteros, racionales, números reales y otros objetos matemáticos. La teoría de conjuntos también desempeña un papel esencial en la formulación de teoremas y en la resolución de problemas en diversas ramas de las matemáticas, como el análisis matemático, la topología, la teoría de números y la lógica misma.

La solución propuesta en este trabajo se basa en desarrollar interfaces interactivas para ejercicios de álgebra utilizando el framework de React NextJs. Este framework permite crear aplicaciones web dinámicas y de alto rendimiento con una experiencia de usuario atractiva y fácil de usar. En estas interfaces, el estudiante podrá solicitar ayuda mediante pistas adaptativas las cuales se organizan en niveles de profundidad y se ajustan a las respuestas incorrectas que el usuario selecciona, proporcionando orientación personalizada y específica para mejorar la comprensión del tema. Para este trabajo es necesario el trabajo en conjunto con profesores del Centro de Docencia de Ciencias Básicas para ingeniería (CDCB) para la selección, estructuración y validación de los ejercicios a modelar, garantizando su calidad y coherencia con los objetivos y competencias de la asignatura.

Toda la implementación del tutor de lógica propuesto se realiza en el marco de trabajo de la plataforma LearnerModel GQL (Sáez, 2019), que establece una plataforma de microservicios y una plantilla de desarrollo para front-end. Todas las interacciones que realiza el usuario con la interfaz del tutor, como pueden ser el tiempo de estadía en el ejercicio, solicitud pistas, responder correcta o incorrectamente un paso, son recopiladas y almacenadas usando la API de microservicios para su posterior análisis y seguimiento del progreso del estudiante.

1.4 Objetivos

1.4.1 Objetivo General

Potenciar las habilidades de lógica y conjuntos de los estudiantes de la Facultad de Ciencias de la Ingeniería por medio del diseño de un tutor cognitivo enfocado en la resolución de ejercicios de estos tópicos e integrando las buenas prácticas de la plataforma en desarrollo del proyecto Fondecyt.

1.4.2 Objetivos Específicos

- O.E.1. Sintetizar información sobre los tutores ya existentes, ejercicios de aplicación de lógica e investigación de plataformas de aprendizaje de lógica y conjuntos.
- O.E.2. Realizar un Modelado Cognitivo del contenido y ejercicios a implementar.
- O.E.3. Construir un prototipo de tutor con una interfaz con enfoque responsivo, homogénea a las interfaces de otros tutores.
- O.E.4. Desplegar un tutor cognitivo de lógica y conjuntos con un conjunto base de ejercicios modelados en la plataforma existente.
- O.E.5. Evaluar el tutor con un grupo de estudiantes para tener retroalimentación de su usabilidad.

Los objetivos específicos pueden ser encontrados en los siguientes capítulos:

- O.E.1. Capítulo 3. TUTORES INTELIGENTES PARA LÓGICA Y TEORÍA DE CONJUNTOS
- O.E.2. Capítulo 5. MODELAMIENTO DEL CONTENIDO
- O.E.3. Capítulo 6. IMPLEMENTACIÓN y Capítulo 7. RESPONSABILIDAD EN LA APLICACIÓN
- O.E.4. Capítulo 6.4 Segunda Iteración
- O.E.5. Capítulo 8. EVALUACIÓN

2. MARCO TEÓRICO

En este capítulo se presentarán distintos conceptos que se encuentran presentes en los tutores cognitivos.

2.1 Modelado Cognitivo

El modelado cognitivo es un enfoque interdisciplinario que se centra en la creación de modelos teóricos o computacionales que representan y simulan los procesos mentales y cognitivos humanos. Estos modelos buscan capturar cómo las personas adquieren, procesan y utilizan la información, así como cómo toman decisiones y resuelven problemas en diversas situaciones. El objetivo fundamental del modelado cognitivo es comprender y explicar los aspectos internos y externos de la cognición humana para poder predecir el comportamiento humano y diseñar sistemas y tecnologías más efectivas y compatibles con el pensamiento humano.

2.2 Sistemas de tutoría inteligente

Los Sistemas de Tutoría Inteligente (ITSs, por sus siglas en inglés) son programas de computadora que utilizan técnicas de inteligencia artificial para mejorar y personalizar la automatización en la enseñanza. (Alkhatlan & Kalita, 2019). Su objetivo principal es apoyar y mejorar el proceso de enseñanza y aprendizaje en el campo de conocimiento elegido.

VanLehn (1988) define un tutor inteligente como "un sistema de software que utiliza técnicas de inteligencia artificial (IA) para representar el conocimiento e interactúa con los estudiantes para enseñárselo". En cambio, Giraffa, Nunes y Viccari (1997) lo definen como "un sistema que incorpora técnicas de IA (Inteligencia Artificial) a fin de crear un ambiente que considere los diversos estilos cognitivos de los alumnos que utilizan el programa"

En las arquitecturas modernas, los sistemas inteligentes de enseñanza están compuestos por cuatro módulos básicos (Nwana, 1990) como se aprecia en la Figura 1. Estos módulos serán descritos posteriormente.

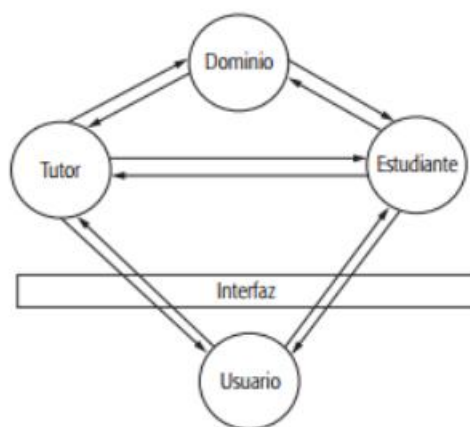


Figura 1. Arquitectura de Sistema Inteligente de Enseñanza.

2.2.1 Módulo del tutor

El módulo del tutor utiliza la información recolectada por los módulos del dominio y del estudiante para escoger los problemas que el estudiante deberá resolver. También es el encargado de definir la planificación y representación del contenido a enseñar. El papel del módulo del tutor no sólo es proporcionar orientación como un tutor, sino también hacer que la interacción del tutor cognitivo con el alumno sea fluida y natural (Bourdeau & Grandbastien, 2010). Por último, se encarga de la interacción entre el alumno y el tutor en caso de que el estudiante necesite ayuda en algún momento.

2.2.2 Módulo del Dominio

El módulo del dominio representa las reglas y estrategias para la resolución de problemas en un dominio (materia) en particular. Toda la información y conocimiento del dominio escogido deben ser presentados en forma adecuada hacia el estudiante para que este pueda adquirir las habilidades y conceptos requeridos. La meta del módulo del dominio es encontrar la materia que enseñará el tutor.

2.2.3 Módulo del Estudiante

Es una representación computacional o teórica del conocimiento, habilidades, aptitudes y características del estudiante que está siendo asistido por el sistema. Este modelo se utiliza para comprender y adaptar la interacción entre el sistema y el estudiante de manera personalizada y efectiva. El modelo del estudiante se crea y se actualiza continuamente a medida que el estudiante interactúa con el sistema. Utilizando este modelo, el STI puede personalizar la entrega de contenido, las actividades y los desafíos para adaptarse a las necesidades y características individuales del estudiante, lo que facilita un proceso de aprendizaje más efectivo y eficiente. Alkhatlan y Kalita (2019) establecen que "El sistema debe ser capaz de inferir aspectos no observables del comportamiento del alumno a partir del modelo. Debe reconstruir conceptos erróneos de los conocimientos del alumno mediante la interpretación de sus acciones".

2.2.4 Módulo Interfaz de Usuario

El módulo de interfaz de usuario gestiona la interacción entre el estudiante y el sistema de tutoría inteligente. Puede incluir elementos visuales, de audio y de texto que permiten al estudiante acceder al contenido y realizar tareas de aprendizaje. La interfaz de usuario también es responsable de recopilar información sobre las interacciones del estudiante con el sistema, que luego se utiliza para actualizar el modelo del estudiante y adaptar la instrucción y el apoyo proporcionado por el modelo del tutor.

2.3 Tutores Cognitivos

Los ITS han permitido el desarrollo de un nuevo tipo de entorno de aprendizaje inteligente llamado tutores cognitivos (Koedinger & Corbett, 2006). Los tutores cognitivos son herramientas tecnológicas diseñadas para ayudar a las personas a mejorar su rendimiento cognitivo en áreas como la memoria, la atención y el razonamiento. Estos programas suelen estar diseñados para adaptarse al nivel individual de habilidad y las necesidades del usuario, cumpliendo dos de las tareas principales características del tutor humano: monitorear el desempeño del estudiante y proporcionar instrucción específica del contexto justo cuando el estudiante individual lo necesita.

Dentro del contexto de los tutores inteligentes, otra tarea relevante consiste en la selección precisa de contenidos, como por ejemplo la resolución de problemas que se encuentren dentro del alcance de conocimiento específico del estudiante. Este monitoreo del desempeño y aprendizaje de los estudiantes hace uso del modelo cognitivo y dos algoritmos clave, el rastreo de modelo y el rastreo de conocimiento. En el rastreo de modelo, el tutor cognitivo ejecuta el modelo cognitivo hacia adelante paso a paso junto con el estudiante para seguir el camino individual del estudiante a través de espacios de problemas complejos, proporcionando retroalimentación de precisión en tiempo real y consejos específicos del contexto. En el rastreo de conocimiento, el tutor utiliza un método bayesiano simple para estimar el conocimiento del estudiante y utiliza este modelo del estudiante para seleccionar problemas apropiados. (Koedinger & Corbett, 2006).

Un ejemplo detrás de la investigación para la realización de tutores cognitivos es el artículo *Cognitive Tutors: Lessons Learned* en el cual se plantea la investigación para el desarrollo de los tutores cognitivos, esto mediante el análisis de tutores de diferentes áreas como geometría, álgebra y el lenguaje de programación LISP. Junto con esto se detalla el rastreo del conocimiento y el modelo de control de retroalimentación (Anderson, Corbett, Koedinger, Pelletier, 1995).

En el artículo *Cognitive Tutor: Applied research in mathematics education* se plantean los avances en la comprensión de la cognición matemática a través del modelado cognitivo y la aplicación de ese conocimiento para desarrollar planes de estudio. Se explica el avance en la cognición en general y de la cognición matemática en particular, junto con esto los planes de estudio resultantes han demostrado ser efectivos en entornos escolares (Ritter, Anderson, Koedinger, Corbett, 2007).

2.4 Lógica y teoría de conjuntos

Lógica y teoría de conjuntos son el punto de partida del curso de *Álgebra para la Ingeniería* y están presentes, de manera implícita, en todos los procesos que involucran extraer información, razonar y comunicar utilizando matemática. Por esto, esta unidad es un pilar fundamental y será de utilidad para todas las asignaturas del área matemática.

Para abordar adecuadamente las temáticas presentadas, es necesario el manejo de conocimientos previos, todos ellos adquiridos en la enseñanza media, tales como: conjuntos numéricos, operatoria básica de números enteros y racionales, además de raíces y manipulación de expresiones algebraicas. Dentro de los contenidos que comprende esta unidad se tienen los siguientes:

- Lógica Proposicional que comprende los conocimientos de proposiciones, tablas de verdad y equivalencias lógicas, conectivos lógicos y sus propiedades, tautologías, contradicciones y contingencias.
- Nociones Básicas de conjuntos que comprende representaciones conjuntistas, intervalos, inclusión e igualdad, operaciones conjuntistas y sucesiones numéricas.
- Proposiciones cuantificadas que comprende Cuantificadores, proposiciones con múltiples cuantificadores y negación de proposiciones cuantificadas.

Estos contenidos van se van dando de forma incremental teniendo que aplicar los conocimientos de las fases iniciales del curso en la resolución de ejercicios posteriores.

La construcción de pruebas lógicas es una habilidad importante tanto para la informática como para la filosofía. Sin embargo, en nuestra experiencia, este tema es particularmente difícil para los estudiantes, especialmente para determinar una estrategia para derivar una conclusión a partir de premisas dadas. Una vez que los estudiantes aplican reglas que les resultan fáciles, a menudo quedan atrapados, realizando pasos innecesarios o dando por vencidos cuando el siguiente paso no está claro. En sesiones de tutoría individual, sin embargo, sugerir uno o más estados de objetivos intermedios ayuda a muchos estudiantes a lograr pruebas completas. (Facultad de Ciencias de la Ingeniería, 2023).

Existen trabajos relacionados con tutores de lógica como por ejemplo EvoLogic, este plantea un sistema de tutoría inteligente para ayudar a los estudiantes en problemas de producción natural en lógica proposicional, el cual admite estrategias pedagógicas inspiradas en la teoría del aprendizaje basado en ejemplos, basado en el uso de un mecanismo de rastreo de modelos automatizado. (Galafassi, Galafassi, Vicari, Reategui, 2023)

Otro ejemplo es el *ProofLab* el cual utiliza bibliotecas de ejercicios interactivos, sistemas de aprendizaje adaptativo y contenido multimedia para crear una experiencia de aprendizaje dinámica y efectiva. Esta plataforma permite realizar pruebas lógicas de la manera más intuitiva posible, también diagnostica errores en el uso de reglas o en la manipulación de pruebas y proporciona informes de errores con enlaces a material instructivo relacionado con los errores. Los estudiantes pueden previsualizar movimientos antes de aplicarlos en su argumento y también pueden guardar trabajos parciales (Sieg, 2007).

3. TUTORES INTELIGENTES PARA LÓGICA Y TEORÍA DE CONJUNTOS

Este capítulo comprende el Objetivo específico 1 (O.E.1) el cual consiste en realizar una revisión sistemática para resumir los avances existentes sobre tutores cognitivos referentes al tópico de Lógica y Teoría de Conjuntos. Las revisiones sistemáticas son el mejor esfuerzo por recopilar y sintetizar información de evidencia científica sobre un tema. Además, permiten encontrar hallazgos relevantes que serán utilizados en este proyecto.

Para el desarrollo de esta Revisión Sistemática (RS), se utilizó la herramienta Parsifal. Esta es una herramienta en línea diseñada para ayudar a los investigadores a realizar revisiones sistemáticas de la literatura en el contexto de la Ingeniería de Software (Parsifal, n. d.).

Todas las consultas a las bibliotecas digitales se hicieron a través de la VPN de la UACH (Acceso remoto VPN, n. d.), este servicio permite establecer un canal seguro de comunicaciones entre un computador personal conectado a una red de datos externa y los servicio de la Red Corporativa de la Universidad.

3.1 Pregunta de revisión

La necesidad de la Revisión Sistemática consiste en confirmar el uso real de tutores cognitivos y cuáles son sus beneficios. Para llevar a cabo esta RS se necesita elaborar una pregunta de investigación, que es la parte más importante en cualquier RS.

La pregunta de revisión es:

¿Qué tutores cognitivos existen para el tópico de lógica y conjuntos?

Esta pregunta fue obtenida luego de dos reuniones con el profesor patrocinante Julio Guerra, debido a que inicialmente se planteó una pregunta demasiado amplia, la cual fue acotada para el trabajo de título.

3.2 Palabras claves seleccionadas

Se realizó una búsqueda preliminar en la cual se pudieron identificar palabras que arrojaban el mayor número de resultados y que respondían de alguna forma a la pregunta de investigación que se planteó, además de una posterior reunión con el profesor patrocinante para poder aumentar la cantidad de resultados. Se obtuvieron las siguientes palabras claves:

Cognitive Tutor, Intelligent Tutor, Tutoring System, Education, College, Instruction, Students, Learning, Logic and Sets, Logic.

Esta búsqueda preliminar se realizó solo en base a los resúmenes de los artículos.

Cadena de búsqueda

La cadena fue definida de la siguiente manera:

("Cognitive Tutor" OR "Intelligent Tutor" OR "Tutoring System") AND
("Education" OR "College" OR "Instruction" OR "Students" OR "learning") AND
("Logic and Sets" OR "Logic")

3.3 Fuente de búsqueda

La fuente seleccionada fue la librería digital de ISI Web Of Science y se tuvo acceso a esta biblioteca virtual mediante el servicio VPN de la universidad, VPN UACH.

3.4 Criterios de Inclusión / Exclusión

A partir de los resultados de la búsqueda, se seleccionaron aquellos artículos que presentaban información relevante sobre el desarrollo de un tutor de lógica o investigaciones sobre el funcionamiento de estos. Obteniendo en base a los siguientes criterios:

- Implementación del Tutor
- Material relacionado a lógica y conjuntos
- Información útil

Además, se tienen un par de artículos entregados por el profesor patrocinante.

Criterios de exclusión:

- Artículos que no pertenecían al área educativa
- Artículos obtenidos en la búsqueda de los cuales el tema no está relacionado a lógica proposicional o teoría de conjuntos.
- Artículos que describen una aplicación diferente a un tutor cognitivo

3.5 Extracción de información

Se definió una pauta de extracción de información (ver Tabla1) para tener una estructura de la información encontrada sobre los tutores de lógica y teoría conjuntos, sus modelos de dominio y estudiante, sus interacciones y el tipo de soporte que se le entrega al estudiante.

La pauta de extracción de información por artículo fue la siguiente:

- Ejercicios e interacciones
- Modelo de dominio/Estudiante
- Tipo de soporte al estudiante
- Existencia de librerías asociadas
- Hallazgos

Código Del Artículo	Ejercicios e interacciones	Modelo de dominio/Estudiante	Tipo de soporte al estudiante	Existencia de librerías asociadas	Hallazgos
[S06]	Los ejercicios se presentan en diferentes formatos, como preguntas de opción múltiple, completar espacios en blanco, y construir pruebas formales.	El modelo de dominio en el Logic-ITA se refiere a la representación del conocimiento y las reglas relacionadas con el dominio específico de la lógica proposicional. Este modelo incluye conceptos como las reglas de inferencia, las estructuras lógicas, los operadores lógicos y las propiedades de la lógica proposicional. El sistema utiliza esta información para adaptar los ejercicios y la retroalimentación a las necesidades individuales del estudiante. El modelo se actualiza continuamente a medida que el estudiante interactúa con el sistema.	El Logic-ITA proporciona varios tipos de soporte para estudiantes, incluyendo retroalimentación contextualizada, sugerencias, y ejercicios adaptativos. Los estudiantes también pueden practicar a su propio ritmo y recibir retroalimentación sobre sus errores. Además, el sistema recopila y almacena los resultados de las interacciones y los ejercicios en una base de datos, lo que permite a los profesores y tutores identificar problemas comunes y evaluar el progreso de los estudiantes.		

Tabla 1. Información por artículo

3.6 Estudios incluidos y excluidos

Al realizar la búsqueda, con la cadena definida, se obtuvieron un total de 27 resultados. Se leyó el resumen de todos ellos para así poder aplicar los criterios de selección definidos, lo que resultó en 7 artículos seleccionados que fueron leídos y resumidos. Respecto a los

artículos rechazados, 15 abordan un tema no relacionado donde se presentaban tutores cognitivos de lógica algorítmica en lugar de lógica proposicional, 5 presentan soluciones que no eran tutores cognitivos, y 1 artículo fuera del área educativa. Se puede observar por los años de publicación que se tienen mayor cantidad de estudios relacionados a partir del año 2006.

3.7 Resultados

Se completó la tabla anteriormente nombrada con los 7 artículos seleccionados (artículos en anexo A), extrayendo la información más relevante de dichos artículos. Los hallazgos claves, obtenidos como resultado de la Revisión Sistemática, se resumen y organizan en las siguientes categorías: Interacciones HCI, Modelo de dominio/estudiante, Tipo de soporte, Librerías asociadas, Hallazgos.

Interacciones HCI: Dentro de las principales interacciones que tienen los estudiantes con los tutores están la selección múltiple, completar espacios en blanco y selección de reglas para la construcción de pruebas formales (Lesta y Yacef, 2002). Además, es importante considerar el enfoque que el tutor adopta al realizar el ejercicio ya sea analizando las diferentes rutas que puede tomar el estudiante para la resolución del ejercicio o una ruta de solución predefinida. En ambos casos se tiene pistas asociadas para apoyar al estudiante en caso de que éste se equivoque (Galafassi, Penteado, Vicari, Reategui, Berni, 2022). Por último, existe la opción de que el estudiante escriba los ejercicios en lugar de recibirlos del tutor, lo que permite que el tutor brinde ayuda paso a paso en la resolución (Barnes y Stamper, 2010). Sin embargo, esta última opción presenta un mayor nivel de complejidad en su implementación, ya que requiere el planteamiento lógico y el análisis de la entrada del usuario, así como la generación automatizada de la solución paso a paso.

Modelo de dominio/estudiante: Uno de los factores más importantes en el desarrollo del tutor es la forma en que se plantea el modelo de dominio y el modelo del estudiante.

- El modelo de dominio se basa en el diseño de reglas, teoremas y el lenguaje formal para la lógica proposicional. Al tener estas bases bien definidas, podemos brindar un mejor soporte al estudiante tanto en términos de la información que el tutor puede proporcionar en las pistas, como en el desarrollo paso a paso de los ejercicios y en su abordaje.
- El modelo del estudiante, principalmente se basaba en la estimación del Nivel de conocimiento o maestría a partir de las acciones realizadas y posteriormente almacenadas por el estudiante a medida que se utiliza el tutor. Sin embargo, existen casos puntuales como el de Barnes y Stamper (2010) donde se implementaron los MDP (Markov Decision Process) para mejorar el modelo del estudiante y su interacción con el tutor, junto con utilizar la información recabada en años anteriores para brindar un mejor apoyo al estudiante.

Tipo de soporte al estudiante:

- Utilización de pistas, ya sea automatizadas en base a la información histórica como es el caso de Barnes, Stamper (2010), mediante el análisis de las posibles rutas que puede tomar el estudiante descrito por Galafassi, et al (2023) o específicas para cada ejercicio existente en el tutor, pudiendo éstas ser solicitadas por el estudiante o aparecer cuando el estudiante cometa un error en la resolución en este último caso existe la posibilidad de que la pista se defina en base a la respuesta específica del estudiante revisando si está en un pool de respuestas la cual posea pistas específicas.
- Mini-lecciones las cuales no son solicitadas por el estudiante. Estas funcionan como una intervención adaptativa cuando se detecta que el estudiante tiene pasos difíciles de entender o resolver (Maniktala, Chi, Barnes 2023). Las Mini-lecciones pueden venir como texto, videos cortos o ejemplos ilustrativos, son una buena alternativa a las pistas solicitadas por el estudiante o mostradas cuando el estudiante comete un error en un paso.
- Comprobación automática de teoremas como la planteada por Galafassi, et al (2022) la cual permite a los estudiantes verificar sus soluciones utilizando demostraciones formales. El sistema proporciona una demostración formal de la solución correcta, lo que permite al estudiante contrastar con su propia solución MDP(Markov Decision Process) y aprender de la demostración formal.

Librerías asociadas: El único artículo que nombró a una librería asociada fue el caso de Sieg (2007), la cual no es una librería privada. Debido a que esta librería no es de código abierto no fue posible utilizarla para el desarrollo del proyecto. Sin embargo, la aplicación de AProS fue de utilidad para el desarrollo del proyecto debido a sus interacciones con el usuario más concretamente en la resolución de ejercicios de proposiciones, ya que permite ejercitar obteniendo proposiciones a partir de textos, y luego evaluando su valor de verdad o realizar ejercicios con proposiciones ya definidas y aplicar tanto conectivos lógicos como propiedades para simplificarlas.

Hallazgos: Se encontraron hallazgos importantes durante la investigación, entre ellos se destaca la implementación de un predictor con el objetivo de reducir la cantidad de falsos positivos en la evaluación del conocimiento del estudiante (Maniktala, Chi, Barnes 2023). Esto ha permitido desarrollar un modelo más preciso de los conocimientos del estudiante, lo cual brinda un mejor apoyo y facilita una selección más efectiva al momento de ofrecer ejercicios. Además, otro aspecto interesante es la posibilidad de que el estudiante proponga ejercicios, los cuales son analizados y desglosados paso a paso por el tutor propuesto por Hsu (2006). De esta manera, se puede analizar la ruta que el estudiante sigue para resolver los ejercicios pudiendo brindar apoyo como pistas para esa ruta. Finalmente tenemos la comparación y resultados del experimento realizado con 2 tipos diferentes de pistas, las pistas de próximo paso las cuales indican como su nombre lo indica guían al estudiante para resolver el paso actual en el que se encuentra y los Waypoints las cuales son un tipo de pista no solicitada que proporciona orientación a los estudiantes sobre como

alcanzar un subobjetivo en un problema. Estas pistas son generadas analizando los caminos más frecuentes y óptimos tomados por los estudiantes al momento de realizar un ejercicio similar.

3.8 Conclusiones

Para el desarrollo de esta Revisión Sistemática (RS) se generó la pregunta en base a los requisitos de este proyecto de título. Utilizando la biblioteca virtual *Web of Science* como fuente de búsqueda encargada de proporcionar artículos relacionados a tutores cognitivos enfocados en Lógica y Teoría de Conjuntos y se establecieron los criterios de inclusión/exclusión en la búsqueda de artículos. La búsqueda obtuvo un total de 27 resultados de los cuales 7 fueron seleccionados, extrayendo la información relevante de cada uno por medio de una pauta de extracción.

Finalmente, los resultados de la RS se clasificaron en cuatro categorías: Interacciones HCI, Modelo de dominio/estudiante, Tipo de soporte, Librerías asociadas, Hallazgos.

Se obtuvo un punto de entrada para el modelamiento de ejercicios de proposiciones mediante el uso de un tutor de lógica, el cual permitía la comprobación de teoremas y resolución de ejercicios de proposiciones verificando paso a paso el valor de verdad de estas utilizando conectivos Lógicos.

Se obtuvieron nuevas ideas para dar soporte a los estudiantes mediante mini-lecciones las cuales presentan la teoría detrás del ejercicio mediante texto o videos al momento de que el estudiante tiene pasos difíciles de resolver.

Finalizando, este proyecto es el encargado del modelo del contenido, debido a que actualmente el modelo del aprendiz de la plataforma ya fue desarrollado. Este trabajo se centra en el "inner-loop" para dar retroalimentación de correcto/incorrecto y pistas (hints) en cada paso en la resolución del ejercicio

4. METODOLOGÍA DE DESARROLLO

4.1 Modelado Cognitivo de los ejercicios

Debido a los requisitos del proyecto, antes de comenzar el desarrollo se debió realizar el modelado de los “Ejercicios Tipo” para realizar este modelado se optó por la metodología de Modelado Cognitivo. El modelado cognitivo es un enfoque interdisciplinario que se centra en la creación de modelos teóricos o computacionales que representan y simulan los procesos mentales y cognitivos humanos. Estos modelos buscan capturar cómo las personas adquieren, procesan y utilizan la información, así como cómo toman decisiones y resuelven problemas en diversas situaciones. En esta metodología se llevan a cabo sesiones de análisis cognitivo para un set de ejercicios elegidos por el estudiante tesista, en conjunto con el profesor co-patrocinante. Durante estas reuniones, se abordan dudas y se destacan con ejemplos los pasos a seguir en la resolución de cada ejercicio (Steps). Finalmente, se elabora un diseño para simplificar la implementación del orden de dichos pasos.

4.2 Metodología de desarrollo

La metodología escogida para el desarrollo de software será una metodología *iterativa incremental*, este se basa en la repetición de ciclos de trabajo cortos e incrementales, con la mejora continua del producto a lo largo del tiempo. Este enfoque se utiliza comúnmente en la gestión de proyectos de software y está diseñado para abordar los desafíos asociados con la incertidumbre y los cambios en los requisitos del proyecto, lo cual es el caso de este proyecto debido a los posibles cambios en la implementación de los ejercicios de los diferentes tópicos. La estructura de los ciclos de trabajo tenía una duración de dos semanas. Durante este período, el estudiante tesista debía completar los avances establecidos en una reunión inicial con el profesor patrocinante. Al final del ciclo, se realizaba una segunda reunión para evaluar el progreso y definir mejoras si eran necesarias. Posteriormente, se organizaba una reunión con el profesor co-patrocinante, un docente de matemáticas que imparte la asignatura de Álgebra para la Ingeniería, con el fin de obtener retroalimentación y sugerencias de cambios. Estos ciclos fueron aplicados tanto al modelamiento de los ejercicios como al avance en la implementación. Adicionalmente, se celebraban reuniones semanales con el equipo de desarrollo de la plataforma para presentar avances y recibir ideas sobre el proceso de desarrollo. Otro punto importante en el desarrollo del proyecto es la realización de TutorCamps en los cuales se reunió al equipo anteriormente mencionado apoyar el desarrollo de otros miembros ya sea en integración, ayudas en el desarrollo de Front-End o Back-End o manejo de componentes ya desarrollados en la plataforma.

4.3 TutorCamp

Los TutorCamp son instancias de desarrollo en los cuales los miembros del equipo comparten sus conocimientos y ayudan en implementación o uso de la plataforma a otros miembros. Estos se han ido realizando a lo largo de los años de desarrollo de la plataforma.

Durante el transcurso de este proyecto de título se ha participado en dos de estos eventos con los siguientes objetivos:

TutorCamp 1° El objetivo general de este TutorCamp fue apoyar a tres proyectos los cuales estaban iniciando su desarrollo en la plataforma. Este es el caso de este proyecto por lo cual se comenzó por crear una rama del repositorio del proyecto en GitHub. Posteriormente un miembro del equipo proporcionó una plataforma de pruebas la cual fue utilizada para comenzar el desarrollo de forma local y se comenzó a trabajar en la estructura del tutor.

TutorCamp 2° El objetivo general es realizar un seguimiento de los proyectos y obtener retroalimentación por parte del resto de miembros del equipo. En lo que corresponde al avance específico de este proyecto comenzar el despliegue a la plataforma los avances realizados en la plataforma de pruebas.

4.4 Plataforma LearnerModelGQL

Learner Model GQL es un sistema basado en microservicios que utiliza *GraphQL* para gestionar y servir datos sobre el aprendizaje de usuarios en una plataforma educativa inteligente. Su propósito es estimar los niveles de conocimiento del usuario mediante la recopilación y análisis de evidencia de aprendizaje registrada en la plataforma. La arquitectura de microservicios permite la separación de servicios especializados (como usuarios, proyectos, contenido y acciones) que, a través de un *Gateway GraphQL*, se comunican mediante un único punto de entrada para simplificar el acceso a los datos y mejorar la eficiencia, escalabilidad y flexibilidad del sistema (Sáez, 2022).

Para implementar aplicaciones de front-end existe una plantilla que tiene como objetivo simplificar el uso de la plataforma y estandarizar las convenciones de los servicios proporcionados. Incluye una selección de librerías y un entorno de desarrollo recomendado, optimizando el desarrollo de aplicaciones en este ecosistema educativo inteligente. Entre las principales herramientas, se emplean *Next.js*, *TypeScript*, y *Chakra UI* para la creación de interfaces web, junto con *Auth0* para autenticación y *GraphQL* para la interacción con los servicios de acceso a datos. Estas herramientas serán explicadas a continuación.

4.4.1 Lenguaje de programación

JSX (<https://www.typescriptlang.org/docs/handbook/jsx.html>), es una extensión de sintaxis para JavaScript que permite escribir código que combina elementos HTML y JavaScript en una sola estructura. JSX es ampliamente usado en el mundo del desarrollo Web, en particular en aplicaciones creadas usando React.

4.4.2 Auth0

Auth0 (<https://auth0.com>), es una plataforma de autenticación y autorización que facilita la implementación de login seguro, autenticación basada en tokens, gestión de usuarios y características avanzadas como autenticación multifactor (MFA). Simplifica la integración de seguridad en aplicaciones al abstraer la gestión de credenciales y flujos de autenticación.

4.4.3 React

React (<https://es.react.dev>), es una biblioteca de JavaScript desarrollada por Meta para construir interfaces de usuario dinámicas y componibles. Utiliza un modelo basado en componentes y un Virtual DOM para optimizar la actualización y renderizado de elementos en la interfaz.

4.4.4 GraphQL

GraphQL (<https://graphql.org>), es un lenguaje de consulta y runtime que permite interactuar con una API de manera flexible y eficiente. A diferencia de REST, permite solicitar únicamente los datos específicos que necesita el cliente, reduciendo el volumen de información transferida y aumentando la eficiencia.

4.4.5 Framework

NextJs (<https://nextjs.org>), es un framework basado en Javascript y en la librería React que proporciona características adicionales como renderizado del lado del servidor, enrutamiento, optimización del rendimiento y gestión del estado.

4.4.6 Componentes

ChakraUI (<https://chakra-ui.com>), es una biblioteca de componentes simple, modular y accesible que proporciona componentes básicos para construir aplicaciones React.

4.4.7 Control de versiones

GitHub (<https://github.com>), es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. GitHub permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso.

4.4.8 Despliegue

Vercel (<https://vercel.com>), es una plataforma unificada en la nube que permite a los desarrolladores desplegar, gestionar y escalar sus aplicaciones y sitios Web. Esta

plataforma es particularmente popular para alojar aplicaciones creadas con frameworks de JavaScript como Next.js, React, Angular y Vue.js, y también es conocida por su velocidad y escalabilidad.

4.5 Validación

Para realizar la validación del tutor de lógica y conjuntos, se realizó una sesión de pruebas con usuarios estudiantes de primer año de ingeniería que estaban cursando la asignatura de *Álgebra para la Ingeniería*. Esta sesión se realizó con once estudiantes de primer año que fueron divididos en dos grupos para abordar los distintos tópicos del tutor, se comenzó por dar una descripción de la plataforma y el objetivo de la sesión de prueba y se hizo entrega de un consentimiento informado. Luego se dio comienzo a la sesión de prueba en la que los estudiantes debían realizar ejercicios de distintos tópicos en el tutor. Finalmente se aplicó un cuestionario SUS tener información sobre la usabilidad de la aplicación.

4.6 Supervisión de la creación de nuevo contenido

Se supervisó la creación de nuevo contenido para el tutor, ya que inicialmente solo se contaba con el set de ejercicios tipo previamente descrito. Para esta tarea, se solicitó la colaboración de otro estudiante del proyecto, quien ya había trabajado en la creación de ejercicios para otros tutores. A este estudiante se le proporcionó una plataforma de pruebas con los ejercicios modelados, se le explicaron las particularidades para modelar ejercicios en el tutor, y se le entregó una plantilla del archivo JSON modificado. Finalmente, los nuevos ejercicios fueron probados y enviados al profesor co-patrocinante de matemáticas para su aprobación y posterior subida a la plataforma.

5. MODELAMIENTO DEL CONTENIDO

En este capítulo se abordará el objetivo específico 2 que corresponde a selección de ejercicios a implementar seguido por la realización de su Modelado Cognitivo. Para llevar a cabo esta selección se trabajó con el Libro guía Unidad Lógica y Conjuntos (Facultad de Ciencias de la Ingeniería, 2023).

5.1 Modelado Cognitivo

El modelado cognitivo consiste en identificar los elementos en el procedimiento de resolución de una tarea o, en este caso un ejercicio. Para un sistema de tutoría inteligente, esto implica identificar los pasos de resolución con sus instrucciones o enunciados y sus posibles respuestas correctas, además de las habilidades implícitas a lograr en cada paso, y potenciales pistas o ayudas para cada paso.

Para realizar el modelado se formó un equipo con el profesor patrocinante y co-patrocinante, el cual es profesor de matemática y que actualmente realiza clases del curso *Álgebra para la Ingeniería*. Con el profesor de matemática se programaron reuniones cada 2 semanas en las cuales se realizaron las siguientes actividades para modelar cada ejercicio:

- Elección de los ejercicios tipo
- Desglose de la resolución en pasos
- Identificación de habilidades asociadas a los pasos del ejercicio
- Realización de pistas y desde mayor a menor abstracción
- Creación y revisión de prototipos no funcionales (mockups)
- Creación y revisión de métodos de ingreso de respuestas
- Implementación de un prototipo funcional
- Revisión de mejoras del prototipo

5.2 Tópicos de contenido seleccionados

Para la elección de tópicos se analizó el contenido que comprende la primera unidad del curso de *Álgebra para la Ingeniería*, revisando el Libro guía Unidad Lógica y Conjuntos. Se realizó una propuesta que abarcaba los siguientes tópicos: reconocimiento de proposiciones, utilización de conectivos lógicos para conjunciones, representación de intervalos, resolución de sucesiones, intersección y unión de conjuntos y comprobación de equivalencias lógicas utilizando tablas de verdad. Por lo tanto, el contenido del tutor son los diferentes ejercicios de estos tópicos. A continuación, se definirán los diferentes tópicos a ser abordados en el desarrollo.

5.2.1 Proposiciones

Una proposición es una afirmación o enunciado que es susceptible de ser verdadero o falso, pero no ambos al mismo tiempo. Es la unidad más básica de significado que se puede evaluar como verdadera o falsa.

5.2.2 Equivalencias Lógicas

Las equivalencias lógicas son relaciones entre proposiciones lógicas que tienen el mismo valor de verdad en todas las circunstancias. Dos proposiciones son consideradas equivalentes si siempre son verdaderas o siempre son falsas bajo las mismas condiciones de verdad. La relación de equivalencia lógica se denota con el símbolo \equiv .

Un ejemplo de equivalencias lógicas es el siguiente:

$$\overline{(\forall x \in U)(p(x))} \equiv (\exists x \in U)(\overline{p(x)})$$

5.2.3 Intervalos

Las formas de representar un intervalo son:

- La notación de intervalo es una convención matemática utilizada para expresar de manera compacta y precisa conjuntos de números reales en la recta numérica. Esta notación se utiliza para describir rangos continuos de valores y puede incluir o excluir extremos. Estos pueden ser de la siguiente forma:
 - Intervalo Abierto: $]a,b[$
 - Intervalo Cerrado: $[a,b]$
 - Intervalo Semiabierto a la izquierda: $]a,b]$
 - Intervalo Semiabierto a la derecha: $[a,b[$
- La representación de un intervalo en la recta numérica es una manera gráfica de mostrar todos los números reales que pertenecen al intervalo. La recta numérica es una línea en la que cada punto corresponde a un número real, y la representación de un intervalo proporciona una visualización de los valores que caen dentro de ese rango específico. La figura 2 muestra la representación de intervalos en la recta numérica



Figura 2. Representación de intervalos en la recta.

- La representación por comprensión de un conjunto de números reales implica describir las condiciones que deben satisfacer los elementos del conjunto para pertenecer a éste, sin enumerar cada elemento individualmente. La siguiente expresión representa un intervalo cerrado entre a, b por comprensión :
 $\{x \in \mathbb{R}: a \leq x \leq b\}$

Además, existen operaciones sobre intervalos algunas de ellas son:

- La unión de intervalos, que es una operación en teoría de conjuntos que combina dos o más intervalos para formar un nuevo intervalo que contiene todos los números presentes en los intervalos originales. La notación para la unión de intervalos es “ \cup ”.
- La intersección de intervalos es una operación en teoría de conjuntos que consiste en encontrar los elementos comunes a dos o más intervalos en la recta numérica. La notación para la intersección es “ \cap ”.

5.2.4 Sucesiones

Las sucesiones son una lista ordenada de números o términos que siguen un patrón específico. Cada término en la sucesión ocupa una posición única y se denota comúnmente por A_n , donde n es el índice que indica la posición en la secuencia. Su notación está representada de la siguiente manera:

$$(a_n)_{n \in \mathbb{N}} = (a_1, a_2, a_3, a_4, \dots)$$

5.2.5 Tablas de Verdad

Las tablas de verdad son una representación sistemática de todas las posibles combinaciones de valores de verdad que puede tomar una proposición compuesta o una función lógica. Un ejemplo de tabla de verdad en este caso la representación de la conjunción es la siguiente:

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

5.3 Definición de ejercicios tipo

En base a los tópicos anteriormente mencionados se buscó “ejercicios tipo” para cada uno, obteniendo un documento con un set base de ejercicios con los cuales comenzar las iteraciones. Véase la tabla 2 donde se muestra una definición de cada tópico.

Tópico	Descripción
Reconocimiento de proposiciones	<p>En este caso se definen diferentes tipos de enunciados y el objetivo es comprobar si son una proposición o no, en caso de serlo se debe obtener el valor de verdad de esta.</p> <p>Reconoce si la siguiente frase representa una proposición. En caso de serlo determine su valor de verdad</p> <ul style="list-style-type: none"> El número 1331 es divisible por 11
Utilización de conectivos Lógicos	<p>En este caso se plantean diferentes proposiciones, el estudiante debe verificar el valor de verdad de cada una y finalmente resolver conjunciones que poseen conectivos lógicos.</p> <p>Dadas las siguientes proposiciones</p> $p: \sqrt{3^2 + 4^2} = 3 + 4$ $q: \sqrt{3^2 \cdot 4^2} = 3 \cdot 4$ <p>Determine el valor de verdad de las siguientes conjunciones</p> $p \wedge q$
Representación de intervalos	<p>En este caso se plantea que el estudiante identifique el tipo de intervalo, este puede ser abierto, cerrado o semiabierto, además debe reconocer los valores extremos y finalmente debe representarlo ya sea en la recta, como conjunto o por compresión.</p> <p>Dado el siguiente intervalo represéntelo en la recta real y también por compresión</p> $[-5, 3]$
Resolución de sucesiones	<p>En este caso se plantea que el estudiante reconozca los elementos de una sucesión, los reemplace y luego represente los “n” primeros elementos de ésta.</p> <p>Dada la siguiente sucesión</p> $B_n = \frac{(-1)^n n^2 - n}{n^2 + 1}$ <p>Obtenga los valores de B_1</p>
Intersección y Unión	<p>Aquí se plantea que el estudiante reconozca los elementos de los intervalos para encontrar los elementos en común en el caso de la intersección o la unión de ambos intervalos.</p> <p>Encuentre C: que corresponde a la intersección de A con B</p> $A: [-3, 7] \cap B: [-1, 3]$
Comprobación de equivalencias lógicas utilizando tablas de verdad	<p>En este caso el estudiante debe verificar todas las posibles combinaciones de valores de verdad para cada conectivo lógico hasta obtener los valores de verdad de ambos lados de la equivalencia. Si ambos lados son iguales se satisface la equivalencia lógica.</p>

	<p>Utilizando una tabla de verdad compruebe que se satisface la siguiente equivalencia lógica:</p> $\overline{p} \Rightarrow \overline{q} \equiv p \wedge \overline{q}$
--	---

Tabla 2. Selección de Tópicos

Entonces, se escogen un ejercicio de cada Tópico para comenzar a realizar un análisis y diseño de éstos, para luego ir agregando nuevos ejercicios a medida que avanza el proyecto.

5.4 Mock-ups Ejercicios

La siguiente sección muestra el proceso de diseño de los ejercicios a ser desarrollados en el tutor. Este proceso fue realizado de forma iterativa debido a que tanto el diseño de los pasos como la identificación de habilidades podía cambiar debido a las retroalimentaciones del profesor co-patrocinante.

Para el diseño se optó por realizar tablas, las cuales se fueron modificando a lo largo del proyecto teniendo diferentes versiones. Para describir los cambios realizados se tomará como ejemplo el ejercicio del tópico de “Representación de intervalos”. Luego de esto, se describirá el análisis cognitivo en su versión final para el resto de los tópicos.

5.4.1 Representación de intervalos

El ejercicio seleccionado para la representación de intervalos fue:

Dada la siguiente representación gráfica, escriba el conjunto como intervalo y por comprensión.



La primera versión del ejercicio está representada en la tabla 3.

Paso1: Escriba el intervalo	
El intervalo está representado entre	<input type="text"/> <div> <input type="button" value="["/> <input type="button" value="]"/> <input type="button" value="∞"/> </div>
Represente el intervalo por comprensión	
El conjunto es : {	<input type="text"/>
<div> <input type="button" value="∈"/> <input type="button" value="ℝ"/> <input type="button" value="ℂ"/> <input type="button" value="ℕ"/> <input type="button" value="ℤ"/> <input type="button" value="<"/> <input type="button" value=">"/> <input type="button" value="≤"/> <input type="button" value="≥"/> </div>	}

Tabla 3. Primera iteración Representación de intervalos

Esta representación del ejercicio presentaba deficiencias en varios aspectos, ya que solo mostraba representaciones finales del ejercicio y dificultaba la labor de elección de pistas y habilidades asociadas a cada paso. Por lo tanto, se decidió cambiar esta representación por la mostrada en la tabla 4.

Paso	Resolución	Expresión	Respuesta	Hints	Habilidades
1	<p>Extremo Inferior <input type="text"/></p> <p>Extremo Superior <input type="text"/></p>	Reconozca los valores extremos de la recta	<p>Extremo inferior -3</p> <p>Extremo superior 7</p>	<p>Los valores extremos se representan a ambos lados de la recta</p> <p>El extremo inferior se representa a la izquierda y el superior a la derecha</p>	Reconocer los valores extremos del intervalo
2	(V F)	El valor -3 pertenece al conjunto	V	<p>Los extremos en un intervalo cerrado pertenecen al conjunto</p> <p>Dado que el intervalo es cerrado si pertenece al conjunto</p>	<p>Reconocer extremo cerrado</p> <p>Reconocer extremo Abierto</p> <p>Reconocer si el valor pertenece al intervalo</p>
3	<input type="text"/> <div>[] ∞</div>	Escriba el intervalo que representa la recta	[-3,7]	<p>Un intervalo extremo cerrado se representa con corchetes cerrados [</p> <p>El un intervalo cerrado se representa como [a,b] Siendo a y b los extremos inferior y superior respectivamente</p>	Representar una recta como intervalo
4	<p>El conjunto es : { <input type="text"/> }</p> <div>∈ ℝ ℂ ℕ ℤ < > ≤ ≥</div>	Represente el intervalo por comprensión	$x \in \mathbb{R} : -3 \leq x \leq 7$	<p>Por comprensión un intervalo cerrado se representa como {x ∈ conjunto: extremo inferior x ≤ extremo superior }</p>	Representar un conjunto por comprensión

Tabla 4. Segunda iteración Representación de intervalos

El cambio de la representación de la tabla se realiza debido a que es necesario identificar todos los aspectos del ejercicio en el proceso de modelado cognitivo. Se agregaron las siguientes columnas a la tabla: Resolución, Expresión, Respuesta, Hints, Habilidades.

La definición de habilidades es importante porque contribuye con el modelo de dominio y, por consiguiente, permite que el tutor implemente el soporte para el outer-loop; más concretamente, en la selección de contenidos.

Junto con este cambio en esta segunda iteración, se agregaron pasos intermedios para la resolución. La descripción de cada paso es la siguiente:

- *Paso 1* en el cual el estudiante debe reconocer los valores extremos de la representación gráfica esto mediante cuadros de texto en los cuales deberá escribir su respuesta.
- *Paso 2* en el cual debe reconocer si el valor pertenece o no al conjunto de números mediante el uso de Verdadero o Falso. En este paso se tienen las siguientes variantes:
 - El valor es un extremo de un intervalo cerrado
 - El valor es un extremo de un intervalo abierto
 - El valor no es un extremo

Por consiguiente, el paso tiene más de una habilidad asociada, y junto con esto, se pide como requisito que el valor no sea necesariamente un número entero, si no que pueda ser un valor decimal o una raíz debido a la dificultad de algunos estudiantes al momento de representar si este tipo de valores pertenecen o no a un conjunto.

- *Paso 3* El estudiante deberá representar el conjunto en notación de intervalo sabiendo previamente los valores extremos. Esto mediante un cuadro de texto. Además, se despliegan botones para facilitar el ingreso de caracteres especiales.
- *Paso 4* El estudiante deberá representar el conjunto en notación de compresión mediante un cuadro de texto, además del despliegue de botones para facilitar el ingreso de caracteres especiales.

A partir de este momento se definirán solo las iteraciones finales de los ejercicios a modelar.

5.4.2 Resolución de Sucesiones

El ejercicio seleccionado para la representación de intervalos fue:

Obtenga los elementos de la siguiente sucesión:

$$B_n = \frac{(-1)^n n^2 - n}{n^2 + 1}$$

Paso	Resolución	Expresión	Respuesta	Hints	Habilidades
1	$B_1 = \frac{(-1)^1 1^2 - 1}{1^2 + 1}$	Encuentre el primer valor de la sucesión	$B_1 = \frac{(-1)^1 1^2 - 1}{1^2 + 1}$	En este caso tienes que representar el primer valor de la serie	Reconocer la notación de sucesión
				Estamos calculando B_1	

				Debes reemplazar los valores de n en la expresión por 1	
2	$B_1 = (\quad)$	Encuentre el primer Valor de la sucesión	0		Ninguna habilidad
3	$0, \frac{2}{5}, \quad , \quad$	Escriba los primeros 4 términos de la sucesión.	$0, \frac{2}{5}, \frac{-3}{5}, \frac{12}{17}$		Calcular los elementos de una sucesión

Tabla 5. Representación de sucesión

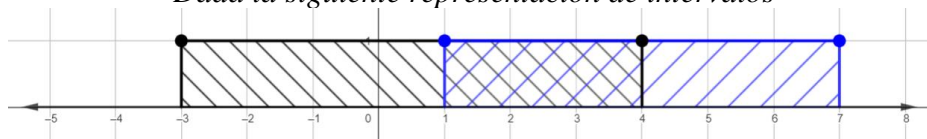
La descripción de cada paso es la siguiente:

- *Paso 1* El estudiante debe reconocer la notación de la sucesión y reemplazar los valores de n por el valor pedido. Esto mediante cuadros de texto (placeholders) en la sucesión en formato LaTeX
- *Paso 2* El estudiante debe obtener el valor de la sucesión obtenida anteriormente, esto mediante un cuadro de texto y pudiendo representar tanto valores enteros como fracciones.
- *Paso 3* El estudiante debe representar la secuencia de números obtenidos anteriormente. Esto mediante cuadros de texto.

5.4.3 Intersección y Unión de conjuntos

El ejercicio seleccionado para la representación de intervalos fue:

Dada la siguiente representación de intervalos



Paso	Resolución	Expresión	Respuesta	Hints	Habilidades
1	<ul style="list-style-type: none"> • [1,4] • [1,7] • [4,7] • [-3,1] • [-3,7] 	¿Cuál de los siguientes tramos en la gráfica representa la intersección de los 2 intervalos?	[1,4]	La intersección está representada por los elementos en común entre los intervalos La intersección corresponde a [1,4]	Reconocer la intersección en la grafica

2	Verdadero Falso	Reconozca si el valor siguiente pertenece a la intersección El valor es -3	Falso	Debes verificar si el valor pertenece a ambos intervalos	Reconocer los valores extremos del intervalo
3	$\{x \in : x\}$	Escriba por comprensión el intervalo que representa la intersección	$\{x \in R : 1 \leq x \leq 4\}$	La intersección está representada por los elementos en común entre los intervalos La intersección corresponde a $[1,4]$	Representar un intervalo por comprensión

Tabla 6. Representación de Intersección

La descripción de cada paso es la siguiente:

- *Paso 1* El estudiante debe reconocer la intersección de los intervalos, esto mediante selección múltiple.
- *Paso 2* en el cual debe reconocer si el valor pertenece o no al conjunto de números mediante el uso de Verdadero o Falso. En este paso se tienen las siguientes variantes:
 - El valor es un extremo de un intervalo cerrado
 - El valor es un extremo de un intervalo abierto
 - El valor no es un extremo
- *Paso 3* El estudiante debe escribir la notación por comprensión del intervalo, a diferencia del ejercicio de representación aquí se utilizan placeholders con una notación parcial a ingresar.

5.4.4 Comprobación de Equivalencias Lógicas utilizando tablas de verdad

El ejercicio seleccionado para la comprobación de equivalencias lógicas fue:

Utilizando una tabla de verdad compruebe que se satisface la siguiente equivalencia lógica:

$$\overline{p \Rightarrow q} \equiv p \wedge \overline{q}$$

Pas o	Resolución	Expresión	Respuesta	Hints	Habilidad e s																																								
1	<table><tr><th>p</th><th>q</th><th>$p \Rightarrow q$</th></tr><tr><td>V</td><td>V</td><td></td></tr><tr><td>V</td><td>F</td><td></td></tr><tr><td>F</td><td>V</td><td></td></tr><tr><td>F</td><td>F</td><td></td></tr></table>	p	q	$p \Rightarrow q$	V	V		V	F		F	V		F	F		Complete la siguiente tabla utilizando conectivos logicos	<table><tr><th>p</th><th>q</th><th>$p \Rightarrow q$</th></tr><tr><td>V</td><td>V</td><td>V</td></tr><tr><td>V</td><td>F</td><td>F</td></tr><tr><td>F</td><td>V</td><td>V</td></tr><tr><td>F</td><td>F</td><td>V</td></tr></table>	p	q	$p \Rightarrow q$	V	V	V	V	F	F	F	V	V	F	F	V	Debe utilizar el reciproco	Identificar conectivo lógico. Utilizar Reciproco										
p	q	$p \Rightarrow q$																																											
V	V																																												
V	F																																												
F	V																																												
F	F																																												
p	q	$p \Rightarrow q$																																											
V	V	V																																											
V	F	F																																											
F	V	V																																											
F	F	V																																											
2	<table><tr><th>p</th><th>q</th><th>$p \Rightarrow q$</th><th>$\overline{p \Rightarrow q}$</th></tr><tr><td>V</td><td>V</td><td>V</td><td></td></tr><tr><td>V</td><td>F</td><td>F</td><td></td></tr><tr><td>F</td><td>V</td><td>V</td><td></td></tr><tr><td>F</td><td>F</td><td>V</td><td></td></tr></table>	p	q	$p \Rightarrow q$	$\overline{p \Rightarrow q}$	V	V	V		V	F	F		F	V	V		F	F	V		Complete la siguiente tabla utilizando conectivos logicos	<table><tr><th>p</th><th>q</th><th>$p \Rightarrow q$</th><th>$\overline{p \Rightarrow q}$</th></tr><tr><td>V</td><td>V</td><td>V</td><td>F</td></tr><tr><td>V</td><td>F</td><td>F</td><td>V</td></tr><tr><td>F</td><td>V</td><td>V</td><td>F</td></tr><tr><td>F</td><td>F</td><td>V</td><td>F</td></tr></table>	p	q	$p \Rightarrow q$	$\overline{p \Rightarrow q}$	V	V	V	F	V	F	F	V	F	V	V	F	F	F	V	F	Debe utilizar la negación	Identificar conectivo lógico Utilizar Negación
p	q	$p \Rightarrow q$	$\overline{p \Rightarrow q}$																																										
V	V	V																																											
V	F	F																																											
F	V	V																																											
F	F	V																																											
p	q	$p \Rightarrow q$	$\overline{p \Rightarrow q}$																																										
V	V	V	F																																										
V	F	F	V																																										
F	V	V	F																																										
F	F	V	F																																										

3	<table><tr><td>p</td><td>q</td><td>$p \Rightarrow q$</td><td>$\overline{p} \Rightarrow \overline{q}$</td><td>$\overline{q}$</td></tr><tr><td>V</td><td>V</td><td>V</td><td>F</td><td></td></tr><tr><td>V</td><td>F</td><td>F</td><td>V</td><td></td></tr><tr><td>F</td><td>V</td><td>V</td><td>F</td><td></td></tr><tr><td>F</td><td>F</td><td>V</td><td>F</td><td></td></tr></table>	p	q	$p \Rightarrow q$	$\overline{p} \Rightarrow \overline{q}$	\overline{q}	V	V	V	F		V	F	F	V		F	V	V	F		F	F	V	F		Complete la siguiente tabla utilizando conectivos logicos	<table><tr><td>p</td><td>q</td><td>$p \vee q$</td><td>$\overline{p \vee q}$</td><td>\overline{p}</td><td>\overline{q}</td></tr><tr><td>V</td><td>V</td><td>V</td><td>F</td><td>F</td><td>F</td></tr><tr><td>V</td><td>F</td><td>V</td><td>F</td><td>F</td><td>V</td></tr><tr><td>F</td><td>V</td><td>V</td><td>F</td><td>V</td><td>F</td></tr><tr><td>F</td><td>F</td><td>F</td><td>V</td><td>V</td><td>V</td></tr></table>	p	q	$p \vee q$	$\overline{p \vee q}$	\overline{p}	\overline{q}	V	V	V	F	F	F	V	F	V	F	F	V	F	V	V	F	V	F	F	F	F	V	V	V	Debe utilizar la negación	Identificar conectivo lógico Utilizar Negación										
p	q	$p \Rightarrow q$	$\overline{p} \Rightarrow \overline{q}$	\overline{q}																																																																		
V	V	V	F																																																																			
V	F	F	V																																																																			
F	V	V	F																																																																			
F	F	V	F																																																																			
p	q	$p \vee q$	$\overline{p \vee q}$	\overline{p}	\overline{q}																																																																	
V	V	V	F	F	F																																																																	
V	F	V	F	F	V																																																																	
F	V	V	F	V	F																																																																	
F	F	F	V	V	V																																																																	
4	<table><tr><td>p</td><td>q</td><td>$p \Rightarrow q$</td><td>$\overline{p} \Rightarrow \overline{q}$</td><td>$\overline{q}$</td><td>$p \wedge \overline{q}$</td></tr><tr><td>V</td><td>V</td><td>V</td><td>F</td><td>F</td><td></td></tr><tr><td>V</td><td>F</td><td>F</td><td>V</td><td>V</td><td></td></tr><tr><td>F</td><td>V</td><td>V</td><td>F</td><td>F</td><td></td></tr><tr><td>F</td><td>F</td><td>V</td><td>F</td><td>V</td><td></td></tr></table>	p	q	$p \Rightarrow q$	$\overline{p} \Rightarrow \overline{q}$	\overline{q}	$p \wedge \overline{q}$	V	V	V	F	F		V	F	F	V	V		F	V	V	F	F		F	F	V	F	V		Complete la siguiente tabla utilizando conectivos logicos	<table><tr><td>p</td><td>q</td><td>$p \vee q$</td><td>$\overline{p \vee q}$</td><td>\overline{p}</td><td>\overline{q}</td><td>$\overline{p \wedge q}$</td></tr><tr><td>V</td><td>V</td><td>V</td><td>F</td><td>F</td><td>F</td><td>F</td></tr><tr><td>V</td><td>F</td><td>V</td><td>F</td><td>F</td><td>V</td><td>F</td></tr><tr><td>F</td><td>V</td><td>V</td><td>F</td><td>V</td><td>F</td><td>F</td></tr><tr><td>F</td><td>F</td><td>F</td><td>V</td><td>V</td><td>V</td><td>V</td></tr></table>	p	q	$p \vee q$	$\overline{p \vee q}$	\overline{p}	\overline{q}	$\overline{p \wedge q}$	V	V	V	F	F	F	F	V	F	V	F	F	V	F	F	V	V	F	V	F	F	F	F	F	V	V	V	V	Debe utilizar la conjunción	Identificar conectivo lógico Utilizar Conjunción
p	q	$p \Rightarrow q$	$\overline{p} \Rightarrow \overline{q}$	\overline{q}	$p \wedge \overline{q}$																																																																	
V	V	V	F	F																																																																		
V	F	F	V	V																																																																		
F	V	V	F	F																																																																		
F	F	V	F	V																																																																		
p	q	$p \vee q$	$\overline{p \vee q}$	\overline{p}	\overline{q}	$\overline{p \wedge q}$																																																																
V	V	V	F	F	F	F																																																																
V	F	V	F	F	V	F																																																																
F	V	V	F	V	F	F																																																																
F	F	F	V	V	V	V																																																																
5	<ul style="list-style-type: none">• Si, porque las columnas de las tablas son iguales• No, porque las tablas son diferentes• Si, porque El primer valor es verdadero• No, porque El primer valor es verdadero	Seleccione si se satisface la equivalencia	Si, porque las columnas de las tablas son iguales		Reconocer equivalencia a lógica																																																																	

Tabla 7. Representación Tablas de Verdad

La descripción de cada paso es la siguiente:

- Paso 1,2,3 Aquí el estudiante debe utilizar los distintos conectivos lógicos siendo estos: negación, conjunción, recíproco. Para la representación se despliega una tabla con 4 celdas vacías que se deben rellenar con V o F, utilizando las propiedades de estos conectivos lógicos, según corresponda.
- Paso 4 El estudiante debe reconocer si se satisface la equivalencia lógica. Esto mediante el despliegue de una selección múltiple.

5.4.5 Reconocimiento de proposiciones

El ejercicio seleccionado para el reconocimiento de proposiciones fue:

Reconoce si la siguiente frase representa una proposición. En caso de serlo, dé su valor de verdad:

6 es divisible por 2 y por 3 a la vez

Paso	Resolución	Expresión	Respuesta	Hints	Habilidades
1	<ul style="list-style-type: none"> • Si, ya que toda frase matemática es una proposición • No, pues no posee una condicion asociada • No, pues la proposición no está bien definida 	Reconozca si la frase es una proposición.	Si, pues posee una condición asociada	Una proposición es una afirmación o declaración que puede ser considerada	Reconocimiento de proposiciones

	<ul style="list-style-type: none"> • Si, pues se puede obtener valor numérico a partir de la frase 			verdadera o falsa.	
2	<ul style="list-style-type: none"> • Verdadero, ya que ambas condiciones son ciertas • Falso, pues 6 no es divisible por 2 • Falso, pues 6 no es divisible por 3 • Verdadero, pues ambas condiciones son falsas 	Determine el valor de verdad de la proposición	Verdadero, ya que ambas condiciones son ciertas	Si ambas condiciones son ciertas el valor de verdad de la proposición es verdadero	Reconocimiento del valor de verdad de la proposición

Tabla 8. Reconocimiento de Proposiciones

La descripción de los pasos es la siguiente:

- Paso 1 Aquí el estudiante debe seleccionar la alternativa que justifique si la frase es o no una proposición.
- Paso 2 El estudiante debe seleccionar la alternativa que justifique el valor de verdad de la proposición.

5.4.6 Utilización de conectivos lógicos para conjunciones.

El ejercicio seleccionado para utilización de conectivos lógicos para conjunciones siguiente:

Dadas las siguientes proposiciones

$$p: \sqrt{3^2 + 4^2} = 3 + 4$$

$$q: \sqrt{3^2 + 4^2} = 3 \cdot 4$$

Determine el valor de verdad de:

$$p \vee q$$

Paso	Resolución	Expresión	Respuesta	Hints	Habilidades
1	$p = \sqrt{3^2 + 4^2} =$ $3 + 4 =$	Resuelva ambos lados de la proposición.	5 7	La respuesta es 5 y 7	Evaluación de Proposiciones.
2	<ul style="list-style-type: none"> • Verdadero, ya que ambas condiciones son ciertas • Verdadero, pues ambas condiciones son falsas • Falso, pues 5 es distinto de 7 	Determine el valor de verdad de la proposición	Falso, pues 5 es distinto de 7	Si ambas condiciones son ciertas el valor de verdad de la proposición es verdadero	Reconocimiento del valor de verdad de la proposición

3	$q = \sqrt{3^2 \cdot 4^2} =$ $3 \cdot 4 =$	Resuelva ambos lados de la proposición.	12 12	La respuesta es 12 y 12	Evaluación de Proposiciones.
4	<ul style="list-style-type: none"> • Verdadero, ya que se cumple la igualdad. • Verdadero, pues ambas condiciones son falsas • Falso, pues 5 es distinto de 7 	Determine el valor de verdad de la proposición	Verdadero, ya que ambas condiciones son ciertas	Si ambas condiciones son ciertas el valor de verdad de la proposición es verdadero	Reconocimiento del valor de verdad de la proposición
5	<ul style="list-style-type: none"> • Ya que una de las 2 proposiciones es falsa, la disyunción es falsa • Ambas proposiciones son falsas por lo que la disyunción es falsa • Ambas proposiciones son verdaderas por lo que la disyunción es verdadera • Ya que una de las 2 proposiciones es verdadera, la disyunción es verdadera 	Determine el valor de verdad de la Disyunción $p \vee q$	Ya que una de las 2 proposiciones es verdadera, la disyunción es verdadera		Reconocimiento del valor de verdad de una disyunción

Tabla 9. Utilización de conectivos lógicos para conjunciones

La descripción de los pasos es la siguiente:

- Paso 1 Aquí el estudiante debe resolver ambos lados de la proposición
- Paso 2 El estudiante debe seleccionar la alternativa que justifique el valor de verdad de la proposición.
- Paso 3 y 4 Corresponden a la misma resolución de los pasos 1 y 2 respectivamente solo que para la proposición q.
- Paso 5 El estudiante debe seleccionar la alternativa que justifique el valor de verdad de la disyunción.

5.5 Requisitos

En esta sección se presentan los requisitos para el desarrollo del tutor de Lógica y teoría de conjuntos. Estos requisitos son el resultado del análisis de los pasos de resolución, habilidades y ayudas descritos en la sección anterior. Este set de requisitos se lista a continuación.

- Múltiples tipos ingreso de respuestas: Debido a la diferencia entre los pasos a realizar fue necesario tener diferentes tipos de interacción por parte de los usuarios los cuales serán los siguientes:
 1. Entrada de texto: Para escribir valores específicos (valores extremos, términos de una sucesión, notación de intervalos, etc.).
 2. Selección múltiple: Para elegir entre opciones (Verdadero/Falso, reconocimiento de intersección, valoración de proposiciones y conectivos lógicos).
 3. Tabla con celdas vacías: Para rellenar con valores V o F en ejercicios de tablas de verdad.
 4. Entrada de texto con placeholders y botones para caracteres especiales: Para facilitar el ingreso de respuestas en notación matemática o valores fraccionales.
- Soporte de Notación Matemática: El sistema debe soportar notación matemática avanzada (por ejemplo, LaTeX).
- Adaptación Iterativa: Posibilidad de modificar los ejercicios en función de retroalimentación docente.

6. IMPLEMENTACIÓN

En este capítulo se aborda el objetivo específico 3 que corresponde a construir un prototipo de tutor con una interfaz con enfoque responsivo, homogénea a las interfaces de otros tutores. Esta implementación comenzó en una plataforma de pruebas desarrollada por un miembro del equipo de desarrollo, lo que permitió un desarrollo en entorno aislado y controlado para realizar pruebas de los diferentes componentes.

6.1 Definición JSON

La información que define un ejercicio se estructura en un objeto JSON que denominaremos como “Contenido”, el cual ha sido utilizado por los otros tutores desarrollados en la plataforma. Fue necesario realizar algunas modificaciones en este formato en base a las necesidades del presente trabajo de título. El “Contenido” inicial posee toda la información del ejercicio a desplegar, esto mediante la conexión de los pasos, donde cada paso contiene la información al siguiente y último paso. En el último paso, la información del siguiente paso es nula. A continuación, se mostrará la estructura del “Contenido” final explicando los cambios respecto al modelo inicial.

Nombre	Descripción
code	Código del ejercicio
title	Título del tópico
type	Id para reconocer el despliegue en la plataforma
url	Url de las imágenes
meta	Campo libre para uso de metadatos
text	Texto del problema/planteamiento/ejercicio
steps	Contiene todo lo relacionado a los pasos del ejercicio y será un arreglo de objetos, donde cada objeto es un paso.

Tabla 10. Definición de los atributos de la estructura para los ejercicios

```
{
  "code": "Codigo Del ejercicio",
  "title": "Titulo del Topico",
  "type": "Id para reconocer el despliegue en la plataforma",
  "url": "url de las imagenes",
  "meta": {},
  "text": "Texto del problema/Ejercicio",
  "steps": [
  ]
}
```

Figura 2. Estructura general de un ejercicio.

```

"steps": [
{
  "stepId": "Id del paso",
  "expression": "Expresion del paso en caso de existir",
  "StepType": "Tipo de paso",
  "stepTitle": "Titulo del paso",
  "displayResult": [
    "Informacion a desplegar"
  ],
  "KCs": [
    "KCs Del ejercicio"
  ],
  "values": [],
  "hints": [
    {
      "hintId": "id del hint",
      "hint": "Hint"
    }
  ],
  "matchingError": [],
  "answers": [
    {
      "answer": [
        "Respuestas"
      ],
      "nextStep": "paso siguiente"
    }
  ],
  "incorrectMsg": "Mensaje en caso de respuesta incorrecta",
  "correctMsg": "Mensaje si la respuesta es correcta",
  "summary": "Mensaje luego de terminar el paso"
}
]

```

Figura 3. Estructura de los pasos.

Nombre	Definición
stepId	Identificador único del paso
Expresión	Expresión algebraica del paso.
StepType	Identificador del tipo de paso
stepTitle	Enunciado del paso.
displayResult	String del resultado de la expresión
KCs	Habilidades a trackear del estudiante
Values	Array de Valores para desplegar
Hints	Arreglo con pistas para el estudiante, cada elemento del arreglo es un objeto que representa un hint y contiene un identificador (hintId) y un enunciado de ayuda (hint).
MatchingError	Arreglo de errores específicos que tienen un hint específico.
Answers	Arreglo de objetos que contienen las respuestas correctas que deben ingresar los estudiantes y el identificador del siguiente paso del ejercicio.
incorrectMsg	Mensaje de error cuando el estudiante ingrese una respuesta incorrecta
correctMsg	Mensaje de validación cuando el estudiante realiza una respuesta correcta
Summary	Mensaje que aparece luego de terminar el paso.

Tabla 11. Estructura del contenido

Los nuevos campos son los siguientes:

- *Url*: Permite el despliegue de imágenes en caso de ser necesarias en el ejercicio. Esto permite flexibilidad a los componentes.
- *StepType*: Permite definir el tipo de paso para desplegar los diferentes componentes. Esto permite tener un código más limpio y hacer la aplicación más escalable.

6.2 Definición de Tipos

Typescript extiende la sintaxis de *JavaScript* añadiendo tipos estáticos, lo que permite desarrollar aplicaciones más robustas y mantenibles. En el contexto de este trabajo de título, la definición de tipos para los pasos ya estaba hecha. Sin embargo, fue necesario realizar algunas modificaciones para agregar los nuevos campos, y esto se realiza extendiendo las definiciones existentes. La figura 4 muestra la definición de tipos.

```
export interface value {
  name: string;
  value: number;
}
export interface hint {
  hintId: number;
  hint: string;
}
export interface matchingError {
  error: Array<string>;
  hintId: number;
  hint: string;
}
export interface answer {
  answer: Array<string>;
  nextStep: string;
}

export interface Step {
  stepId: string;
  KCs: Array<string>;
  expression: string;
  stepTitle: string;
  displayResult: Array<string>;
  values?: Array<value>;
  hints: Array<hint>;
  matchingError?: Array<matchingError>;
  validation?: "stringComparison" | "evaluate" | "countElements" | "evaluateAndCount";
  answers: Array<answer>;
  incorrectMsg: string;
  correctMsg: string;
  summary: string;
}

export interface ExType {
  code: string;
  meta: {};
  title: string;
  text: string;
  type: string;
  eqc?: string | undefined;
  steps: Array<Step>;
}

export interface StepLog extends Step {
  StepType: "Alternatives" | "TrueFalse" | "Sucesion" | "Blank" | "DualInputs" | "Extension" | "Notation" | "Rect" | "Table";
}

export interface ExLog extends ExType {
  url?: string | undefined;
  steps: Array<StepLog>;
}
```

Figura 4. Definición de Tipos

La interface *StepLog* se extiende de la anterior *Step* para agregar *StepType* definiendo los diferentes tipos de pasos. La interface *ExLog* se extiende de la anterior *ExType* para agregar la url de las imágenes y modificando la variable Steps que antes correspondía a un array del objeto Steps, cambiando estos por un array de *StepLog*.

6.3 Primera Iteración

En esta sección se definirán los componentes de la primera iteración de la aplicación. Esta iteración comprende el prototipo de la aplicación. En futuras iteraciones, se darán a conocer las modificaciones a este prototipo para su versión final. Estos componentes fueron modularizados para tener un código más limpio y legible y facilitar la corrección de errores.

6.3.1 Exercise

El componente principal de la aplicación es *Exercise*, que recibe como parámetro un objeto llamado *exc*, el cual es una instancia de la clase *ExLog* definida previamente. Este objeto se obtiene al cargar un Contenido desde el back-end de la plataforma *LearnerModelGQL*. El propósito del componente es mostrar el título del ejercicio, el enunciado del problema y, si están disponibles, las imágenes asociadas al planteamiento. Además, el componente llama a *StepComponent*, al que le pasa el objeto *exc* junto con el parámetro *nStep*, que representa el número de paso e inicia con el valor 0.

6.3.2 StepComponent

StepComponent recibe los parámetros *exc* correspondiente a la información del Contenido y *nStep* que es el número del paso, este realiza una comprobación del valor del número de paso puesto que si este es -1 despliega que el ejercicio está terminado, en caso contrario despliega el enunciado del paso, pudiendo estar éste en formato *LaTeX*. Posteriormente llama al componente *DisplayComponent*. La figura 5 muestra el código de *StepComponent*.

```
export function StepComponent({ exc, nStep }: { exc: ExLog; nStep: number }) {
  /* console.log(exc.steps[nStep].StepType)*/
  return (
    (nStep !== -1) ? (
      <>
      <Stack spacing={8} mb={2} direction='row'>
        <Box mr={1}>
          <Latex>{exc.steps[nStep].stepTitle}</Latex>
        </Box>
      </Stack>
      <DisplayComponent exc={exc} nStep={nStep}/>
    </>
  ):
  <>
    <Alert status='success'>
      <AlertIcon />
      Ejercicio Terminado
    </Alert>
  </>
);
}
```

Figura 5. Código *StepComponent*.

6.3.3 DisplayComponent

DisplayComponent recibe los parámetros *exc* correspondiente a la información del Contenido y *nStep* que corresponde al número del paso y tiene como función renderizar los diferentes subcomponentes dependiendo del tipo de paso. Esto se realizó como un

componente inicial para probar el despliegue de los componentes, siendo reemplazado posteriormente por *showSteps*. La figura 6 muestra el código de *DisplayComponent*.

```
export const DisplayComponent = ({ exc, nStep }: { exc: ExLog; nStep: number }) => {
  const type = exc.steps[nStep].StepType
  return type === "DualInputs"
    ? <DualInputs exc={exc} nStep={nStep} />
    : type === "TrueFalse"
      ? <TrueFalse exc={exc} nStep={nStep}/>
      : type === "Blank"
        ? <Blank exc={exc} nStep={nStep}/>
        : type === "Alternatives"
          ? <Alternatives exc={exc} nStep={nStep}/>
          : type === "Extension"
            ? <Extension exc={exc} nStep={nStep}/>
            : type === "Sucesion"
              ? <Sucesion exc={exc} nStep={nStep}/>
              : null
}
```

Figura 6. Código de DisplayComponent.

6.3.4 DualInputs

Este componente recibe los parámetros *exc* correspondiente al Contenido y *nStep* que es el número del paso, es utilizado en el paso 1 del tópico de representación de intervalos, el cual corresponde al reconocimiento de los valores extremos del intervalo. Éste renderiza la información de valores del Contenido y despliega 2 inputs para la entrada del usuario y un set de botones para facilitar la entrada de algunos símbolos como pueden ser las raíces cuadradas o el símbolo de infinito. Este componente posee 4 estados los cuales son un componente *useState* de *React* con las siguientes funciones:

- *Input1Value*, *Input2Value* su función es manejar la entrada del usuario para su posterior evaluación
- *firstTime* comprueba si es la primera vez que el usuario responde el paso
- *isCorrectValue* manejo de respuesta del usuario post-evaluación

Finalmente, está la función *evaluar* la cual comprueba la entrada del usuario cuando este pulsa el botón evaluar. Si la respuesta es correcta, se oculta la resolución del paso y se despliega el siguiente paso. Este despliegue actualiza el valor del siguiente paso y lo envía como *prompt* junto a *exc* al componente *StepComponent*. Si la respuesta es incorrecta se le da una retroalimentación al usuario. La Figura 7 muestra el despliegue en pantalla del componente.

Paso 1: Reconozca los valores extremos de la recta

Extremo Inferior

Extremo Superior

Figura 7. Despliegue *DualInputs*

6.3.5 *TrueFalse*

Este componente recibe los parámetros *exc* correspondiente al Contenido y *nStep* que es el número del paso. Su objetivo es la representación de pasos donde su respuesta pueda estar limitada por un verdadero o falso. Este componente despliega la información del paso seguido por 2 botones. Cuando el usuario hace click en alguno de estos se evalúa la respuesta. Si la respuesta es correcta, se oculta la resolución del paso y se despliega el siguiente paso. Este despliegue actualiza el valor del siguiente paso y lo envía como *prompt* junto a *exc* al componente *StepComponent*. Si la respuesta es incorrecta se le da una retroalimentación al usuario. La Figura 8 muestra el despliegue en pantalla del componente.

Paso 2: Reconozca si el valor pertenece al conjunto

El valor es:

$\sqrt{3}$

Figura 8. Despliegue *TrueFalse*

6.3.6 *Blank*

Este componente recibe los parámetros *exc* correspondiente al Contenido y *nStep* que es el número del paso y es utilizado en el paso 3 del tópico de representación de intervalos el cual corresponde al ingreso del intervalo. Éste renderiza la información de *values* del Contenido y despliega un input para la entrada del usuario y un set de botones para facilitar la entrada de algunos símbolos como pueden ser el infinito o paréntesis cuadrados. Este componente posee 4 estados los cuales son un componente *useState* de React con las siguientes funciones:

- *InputText* su función es manejar la entrada del usuario para su posterior evaluación
- *firstTime* comprueba si es la primera vez que el usuario responde el paso
- *isCorrectValue* manejo de respuesta del usuario posevaluación

Finalmente, está la función evaluar la cual comprueba la entrada del usuario cuando éste pulsa el botón evaluar. Si la respuesta es correcta, se oculta la resolución del paso y se despliega el siguiente paso. Este despliegue actualiza el valor del siguiente paso y lo envía como prompt junto a exc al componente *StepComponent*. Si la respuesta es incorrecta se le da una retroalimentación al usuario. La Figura 9 muestra el despliegue en pantalla del componente.

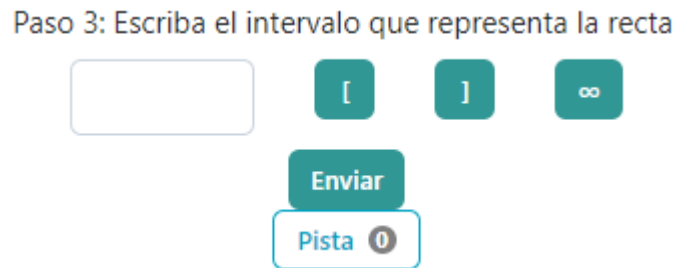


Figura 9. Despliegue Blank

6.3.7 InputButtons

Este componente recibe los parámetros exc correspondiente a la información del Contenido y *nStep* que es el número del paso y es utilizado en el paso 4 del tópico de representación de intervalos, el cual corresponde a la representación de intervalos por comprensión. Este componente renderiza la información de *values* del Contenido y despliega un input para la entrada del usuario y un set de botones para facilitar la entrada de algunos símbolos como pueden ser el infinito, conjuntos de números y los símbolos de $<$, $>$, \leq , \geq . El componente posee 3 estados los cuales son un componente useState de React con las siguientes funciones:

- *InputText* su función es manejar la entrada del usuario para su posterior evaluación
- *firstTime* comprueba si es la primera vez que el usuario responde el paso
- *isCorrectValue* manejo de respuesta del usuario posevaluación

Finalmente, está la función *evaluar* que comprueba la entrada del usuario cuando éste pulsa el botón evaluar. Si la respuesta es correcta, se oculta la resolución del paso y se despliega el siguiente paso. Este despliegue actualiza el valor del siguiente paso y lo envía como prompt junto a exc al componente *StepComponent*. Si la respuesta es incorrecta se le da una retroalimentación al usuario. La figura 10 muestra el despliegue en pantalla del componente.

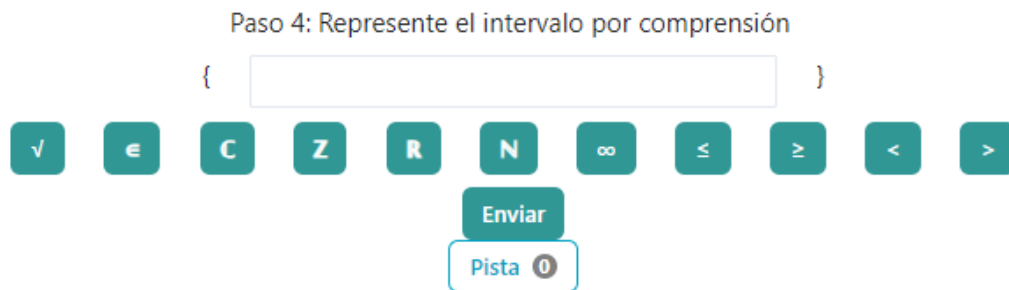


Figura 10. Despliegue *InputButtons*

6.4 Segunda Iteración

Esta segunda iteración comenzó con la necesidad del uso de “*Placeholders*”. Estos son marcadores temporales que indican espacios donde se debe ingresar contenido en expresiones matemáticas. Son útiles para mantener la estructura correcta de las fórmulas, especialmente al trabajar con fracciones, potencias o matrices en modo LaTeX. Estos placeholders se obtienen de la librería MathLive. Posterior a la revisión de esta librería se notó que se podía reemplazar algunos de los componentes anteriormente desarrollados por placeholders reduciendo la cantidad de componentes a utilizar y obteniendo un manejo más simple durante el desarrollo.

Para obtener una visión más clara de las interacciones entre los componentes y evitar el desarrollo innecesario de componentes que podrían ser reemplazados, el profesor patrocinante propuso la creación de un diagrama de componentes. Este diagrama no solo ayudo a visualizar de manera efectiva la arquitectura del sistema, sino que también permitió identificar con antelación posibles redundancias o incoherencias en el diseño. La figura 11 representa el diagrama de componentes.

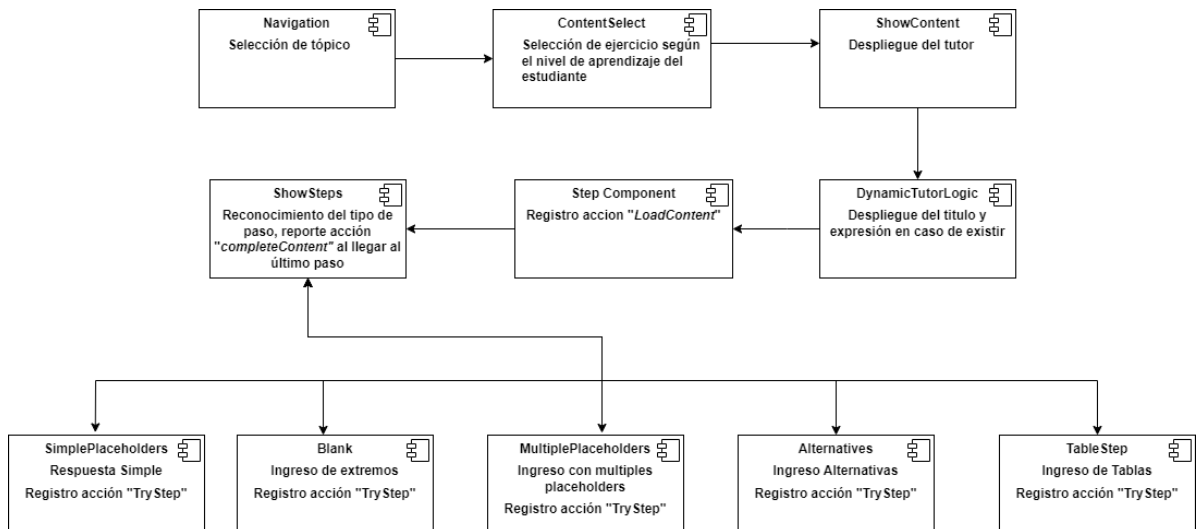


Figura 11. Diagrama de componentes

6.4.1 showSteps

Uno de los requisitos establecidos por el tutor fue que los pasos del ejercicio se desplegaran progresivamente conforme el estudiante avanza. Para cumplir con este objetivo, se decidió implementar un componente recursivo. Este componente recibe como parámetros el Contenido del ejercicio y *nStep*, que corresponde al número del paso que debe mostrarse. Además, utiliza tres elementos clave:

- *Step*, que indica el paso actualmente visible
- *setStep*, que permite actualizar ese valor para mostrar pasos anteriores si es necesario
- *topic* que corresponde al tópico del ejercicio.

Dentro del componente, se crea *completed*, un estado booleano manejado mediante un *useState* para controlar si el paso ha sido completado o no. Inicialmente, el valor de *completed* se establece en *false*. Luego se crea un *AccordionItem* de *Chakra UI*, que representa un elemento de acordeón capaz de expandirse o colapsarse. Este acordeón se ha personalizado para que, por defecto, el paso actual se mantenga expandido, aunque los pasos anteriores también pueden visualizarse si se hace clic en ellos. Posteriormente, se renderiza el título del paso correspondiente dentro del *AccordionItem* y se revisa una variable dentro del contenido llamada *StepType* esta contiene un identificador correspondiente al tipo de paso que debe ser desplegado. Utilizando este identificador se reconoce el componente a desplegar. Este componente recibe como parámetro la función *setCompleted*, lo que permite modificar el estado del paso actual cuando el estudiante lo completa.

Una vez que *completed* cambia a *true*, el componente verifica si existe un paso siguiente a través del campo *nextStep* dentro del Contenido. Si se encuentra un siguiente paso, el componente se llama a sí mismo de forma recursiva con los nuevos valores, permitiendo continuar con el flujo del ejercicio. En caso de que no haya más pasos disponibles, se notifica al estudiante que ha finalizado el ejercicio, y se reporta la acción *completeContent*. La figura 12 muestra un extracto del código de *showSteps* más específicamente el funcionamiento del estado *completed*.

```
{completed && next !== -1 ? (  
  <>  
    <Alert status="success">  
      <AlertIcon />  
      Ejercicio Terminado  
    </Alert>  
    {!changed ? (setColor("#C6F6D5"), setChanged(true)) : null}  
    {report ? (  
      <>  
        {action({  
          verbName: "completeContent",  
          contentID: exc.code,  
          topicID: topic,  
          result: 1,  
          extra: extras,  
        })}  
        {setReport(false)}  
      </>  
    ) : null}  
  </>  
): completed && next !== -1 ? (  
  <>  
    <ShowSteps exc={exc} nStep={next} Step={Step} setStep={setStep} topic={topic} />  
    {!changed ? (setColor("#C6F6D5"), setStep(next), setChanged(true)) : null}  
  </>  
): null}
```

Figura 12. Extracto de *showSteps*.

6.4.2 multiplePlaceholders

Este componente recibe como parámetros el contenido del ejercicio y *nStep*, que indica el número del paso que debe mostrarse. Además, emplea tres elementos clave:

- *Step*: Representa el paso actualmente visible.
- *setCompleted*: Permite actualizar el estado de finalización (*completed*) en el componente *showSteps*.
- *topic*: Corresponde al tópico del ejercicio.

Este componente se encarga de renderizar expresiones matemáticas en formato LaTeX y gestionar entradas tipo placeholder mediante el uso del componente *mathFieldElement* de la librería *mathLive*. Las respuestas del usuario se almacenan en el estado local mediante *ValuesArray*, que se actualiza automáticamente al detectar cambios en los placeholders. Una vez que el usuario ingresa una respuesta, puede hacer clic en el botón "Enviar", que llama a la función *evaluar*. Esta función compara el contenido de *ValuesArray* con la respuesta esperada definida en el contenido del paso. Dado que algunas respuestas pueden escribirse de distintas formas equivalentes, la comparación puede realizarse de dos maneras, dependiendo del valor del parámetro *validation* presente en el contenido:

1. Si el parámetro es "Evaluate": Se comparan los valores resultantes de ambas respuestas. Para realizar esta evaluación se utilizan las siguientes funciones:
convertirNotacion: Convierte la entrada a un formato estándar para representar fracciones o raíces cuadradas en LaTeX. Si no encuentra coincidencias relevantes, devuelve la entrada original. Esta función utiliza expresiones regulares para identificar fracciones y raíces, unificando así el formato de las respuestas.
MQPostfixparser: Transforma la expresión ingresada a notación postfija.
MQPostfixSolver: Evalúa el valor de la expresión en notación postfija.
Cada elemento de la lista es procesado y evaluado utilizando este flujo.
Las funciones *MQPostfixparser* y *MQPostfixSolver* estaban desarrolladas en la plataforma y fueron integradas a este componente.
2. Si el parámetro *validation* no es "Evaluate": La comparación se realiza verificando si ambas respuestas son idénticas.

Una vez realizada la comparación la función *evaluar* también registra la cantidad de veces que se realizó el paso y realiza el reporte de acción *TryStep*.

Este enfoque permite manejar respuestas equivalentes y mejorar la flexibilidad en la evaluación de los ejercicios. La versatilidad de las expresiones LaTeX, junto con la gestión eficiente de respuestas, ha permitido sustituir componentes como *DualInputs* e *InputButtons*, dado que *mathFieldElement* incorpora su propio teclado en pantalla, facilitando la entrada de símbolos matemáticos. La figura 13, 14 y 15 muestran diferentes usos del componente.

Paso 1: Represente el término indicado de la sucesión ^

$$B_1 = \frac{(-1)^n \cdot (n^2 - n)}{n^2 + 1}$$

enviar Pista ⓘ

Figura 13. Despliegue para sucesión.

Paso 1: Reconozca los valores extremos del intervalo ^

Extremo Inferior ☐ Extremo Superior ☐

enviar Pista ⓘ

Figura 14. Despliegue para reconocimiento de extremos.

Paso 4: Represente el intervalo por comprensión ^

$x \in$ $:$ $x \in$

enviar Pista ⓘ

Figura 15. Despliegue en intervalo por comprensión.

6.4.3 singlePlaceholder

Este componente recibe como parámetros el contenido del ejercicio y $nStep$, que indica el número del paso que debe mostrarse. Además, emplea tres elementos clave:

- *Step*: Representa el paso actualmente visible.
- *setCompleted*: Permite actualizar el estado de finalización (*completed*) en el componente *showSteps*.
- *topic*: Corresponde al tópico del ejercicio.

Este componente es utilizado en pasos que requieren solo un ingreso de *placeholder* como respuesta, en lugar de utilizar el componente anterior se decidió crear un componente extra en caso de que solo se tenga una respuesta. La lógica de evaluación del componente es la misma que *multiplePlaceholders*, pero en lugar de usar una lista, se evalúa solo un elemento. Una vez realizada la comparación la función evaluar también registra la cantidad de veces que se realizó el paso y realiza el reporte de acción *TryStep*. La figura 16 muestra el despliegue en pantalla del componente *singlePlaceholder*.

Paso 2: Encuentre el primer valor de la sucesión, si no es entero, represéntelo como fracción de la forma $\frac{a}{b}$

$B_1 =$

enviar
Pista

Figura 16. Despliegue *singlePlaceholder*.

6.4.4 tableStep

Este componente recibe como parámetros el contenido del ejercicio y *nStep*, que indica el número del paso que debe mostrarse. Además, emplea tres elementos clave:

- *Step*: Representa el paso actualmente visible.
- *setCompleted*: Permite actualizar el estado de finalización (*completed*) en el componente *showSteps*.
- *topic*: Corresponde al tópico del ejercicio.

Este componente se encarga de renderizar tablas cuya información se encuentra en el contenido mediante el uso del componente *table* de *chakraUI*. Debido a que fue diseñado para ejercicios de Tablas de Verdad los encabezados de las columnas pueden ser expresiones matemáticas o texto y las respuestas del usuario se manejan mediante botones de estado los cuales pueden tener valor V o F y se cambian al hacer click. Al hacer click en el botón enviar se llama a la función *evaluar*. La función *evaluar* se encarga de comparar la respuesta del usuario con la respuesta esperada definida en el contenido, registra la cantidad de intentos del usuario al realizar el paso y realiza el reporte de acción *TryStep*. La figura 17 muestra el despliegue del componente *tableStep*.

Paso 1: Complete la tabla de verdad para la siguiente expresión

La expresión es: $p \Rightarrow q$

p	q	$p \Rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Paso 1

Enviar
Pista

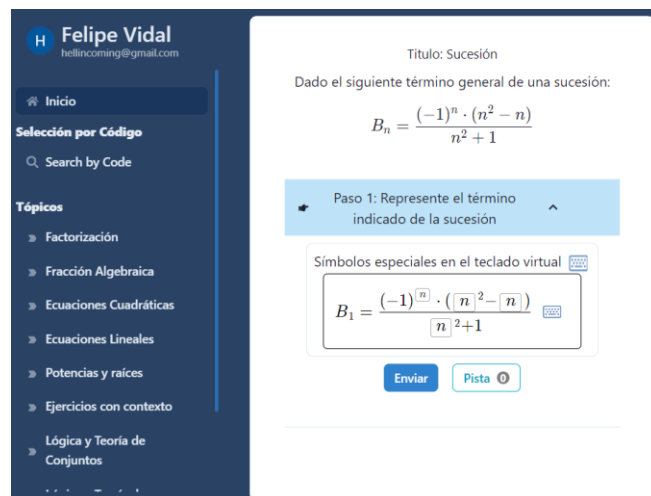
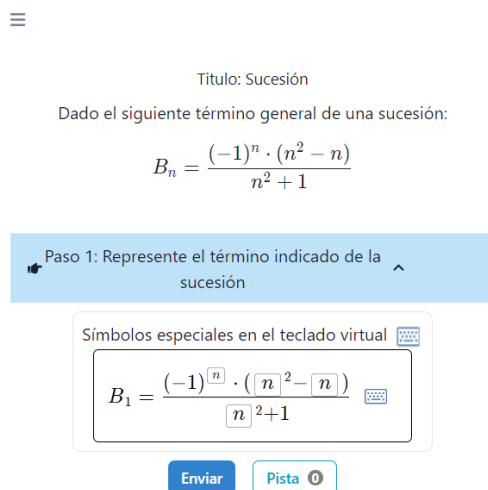
Figura 17. Despliegue de tablas.

7. Responsividad en la Aplicación

Este capítulo forma parte del O.E.3: “Construir un prototipo de tutor con una interfaz con enfoque responsivo, homogénea a las interfaces de otros tutores.” Aquí se aborda el diseño responsivo de la interfaz, que quedó pendiente durante el desarrollo inicial debido a que se priorizó la funcionalidad y los componentes recibían constantes cambios en su despliegue. El desarrollo de una interfaz responsiva es una característica importante, ya que permite que la aplicación se adapte de manera óptima a diferentes dispositivos, lo cual es fundamental para los estudiantes. Al contar con una interfaz responsiva, los estudiantes pueden acceder al tutor desde dispositivos como teléfonos móviles, tabletas y computadoras portátiles, sin experimentar problemas de visualización o navegación. Esto mejora la accesibilidad y facilita el aprendizaje, permitiendo que los estudiantes interactúen de manera cómoda y efectiva con la aplicación, independientemente del dispositivo que utilicen. A continuación, se listarán los componentes y sus despliegues en diferentes resoluciones correspondientes a teléfono, Tablet y computadora.

7.1 multiplePlaceholders

En las siguientes figuras podemos ver como se ve el paso en teléfono, Tablet y computadora respectivamente, además se agregó una leyenda al componente para especificar a los estudiantes el uso del teclado de pantalla que permite el ingreso de símbolos especiales.



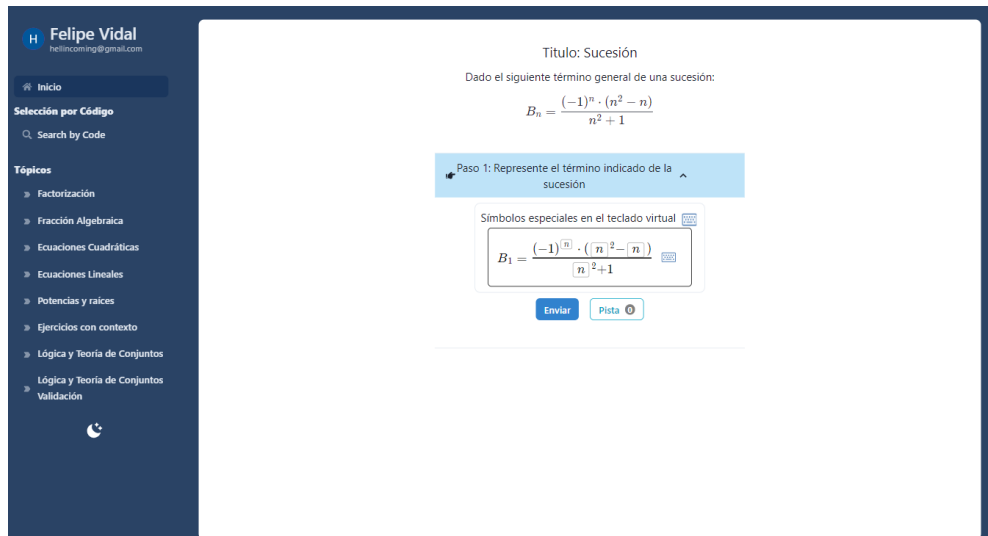
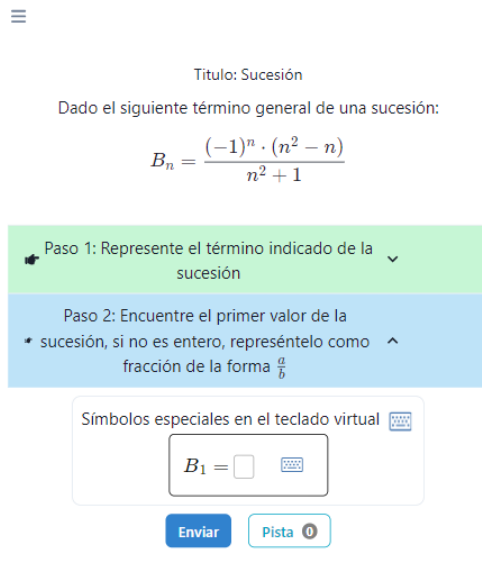


Figura 18. Representación *multiplePlaceholders* en diferentes resoluciones

7.2 singlePlaceholders

En las siguientes figuras podemos ver como se ve el paso en teléfono, Tablet y computadora respectivamente.



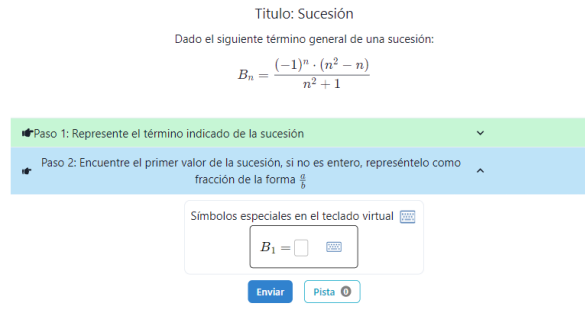
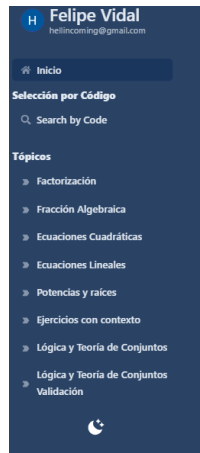
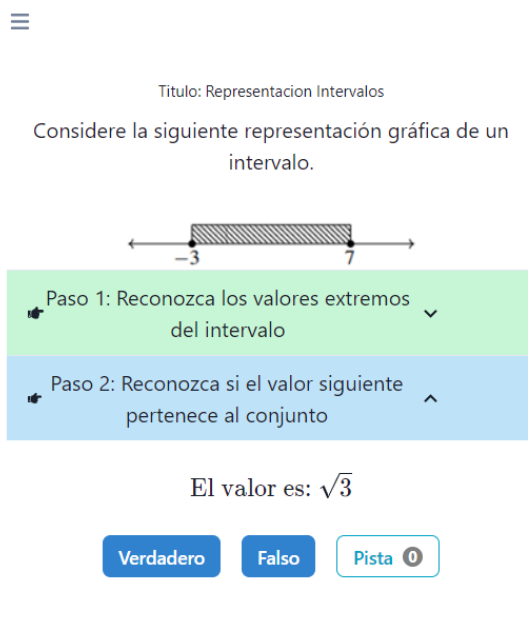


Figura 19. Representación *singlePlaceholders* en diferentes resoluciones

7.3 TrueFalse

En las siguientes figuras podemos ver como se ve el paso en teléfono, Tablet y computadora respectivamente.



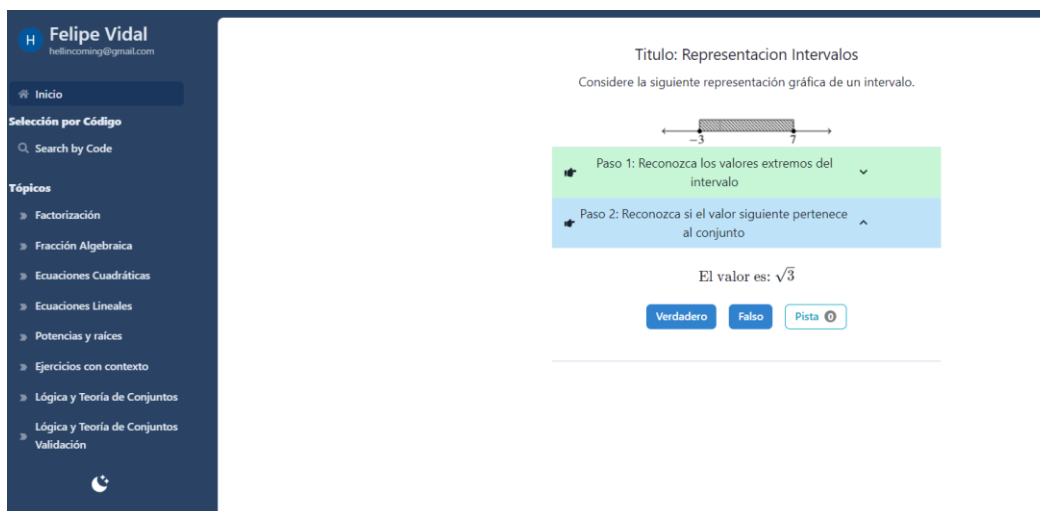
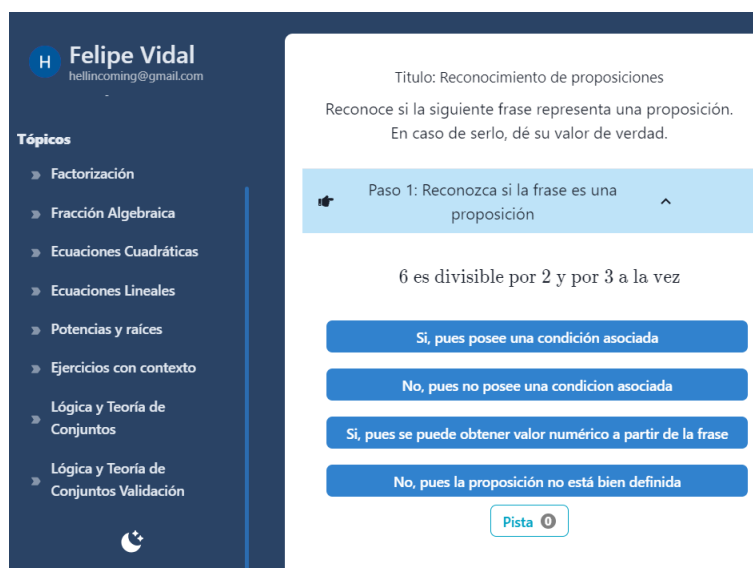
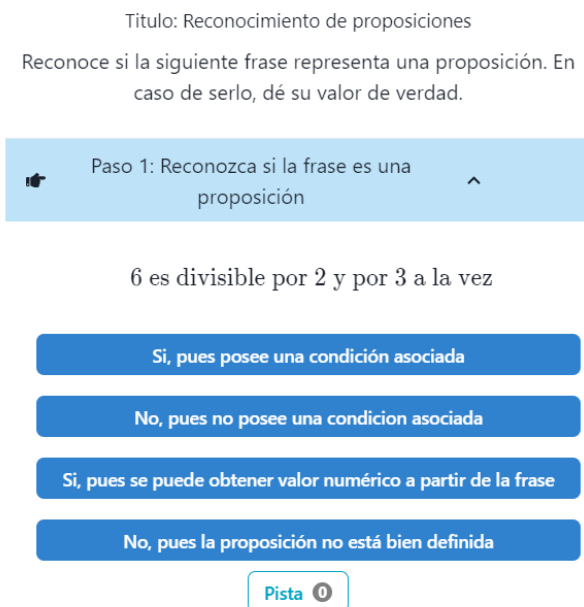


Figura 20. Representación *trueFalse* en diferentes resoluciones

7.4 Alternatives

En las siguientes figuras podemos ver como se ve el paso en teléfono, Tablet y computadora respectivamente.




Felipe Vidal
 hellincomng@gmail.com

Tópicos

- » Factorización
- » Fracción Algebraica
- » Ecuaciones Cuadráticas
- » Ecuaciones Lineales
- » Potencias y raíces
- » Ejercicios con contexto
- » Lógica y Teoría de Conjuntos
- » Lógica y Teoría de Conjuntos Validación

Título: Reconocimiento de proposiciones

Reconoce si la siguiente frase representa una proposición. En caso de serlo, dé su valor de verdad.

Paso 1: Reconozca si la frase es una proposición

6 es divisible por 2 y por 3 a la vez

[Pista](#)

Figura 21. Representación *Alternatives* en diferentes resoluciones

7.5 TableStep

En las siguientes figuras podemos ver como se ve el paso en teléfono, Tablet y computadora respectivamente.

≡

Título: Tablas de Verdad

Utilizando una tabla de verdad compruebe que se satisface la siguiente equivalencia lógica:

$$\overline{p} \Rightarrow \overline{q} \equiv p \wedge q$$

Paso 1: Complete la tabla de verdad para la siguiente expresión

La expresión es: $p \Rightarrow q$

p	q	$p \Rightarrow q$
V	V	<input type="button" value="V"/>
V	F	<input type="button" value="V"/>
F	V	<input type="button" value="V"/>
F	F	<input type="button" value="V"/>

Paso 1


Felipe Vidal
 hellincomng@gmail.com

Inicio

Selección por Código

Search by Code

Tópicos

- » Factorización
- » Fracción Algebraica
- » Ecuaciones Cuadráticas
- » Ecuaciones Lineales
- » Potencias y raíces
- » Ejercicios con contexto
- » Lógica y Teoría de Conjuntos

Título: Tablas de Verdad

Utilizando una tabla de verdad compruebe que se satisface la siguiente equivalencia lógica:

$$\overline{p} \Rightarrow \overline{q} \equiv p \wedge q$$

Paso 1: Complete la tabla de verdad para la siguiente expresión

La expresión es: $p \Rightarrow q$

p	q	$p \Rightarrow q$
V	V	<input type="button" value="V"/>
V	F	<input type="button" value="V"/>
F	V	<input type="button" value="V"/>

Felipe Vidal
hellincoming@gmail.com

Inicio

Selección por Código

Search by Code

Tópicos

Factorización

Fracción Algebraica

Ecuaciones Cuadráticas

Ecuaciones Lineales

Potencias y raíces

Ejercicios con contexto

Lógica y Teoría de Conjuntos

Lógica y Teoría de Conjuntos

Validación

Titulo: Tablas de Verdad

Utilizando una tabla de verdad compruebe que se satisface la siguiente equivalencia lógica:

$$\bar{p} \Rightarrow q \equiv p \wedge q$$

Paso 1: Complete la tabla de verdad para la siguiente expresión

La expresión es: $p \Rightarrow q$

p	q	$p \Rightarrow q$
V	V	V
V	F	V
F	V	V
F	F	V

Paso 1

Enviar

Pista

Figura 22. Representación *tableStep* en diferentes resoluciones

7.6 Blank

En las siguientes figuras podemos ver como se ve el paso en teléfono, Tablet y computadora respectivamente.

Felipe Vidal
hellincoming@gmail.com

Inicio

Selección por Código

Search by Code

Tópicos

Factorización

Fracción Algebraica

Ecuaciones Cuadráticas

Ecuaciones Lineales

Potencias y raíces

Ejercicios con contexto

Titulo: Representacion Intervalos

Considere la siguiente representación gráfica de un intervalo.

Paso 1: Reconozca los valores extremos del intervalo

Paso 2: Reconozca si el valor siguiente pertenece al conjunto

Paso 3: Escriba el intervalo representado en la recta

[-3,7]

Enviar

Pista

Felipe Vidal
hellincoming@gmail.com

Inicio

Selección por Código

Search by Code

Tópicos

Factorización

Fracción Algebraica

Ecuaciones Cuadráticas

Ecuaciones Lineales

Potencias y raíces

Ejercicios con contexto

Titulo: Representacion Intervalos

Considere la siguiente representación gráfica de un intervalo.

Paso 1: Reconozca los valores extremos del intervalo

Paso 2: Reconozca si el valor siguiente pertenece al conjunto

Paso 3: Escriba el intervalo representado en la recta

[-3,7]

Enviar

Pista

8. EVALUACIÓN

Este capítulo aborda el objetivo específico “OE5. Evaluar el tutor con un grupo de estudiantes para tener retroalimentación de su usabilidad.”. Para esto se realizó una prueba de validación con un total de 11 estudiantes, los cuales fueron divididos en dos grupos para la realización de ejercicios. El primer grupo realizó ejercicios de *Reconocimiento de proposiciones*, *Intersección de intervalos*, y *Tablas de Verdad*. El segundo grupo realizó ejercicios de *Proposiciones y Conjunciones*, *Unión de intervalos* y *Sucesiones*. Posterior a la resolución de los ejercicios planteados se entregó un cuestionario y una tabla a rellenar en caso de tener problemas con alguna de las implementaciones. A continuación, se muestran los campos a completar y la Tabla:

- ¿Hubo alguna instrucción que te pareció confusa?
- ¿Encontraste alguna pista que te pareció confusa?
- ¿Tuviste algún problema al ingresar tu respuesta?, Si es así indica donde

No	Alternativas	Verdadero Falso	Ingreso Intervalos	Reemplazo de variables en una ecuación	Ingreso LaTeX

Tabla 12. Validación

Una vez finalizada la sección de resolución de ejercicios, se les entregó un cuestionario SUS, compuesto por diez aseveraciones que se responden en un nivel de acuerdo de 5 puntos, desde *totalmente en desacuerdo* hasta *totalmente de acuerdo*. Estas preguntas se listan a continuación:

1. Creo que usaría este sistema frecuentemente.
2. Encuentro este sistema innecesariamente complejo.
3. Creo que el sistema fue fácil de usar .
4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar este sistema.
5. Las funciones de este sistema están bien integradas.
6. Creo que el sistema es muy inconsistente.
7. Imagino que la mayoría de la gente aprendería a usar este sistema en forma muy rápida.
8. Encuentro que el sistema es muy difícil de usar.
9. Me siento confiado al usar este sistema.
10. Necesité aprender muchas cosas antes de ser capaz de usar este sistema

Una vez realizada la prueba se obtuvieron los siguientes resultados:

Pregunta	Promedio
P1	4.55
P2	1.82
P3	4.64
P4	1.64
P5	4.55
P6	1.73
P7	4.73
P8	1.55
P9	4.36
P10	1.64

Tabla 13. Resultados

Según la metodología proporcionada por la métrica SUS, se sumaron las puntuaciones de las preguntas impares y se les restó 5, mientras que a las puntuaciones de las preguntas pares se les restó 25. Esto se debe a que se consideran positivas las preguntas impares y negativas las pares (Brooke, 1996). Luego, para obtener la puntuación final del cuestionario SUS, se sumaron ambos resultados y se multiplicaron por 2.5. En esta escala, la puntuación máxima es 100 y la mínima es 0.

Respuestas Positivas: $(4.55+4.64+4.55+4.73+4.36) - 5 = 17.83$

Respuestas Negativas: $25 - (1.82+1.64+1.73+1.55+1.64) = 16.62$

Puntuación SUS: $(17.83+16.62) * 2.5 = 86.13$

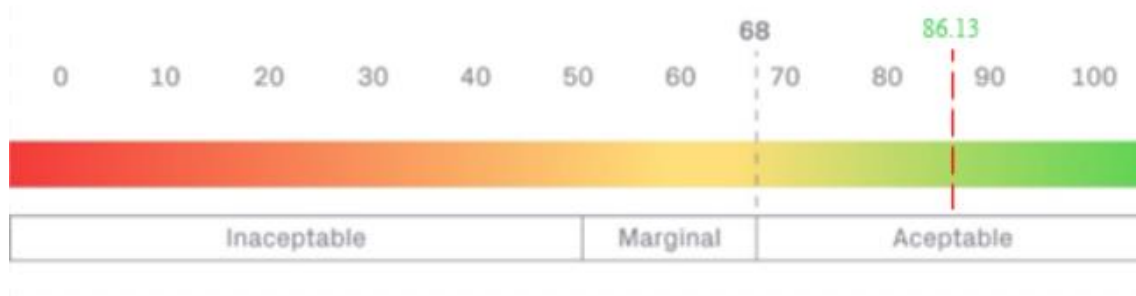


Figura 24. Rangos de la puntuación SUS.

Como podemos ver en la figura anterior el resultado de la puntuación SUS está en un rango aceptable, esto también concuerda con la retroalimentación recibida por parte de los estudiantes al final del laboratorio.

Al analizar esta retroalimentación, identificamos que el sub-tópico con mayor cantidad de errores fue la *Representación de intervalos*, estos problemas fueron principalmente la dificultad de encontrar símbolos como $<$, $>$ pues estos se ingresan mediante un teclado de pantalla el cual los estudiantes no estaban utilizando. En particular, las dificultades se concentraron en el paso relacionado con la *Representación de conjuntos por comprensión*.

9. CONCLUSIONES

En este trabajo se abordó la implementación de un tutor inteligente para lógica y teoría de conjuntos mediante el desarrollo de nuevos componentes en la plataforma *LearnerModelGQL*, más el modelamiento de ejercicios para lógica y teoría de conjuntos del curso *Álgebra para la Ingeniería*; curso que está presente en la malla curricular de primer año en las carreras de ingeniería de la Universidad Austral de Chile. *LearnerModelGQL* es una plataforma de microservicios que permite la implementación de tutores inteligentes. Los tutores inteligentes presentes en la plataforma están orientados a la nivelación de matemáticas y financiados por el proyecto Fondecyt 11220709.

Se realizó una revisión de la literatura de tutores inteligentes en esta área y se obtuvieron los siguientes hallazgos: un punto de entrada para el modelamiento de los ejercicios de proposiciones mediante el uso de un tutor de lógica, obteniendo un modelado de ejercicios de proposiciones mediante el uso de alternativas. Se adaptó la idea de las mini-lecciones para el desarrollo de pistas que presentan la teoría detrás del ejercicio.

El modelado de los ejercicios implicó la identificación de un "Ejercicio Tipo", es decir, un ejercicio representativo de cada tópico a abordar. A partir de este ejercicio, se descompuso la resolución en pasos, se identificaron las habilidades clave requeridas en cada etapa y se diseñaron pistas para facilitar su resolución. Este trabajo se llevó a cabo en colaboración con un profesor de matemáticas encargado del curso de Álgebra para Ingeniería. El proceso se aplicó a los siguientes sub-tópicos: Proposiciones, Equivalencias Lógicas, Intervalos, Sucesiones y Tablas de Verdad. Cada iteración incluyó ajustes en los ejercicios desarrollados, evaluación de mock-ups, propuestas para el formato de entrada de respuestas y pruebas de un prototipo funcional. Una vez aprobado el prototipo, se procedió con la implementación en la plataforma. Finalmente obtuvo un set de 31 ejercicios con variaciones a partir de 6 ejercicios inicialmente modelados entre todos los sub-tópicos anteriormente mencionados. Los ejercicios, al definirse completamente en un formato JSON regular son fácilmente replicables.

Utilizando una metodología de desarrollo iterativa incremental se creó un nuevo tutor para la plataforma y nuevos componentes que permiten el ingreso de diferentes tipos de respuesta, como por ejemplo el uso de placeholders los cuales permiten flexibilidad a los ejercicios, mediante el ingreso de campos vacíos a expresiones matemáticas en formato *LaTeX*. Además, se implementó un teclado en pantalla que permite el ingreso de nuevas expresiones matemáticas como pueden ser derivadas, integrales y sumatorias. Esta metodología de desarrollo fue beneficiosa ya que las iteraciones se realizaban en paralelo al modelamiento, logrando así que una vez se terminaba el modelado de un ejercicio se comenzaba a trabajar en el desarrollo de los componentes asociados a sus pasos.

Para realizar la validación de la aplicación y los ejercicios, se realizó una sesión de validación, ésta fue conformada por once estudiantes, que se encontraban realizando el curso de *Álgebra para la Ingeniería*. En esta sesión, se dividió el grupo en dos, para que los estudiantes realicen diferentes tipos de ejercicios durante 45 minutos. En ese tiempo,

el estudiante tesista resolvió dudas que los estudiantes tenían mientras realizaban los ejercicios. Finalmente, se les hizo entrega de dos cuestionarios, un cuestionario sobre los ejercicios que debieron realizar y un cuestionario SUS (System Usability Scale). Luego del análisis de ambos cuestionarios, se detectaron ciertos problemas a resolver en la aplicación. Esta validación permitió encontrar problemas y limitaciones. Los principales problemas encontrados fueron errores de tipeo y problemas para encontrar expresiones matemáticas en los ejercicios de placeholders, pues, si bien estos tenían un teclado virtual para el ingreso de dichas expresiones, los estudiantes no hacían uso de éste. Para resolverlo se agregó un texto en este tipo de ejercicios, describiendo la funcionalidad de este teclado. Las principales limitaciones de esta sesión de validación fueron la cantidad de estudiantes de la sesión y su manejo del tópico, pues a pesar de ya haber cursado esa unidad del curso, algunos cometían errores al no reconocer lo que se estaba preguntando.

Como parte del trabajo futuro, se identifican varios aspectos por mejorar. En primer lugar, es fundamental trabajar en la estandarización del formato de los JSON, ya que actualmente existen múltiples formatos utilizados por los diferentes tutores, lo que dificulta la compatibilidad de componentes entre los ejercicios pertenecientes a diferentes tutores. Además, sería beneficioso implementar interactividad con imágenes, para enriquecer el ingreso de respuestas en ejercicios como Unión, Intersección y Representación de Intervalos, lo que mejoraría la experiencia del usuario. También se destaca la importancia de asegurar la compatibilidad con ejercicios de otros tutores, permitiendo simplificar y optimizar el proceso de ingreso de respuestas en ciertos ejercicios específicos.

Este trabajo resultó fundamental para mi formación como Ingeniero Civil en Informática, ya que implicó el uso de tecnologías con las que inicialmente no estaba familiarizado, como *React*, *Next.js* y *TypeScript*. Esto requirió estudiar la documentación oficial y realizar pruebas adicionales, lo cual incrementó el tiempo dedicado al manejo de errores debido al desconocimiento inicial de estas herramientas. Asimismo, fue necesario trabajar con código externo perteneciente a la plataforma, lo que demandó comprender el funcionamiento de sus componentes y las funcionalidades específicas, como la implementación de componentes que se utilizó como base del desarrollo (*lvltutor*), el registro de acciones, *contentSelect*, y el componente encargado del uso y despliegue de pistas (*Hints*). Para gestionar las distintas fases del desarrollo, se utilizó un sistema de control de versiones mediante *Git* y *GitHub*, aprovechando el manejo de ramas. Además, se adoptaron buenas prácticas de desarrollo, incluyendo el uso de variables autodescriptivas, *commits* que documentaban los cambios realizados, y la documentación detallada de la lógica de los componentes. Finalmente, se llevó a cabo una fase de integración a producción, para garantizar que los cambios desarrollados de forma local se implementaran correctamente en la plataforma.

10. REFERENCIAS

Alkhatlan, A. & Kalita, J. (2019). Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments. *International Journal of Computer Applications* 181(43):1-20, March 2019.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences*, 4(2), 167–207.

Bourdeau, J., & Grandbastien, M. (2010). Modeling tutoring knowledge. *Studies in Computational Intelligence* (Vol. 308, pp. 123–143).

Facultad de Ciencias de la Ingeniería (2023) Libro guía Unidad Lógica y conjuntos.

Galafassi, C., Galafassi, F.F.P., Vicari, R.M. & Reategui, E.B. (2023) EvoLogic: Toward an ITS for Teaching Propositional Logic. *Int J Artif Intell Educ* 33, 35–58

Giraffa, L., Nunes, M. A, & Vicari, R.M. (1997) MultiEcological: A Learning Environment using multi-Agent architecture. *MASTA'97: Multi-Agent System: Theory and Applications. Proceedings. Coimbra: DE-Univ. de Coimbra.*

Kenneth R., Corbett A. (2006) Cognitive Tutors: Technology Bringing Learning Science to the Classroom, 135-168.

Nwana, H.S. (1990) Intelligent tutoring systems: an overview. *Artif Intell Rev* 4, 251–277 (1990).

Ritter, S., Anderson, J.R., Koedinger, K.R. & Corbett, A. (2007) Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review* 14, 249–255

Sáez, P. (2019). Arquitectura de microservicios para el Modelamiento del Aprendiz. Trabajo para optar al título de Ingeniero Civil en Informática

Uribe, M. (2023,). Más de 400 estudiantes ingresaron a primer año en carreras de la Facultad de Ciencias de la Ingeniería. *Diario UACH*, Retrieved 10 03, 2023, from <https://diario.uach.cl/mas-de-400-estudiantes-ingresaron-a-primer-ano-en-carreras-de-la-facultad-de-ciencias-de-la-ingenieria/>.

VanLehn, K. (1988). Student Modelling. M. Polson. *Foundations of Intelligent Tutoring Systems*. Hillsdale. N.J. Lawrence Erlbaum Associates, 55-78.

Brooke, J. (1996). SUS: a quick and dirty usability scale. 1996. *Usability Eval Ind*, 189(194), 4-7.

Maniktala, M., Chi, M., & Barnes, T. (2023). Enhancing a student productivity model for adaptive problem-solving assistance.

Hsu, S.-C. (2006). The design of a tutoring system with learner-initiating instruction strategy for digital logic problems.

Lesta, L., & Yacef, K. (2002). An intelligent teaching assistant system for logic. In *Proceedings of the 6th International Conference on Intelligent Tutoring Systems (ITS '02)* (pp. 421–431).

Barnes, T. & Stamper, J. (2010). Automatic Hint Generation for Logic Proof Tutoring Using Historical Data. *Journal of Educational Technology & Society*, 13(1), 3-12

Sieg, W. (2007). The AProS project: Strategic thinking & computational logic. *Logic Journal of the IGPL*, 15(4), 359–368.