



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería

Escuela de Ingeniería Civil en Informática

Tutor cognitivo de álgebra en contenidos remediales de fracciones y potencias

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:

Julio Daniel Guerra Hollstein

Doctor en Ciencias de la Información

Ingeniero Civil en Informática

PROFESOR INFORMANTE

Erick Alexis Araya Araya

Magister en Ingeniería Electrónica

Ingeniero Electrónico

PROFESOR INFORMANTE

Juan Pablo Concha Rosales

Magíster en Ciencias de la Ingeniería con
especialización en Modelamiento Matemático

Ingeniero Matemático

FELIPE ANDRES ROSSO DEL BARRIO

VALDIVIA – CHILE

2024

ÍNDICE

ÍNDICE	2
ÍNDICE DE TABLAS	5
ÍNDICE DE FIGURAS	6
1. INTRODUCCIÓN	10
1.1 Antecedentes del proyecto	11
1.2 Definición del problema.....	11
1.3 Propuesta de solución	12
1.4 Objetivos generales y específicos.....	12
1.5 Objetivo General.....	12
1.6 Objetivos específicos	12
2. METODOLOGÍA Y HERRAMIENTAS	13
2.1 Listado de metodologías por objetivo específico	13
2.1.1 Realizar un análisis del estado del arte de sistemas de tutores para contenidos de nivelación	13
2.1.2 Diseñar ejercicios y su interacción siguiendo lineamientos de UX para los contenidos de fracciones y potencias	13
2.1.3 Desarrollar la interfaz de usuario de un tutor cognitivo siguiendo un proceso centrado en el usuario	13
2.1.4 Evaluar la usabilidad del tutor	13
2.1.5 Desplegar la interfaz de usuario desarrollada en plataforma Learner model GQL	14
2.2 Arquitectura de la plataforma Learner Model GQL	14
2.2.1 Tecnologías de la plataforma	14
2.2.2 Detalles técnicos de la plataforma	15
2.3 Herramientas utilizadas.....	15
3. MARCO TEÓRICO Y ESTADO DEL ARTE	17
3.1 Tutores inteligentes.....	17
3.2 Tutores inteligentes para matemáticas	20
3.3 Desarrollos actuales	21
3.4 Resumen	21
4. DISEÑO Y SELECCIÓN DE CONTENIDOS	23
4.1 Reuniones de trabajo con expertos	23
4.2 Material de trabajo inicial	23
4.3 Estructuración del contenido	24
4.4 Análisis cognitivo por tipo de ejercicio	24
4.4.1 Ejercicios de Fracción algebraica	26
4.4.1.1 Reduccion de expresión	26
4.4.1.2 Fracciones con MCM/MCD	28
4.4.2 Ejercicios de Potencias.....	29
4.4.2.1 Evaluacion de expresión numérica 1	29
4.4.2.2 Evaluacion de expresión numérica 2	32
4.4.2.3 Reducción de expresión algebraica.....	33
4.4.2.4 racionalización de expresión	35
4.5 Aclaraciones y supuestos	36

5. DOCUMENTO DE REQUERIMIENTOS DE PRODUCTO	37
5.1 Introducción.....	37
5.1. Propósitos del producto.....	37
5.2 Los principios de LvlTutor.....	37
5.4 Perfil de usuario final y escenario	37
5.5 Problemas encontrados	37
5.6 Solución a los problemas detectados	38
5.7 Declaración de producto	38
5.8 Riesgos	38
5.9 Objetivos del producto	38
5.10 Funciones de LvlTutor	38
5.11 Arquitectura del sistema.....	39
5.12 Criterio de publicación	39
5.13 Requerimientos.....	39
5.13.1 Requisitos Funcionales	39
5.13.2 Requisitos No funcionales.....	39
5.13.3 Requisitos adicionales nacidos durante el proyecto	39
5.14 Casos de uso	39
5.14.1 Resolución de un tipo de ejercicio.....	39
5.14.2 Ingreso de expresiones matemáticas.....	40
5.14.3 Validación de respuestas ingresadas.....	40
5.14.4 Solicitud de mensaje de ayuda	40
5.14.5 Reporte de acciones	40
5.15.6 Diagrama de casos de uso	41
5.15 Estimación	41
5.16 Gestión de configuraciones de cambios y versiones	42
5.17 Crecimiento de los datos del producto.....	42
5.18 Bosquejos	42
6. DESARROLLO DE LA SOLUCIÓN	43
6.1 Conceptos tecnológicos relevantes para el desarrollo	43
6.2 Digitalización de contenido.....	43
6.3 Metodología de desarrollo de código.....	44
6.4 Resolución de un tipo de ejercicio.....	44
6.5 Ingreso de expresiones matemáticas.....	46
6.6 Validación de respuestas ingresadas.....	46
6.6.1 Traductor	47
6.6.2 Calculadora.....	52
6.6.3 Comparación estricta	52
6.7 Solicitud de mensaje de ayuda	52
6.8 Reporte de acciones	53
6.8 Diagramas estructurales y de comportamiento.....	53
7. VALIDACIÓN DE INTERFAZ.....	55
7.1 Evaluación experimental.....	55
7.3 Definición de hipótesis.....	55
7.4 Definición de variables	55
7.5 Diseño experimental	55
7.5.1 Encuesta SUS	57

7.5.2 Encuesta de satisfacción.....	57
7.5.3 Guión del laboratorio	58
7.6 Implementación de aplicación para el laboratorio	59
7.7 Ejecución del laboratorio	59
7.9 Resultados	60
7.9.1 Datos de la primera fase	60
7.9.2 Resultados segunda fase.....	68
7.9.3 Código de preparación de los datos	68
7.10 Análisis de resultados.....	69
8. INTEGRACIÓN Y DESPLIEGUE	73
8.1 Github.....	73
8.2 Despliegue de los ejercicios	73
8.3 Reusabilidad y extensibilidad del componente de evaluación numérica.....	76
8.3.1 Integración en otros componentes	76
8.3.2 Incorporación de nuevas funciones u operadores al traductor y calculadora...	77
9. CONCLUSIONES	78
10. REFERENCIAS.....	80

ÍNDICE DE TABLAS

Tabla 1, Listado de servicios de la plataforma.....	14
Tabla 2, Listado de tecnologías con su descripción.....	15
Tabla 3, Listado de herramientas utilizadas.	16
Tabla 4, estructura de ejercicios.	24
Tabla 5, Ocho principios de la teoría ACT-R	25
Tabla 6, Estimación por puntos de uso	41
Tabla 7, Estructura donde se almacena un ejercicio.....	43
Tabla 8, Estructura donde se almacena un paso.	44
Tabla 9, Expresión transformada con Mathquill.	46
Tabla 10, Ejemplo de ejecución del código	51
Tabla 11, Expresiones definidas	56
Tabla 12, Orden de expresiones por grupo de cuadrado latino.	57
Tabla 13, Definiciones de etiquetado.....	60
Tabla 14, Cantidad de envío de respuestas del participante (q) por tipo de entrada y usuario.	61
Tabla 15, resultados de encuesta sus	68
Tabla 16, resultados de encuesta de satisfacción.....	68
Tabla 17, retroalimentación del usuario.....	68
Tabla 18, resumen del tiempo (en segundos) que un usuario tardó en pasar a la siguiente expresión	71

ÍNDICE DE FIGURAS

Figura 1, Arquitectura de microservicios.....	14
Figura 2, arquitectura de un ITS.....	18
Figura 3, representación de descomposición algorítmica.	20
Figura 4, Diseño de contenido con la herramienta ITSB.....	21
Figura 5, Ejercicio seleccionado.....	23
Figura 6, Evidencia de diferencia en cantidad de pasos.	24
Figura 7, expresión a reducir.....	26
Figura 8, Solución esperada.	26
Figura 9, Soluciones esperadas.....	27
Figura 10, Solución esperada.	27
Figura 11, Solución esperada.	28
Figura 12, expresión a sumar.....	28
Figura 13, Contexto de resolución.....	28
Figura 14, Solución esperada.	29
Figura 15, Solución esperada.	29
Figura 16, expresión a evaluar.....	30
Figura 17, Solución esperada.	30
Figura 18, Solución esperada.	30
Figura 19, Solución esperada.	31
Figura 20, Solución esperada.	31
Figura 21, Solución esperada.	31
Figura 22, expresión a evaluar.....	32
Figura 23, Solución esperada.	32
Figura 24, Solución esperada.	32
Figura 25, Solución esperada.	33
Figura 26, expresión a reducir.....	33
Figura 27, Propiedad de potencias en matriz cuadrada.....	34
Figura 28, Propiedad de potencias de un producto.....	34
Figura 29, Solución esperada.	34
Figura 30, Expresión a racionalizar.	35
Figura 31, Propiedad de fracciones.....	36
Figura 32, Solución esperada.	36
Figura 33, Diagrama de casos de usos.	41
Figura 34, bosquejo de aplicación móvil	42
Figura 35, Notificación de un nuevo hint al cometer un error.	42
Figura 36, Ejemplo de resolución de un ejercicio.	45
Figura 37, Despliegue del resumen al terminar el ejercicio.....	45
Figura 38, Código de interfaces con variables de interés.....	45
Figura 39, Traducción automática de ASCII a lenguaje matematico.	46
Figura 40, Barra de herramientas para ingreso de operaciones.....	46
Figura 41, algoritmo <i>shunting yard</i>	48
Figura 42, Preparación del algoritmo.....	48
Figura 43, Construcción de una palabra reservada.....	49
Figura 44, Construcción de una palabra numérica.	49
Figura 45, Manejo de operadores.	50

Figura 46, Continuación del manejo de operadores.	51
Figura 47, Manejo de operadores rezagados.	51
Figura 48, Despliegue de una ayuda.	53
Figura 49, Diagrama de componentes de LvlTutor.	53
Figura 50, Diagrama de secuencia para el cargado de un ejercicio.	54
Figura 51, Diagrama de secuencia para la resolución de un ejercicio.	54
Figura 52, Bosquejo de ingreso de expresiones.	59
Figura 53, Ingreso con ASCII para el Usuario D.	62
Figura 54, Ingreso con MQ2 para el Usuario D.	63
Figura 55, Ingreso con ASCII para el Usuario A.	64
Figura 56, Ingreso con MQ2 para el Usuario A.	65
Figura 57, Ingreso con ASCII para el Usuario C.	66
Figura 58, Ingreso con MQ2 para el Usuario C.	67
Figura 59, Código python trabajado en la herramienta Jupyter.	69
Figura 60, Error de cálculo en punto flotante.	70
Figura 61, Error en la validación de raíz en ASCII.	70
Figura 62, Ingreso de un participante al intentar probar el funcionamiento.	70
Figura 63, histograma para la diferencia de cantidad de acciones utilizadas (q) entre los ingresos MQ2 y ASCII. Valores negativos reflejan menos acciones con MQ2.	71
Figura 64, Conector entregado para el producto “LvlTutor”.	73
Figura 65, despliegue del desarrollo de un paso con LvlTutor.	74
Figura 66, despliegue del completado de un ejercicio con LvlTutor.	75
Figura 67, Ejemplo de uso del traductor y evaluación numérica.	76
Figura 68, Código del ejemplo de uso para uso del traductor y evaluación.	76
Figura 69, Listado de funciones y operadores implementados.	77
Figura 70, constantes para definir precedencia y asociatividad.	77

RESUMEN

Este trabajo describe el desarrollo de la solución Lvlutor para el despliegue de ejercicios interactivos en el tutor inteligente de la plataforma Learner Model GQL. Lvlutor es un interfaz flexible que permite el despliegue de contenidos. El diseño de interfaz fue inspirado en la revisión del estado del arte en tutores de matemática y la creación de contenidos fue realizada mediante un trabajo en conjunto a profesores de matemática, haciendo un modelamiento de ejercicios para contenidos de nivelación de álgebra (fracciones algebraicas y potencias). Con el fin de adquirir los conocimientos de las diferentes tecnologías, Lvlutor fue desarrollado en un proyecto independiente a la plataforma Learner Model GQL, centrándose en una validación de respuestas realizada por evaluación matemática, una entrada amigable para el ingreso de expresiones matemáticas y una flexibilidad para desplegar contenidos con una cantidad de pasos variable. Las primeras evaluaciones de Lvlutor muestran resultados prometedores en comparación con el ingreso de texto tradicional (ASCII) al no aumentar el tiempo necesario para ingresar una expresión y disminuir los errores cometidos por el estudiante en el contexto de la resolución de problemas. Una puntuación media de la encuesta SUS de 53,75 (resultados deficientes) muestra que la solución tiene margen de mejora, especialmente en el departamento de comunicación, debido a la retroalimentación proporcionada por los estudiantes que muestra las instrucciones y ayudas entregadas durante el contexto de resolución del problema no fueron lo suficientemente claras. Finalmente, luego de hacer correcciones de incidencias encontradas en la validación se logró integrar Lvlutor a la plataforma Learner Model GQL mediante la creación de una nueva rama en la solución para el control de versiones entregada por Github.

ABSTRACT

This work describes the development of the Lvlutor solution for the deployment of interactive exercises in the intelligent tutor of the Learner Model GQL platform. Lvlutor is a flexible interface that allows the deployment of content. The interface design was inspired by the review of the state of the art in mathematics tutors and the creation of content was carried out through joint work with mathematics teachers, modeling exercises for algebra leveling content (algebraic fractions and powers). In order to acquire knowledge of different technologies, Lvlutor was developed in a project independent of the Learner Model GQL platform, focusing on a validation of answers carried out by mathematical evaluation, a friendly input for the entry of mathematical expressions and a flexibility to deploy content with a variable number of steps. Early evaluations of Lvlutor show promising results compared to traditional text entry (ASCII) by not increasing the time needed to enter an expression and decreasing errors made by the student in the context of problem solving. An average SUS survey score of 53.75 (poor results) shows that the solution has room for improvement, especially in the communication department, due to the feedback provided by students showing that the instructions and aids delivered during the context of resolution of the problem were not clear enough. Finally, after making corrections to incidents found in the validation, Lvlutor was integrated into the Learner Model GQL platform by creating a new branch in the version control solution delivered by Github.

1. INTRODUCCIÓN

La Facultad de Ciencias de Ingeniería (FCI) de la Universidad Austral de Chile (UACH) logra evidenciar en los últimos 5 años del curso álgebra I de primer año, un índice elevado de reprobación que fluctúa entre un 45% y un 61%. Se estima que las causales son de factores educacionales y socioeconómicos previos al ingreso universitario.

Entre las múltiples estrategias aplicadas por los profesionales de la educación en la UACH, existen programas de nivelación para diferentes cursos, álgebra entre ellos. El objetivo de estos programas de nivelación es poder suplir las herramientas e información necesaria que permitan al estudiante generar el conocimiento suficiente para enfrentar los cursos de primer año.

Debido a una diferencia en las habilidades y saber de cada estudiante se dificulta generar un proceso general que permita suplir las necesidades de todos. Además, la nivelación estudiantil se realiza en corto tiempo, lo que dificulta poder identificar el tiempo necesario para cada estudiante en el cual se pueda practicar las herramientas e información que permitan generar un conocimiento suficiente en los estudiantes para los contenidos educacionales enfrentados.

Con el fin de identificar y enfrentar de mejor manera los diferentes factores que dificultan la nivelación estudiantil en este contexto, el proyecto Fondecyt 11220709 propone incorporar tutores cognitivos e investigar factores motivacionales. Los tutores cognitivos (CT) pertenecen al grupo de sistemas de tutoría inteligente (ITS) que incorporan un modelo cognitivo y proveen instrucción personalizada. Los CT se han refinado durante las últimas cinco décadas haciendo uso de los principios de las teorías de control adaptativo del pensamiento (ACT), ACT* y ahora conocida como ACT-R (JR. Anderson, et al, 1996).

Un CT busca potenciar la adquisición de habilidades y generación del conocimiento en el estudiante mediante una interfaz interactiva de resolución de problemas que logre entregar ayuda y retroalimentación mediante diferentes intervenciones. Un CT monitorea el crecimiento del conocimiento del estudiante y debe proveer diferentes oportunidades de aprendizaje para que el estudiante adquiriera maestría del conocimiento.

Numerosas investigaciones logran mostrar resultados positivos tanto en la adquisición de habilidades del estudiante en álgebra, efectividad en pruebas estandarizadas o en la percepción de uso por estudiantes después de hacer uso de tutores cognitivos (ver, por ejemplo, Corbett y Trask, 2000; Ritter et al., 2007; Pane y Griffin 2014; AbuEloun y Naser, 2017). Esta evidencia motiva a los investigadores del proyecto en la implementación de un tutor de álgebra que cubra los contenidos de nivelación necesarios para el curso de Álgebra I en el contexto de la FCI de la UACH.

1.1 Antecedentes del proyecto

Con el fin de comenzar el desarrollo de tutores cognitivos adecuados a la realidad educacional presente a los alumnos de primer año en la UACH, se han realizado o están activos los siguientes proyectos:

- I. Julio Daniel Guerra Hollstein, a raíz del proyecto fondecyt 11220709 busca investigar los efectos de los tutores cognitivos y las características motivacionales en el perfil motivacional del aprendiz, y "engagement" con el sistema y resultados de aprendizaje, entre otras hipótesis es que, propone construir tutores inteligentes para contenidos de habilitación de álgebra los cuales están cubiertos por programas de nivelación en la Universidad Austral de Chile (Guerra, 2022).
- II. Pablo Sáez Parra, desarrolló la plataforma Learner Model GQL. La plataforma busca fomentar la creación y experimentación de un ambiente de desarrollo innovador que permite una gran flexibilidad y control sobre todos los procesos del sistema, dando una base completa para el avance de proyectos de nuevos sistemas educativos inteligentes de la universidad (Sáez, 2022).
- III. Miguel Nahuelpán, desarrolló un tutor inteligente para el contenido de factorización de las actividades de nivelación del curso de álgebra. El tutor fue desarrollado con la metodología scrum, realizando prototipos mediante mockups para determinar los elementos de la UI y probado en términos de usabilidad obteniendo buenos resultados, pero requiriendo un piloto con mayor audiencia (Nahuelpán, 2021).
- IV. Mathias Oyarzun Altamirano, desarrolló y evaluó una interfaz inteligente para el contenido de ecuaciones del curso de nivelación satisfaciendo las definiciones de habilidad necesarias y contenido a resolver (Oyarzun, 2023).

1.2 Definición del problema

En la plataforma Learner Model GQL se logran definir los modelos de dominio, contenido, pedagógico y del aprendiz sobre los cuales se trabajarán en la construcción del ITS utilizado en el proyecto fondecyt 11220709 liderado por el Dr. Julio Daniel Guerra Hollstein. La existencia de la plataforma y las definiciones de los modelos trabajados significa que el siguiente trabajo corresponde a crear el material y código utilizado por los diferentes modelos.

Previamente en los antecedentes del proyecto se menciona el desarrollo del tutor inteligente para el contenido de factorización por Miguel Nahuelpán. En el trabajo de Miguel se logra desarrollar la interfaz para diferentes tipos de ejercicios en el tópico de factorización. Sin embargo, las interfaces definidas son implementadas ad hoc para los ejercicios seleccionados. Esto implica que realizar ajustes a los diferentes tipos de ejercicio de otros tópicos requiera un trabajo considerable.

Debido a que las implementaciones previas solo cubren los contenidos de factorización y ecuaciones de una manera estricta, nace la necesidad de una implementación más flexible que facilite la creación de nuevo contenido para satisfacer tanto los modelos de dominio y contenido en los diversos tópicos de nivelación tales como los de fracciones y potencias.

1.3 Propuesta de solución

Para poder satisfacer las necesidades presentadas se propone la creación de una interfaz dinámica para los contenidos de fracciones y potencias que además sea capaz de ajustarse a los diferentes contenidos sin la necesidad de alterar código. Esto facilita poder crear nuevo contenido simplemente modificando la información almacenada en una estructura de datos definida en proyectos anteriores.

1.4 Objetivos generales y específicos

A continuación, se presentan los objetivos generales y específicos de este proyecto, los cuales tendrán una mayor explicación en el capítulo dos, donde se expondrán las metodologías y herramientas.

1.5 Objetivo General

Desarrollar el contenido y la interfaz de usuario de un tutor cognitivo para los temas remediales de fracciones y potencias que apoye a estudiantes de ingeniería de primer año.

1.6 Objetivos específicos

- I. Realizar un análisis del estado del arte de sistemas de tutores para contenidos de nivelación.
- II. Diseñar ejercicios y su interacción siguiendo lineamientos de UX para los contenidos de fracciones y potencias.
- III. Desarrollar la interfaz de usuario de un tutor cognitivo siguiendo un proceso centrado en el usuario.
- IV. Evaluar la usabilidad de la interfaz de usuario desarrollada.
- V. Desplegar la interfaz de usuario desarrollada en la plataforma Learner Model GQL.

2. METODOLOGÍA Y HERRAMIENTAS

En este capítulo se presentan las metodologías utilizadas para el desarrollo de los objetivos definidos en el capítulo de Introducción, y luego se introduce la arquitectura y tecnologías relacionadas al proyecto.

2.1 Listado de metodologías por objetivo específico

2.1.1 Realizar un análisis del estado del arte de sistemas de tutores para contenidos de nivelación

Se realizó un estudio separado en varias fases durante el curso del proyecto de la literatura recomendada por el patrocinante complementada con literatura de Albert Corbert en Tutores cognitivos. Este estudio en primera instancia fue exploratorio para comprender los conceptos y diferentes acercamientos en la construcción de sistemas de tutoría inteligentes, luego se profundizó en los trabajos donde se construyen tutores cognitivos. Como producto de este estudio literario se logra generar una mejor percepción en la construcción de sistemas de tutoría inteligentes y tutores cognitivos, generando así material para la redacción del estado del arte.

2.1.2 Diseñar ejercicios y su interacción siguiendo lineamientos de UX para los contenidos de fracciones y potencias

Haciendo uso del marco teórico obtenido en la revisión literaria se desarrollará contenido de fracciones haciendo uso del material del libro de nivelación, para luego ser presentado y trabajado en múltiples sesiones con profesores de matemática de la FCI UACH. Los resultados obtenidos permitirán realizar un mejor diseño de los tipos de ejercicios y de requerimientos de producto del proyecto (PRD).

2.1.3 Desarrollar la interfaz de usuario de un tutor cognitivo siguiendo un proceso centrado en el usuario

Se realizó uso de la metodología incremental para implementar los casos de uso definidos en la documentación presentada en el PRD en un nuevo proyecto con la tecnología NextJS. Como resultados se obtiene experiencia con las tecnologías involucradas en la plataforma y el desarrollo del código de producto.

2.1.4 Evaluar la usabilidad del tutor

Teniendo la implementación de la solución se procede a definir y ejecutar un laboratorio de evaluación de producto donde se busca validar diferentes elementos de ingreso de expresión en interfaz siguiendo la estrategia de estudio de cuadrado latino con el fin de controlar variables de dificultad y orden de presentación. Luego de evaluar los ingresos de expresiones se procederá a evaluar la ejecución de los tipos de ejercicios implementados. Como resultados de este laboratorio se obtendrá las respuestas de encuestas de usuario y usabilidad, que permitirán detectar nuevos requerimientos que deben ser resueltos antes de desplegar el producto a la plataforma Learner Model GQL.

2.1.5 Desplegar la interfaz de usuario desarrollada en plataforma Learner model GQL

Antes de integrar el producto con la plataforma LMGQL se resuelven requerimientos detectados a partir del laboratorio de evaluación. Luego de realizar la integración con la plataforma principal el equipo de trabajo realiza el despliegue en Vercel. Producto al intento de despliegue en vercel se corrige el código de javascript a typescript y se implementan nuevos requerimientos presentados durante el proyecto.

2.2 Arquitectura de la plataforma Learner Model GQL

La plataforma fue diseñada bajo la arquitectura de microservicios mostrada en la Figura 1. Los microservicios son pequeños, independientes y burdamente acoplados. Un servicio se debe encargar de su propia persistencia de información. La comunicación de los servicios se puede realizar mediante una puerta de enlace de interfaz de programación de aplicación bien definida. Como ventaja/desventaja, cada servicio puede tener su propio ecosistema. La Tabla 1 muestra el listado de servicios de la plataforma.

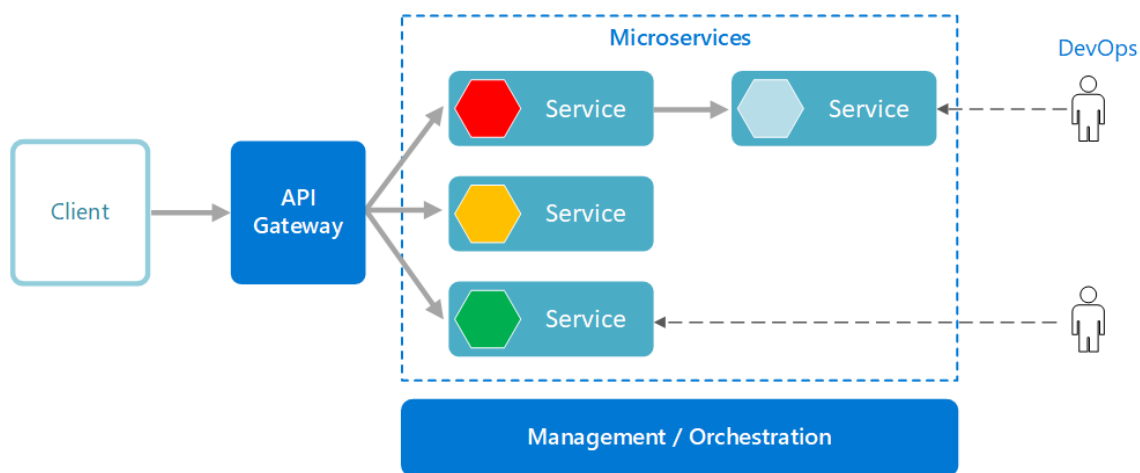


Figura 1, Arquitectura de microservicios.¹

Tabla 1, Listado de servicios de la plataforma

Nombre	Actions	Content	Domain	Projects	State	Users
Función	Administración de datos monitoreados	Administración de contenido	Administración de dominio tópico y KC	Administración de proyectos	Administración del modelo de aprendizaje	Administración de usuarios

2.2.1 Tecnologías de la plataforma

El listado de tecnologías utilizadas por la plantilla de cliente entregada por plataforma Learner Model GQL puede ser visto en la Tabla 2.

¹ Sáez, P. (10 de octubre de 2023). *Architecture - Learner Model GQL*. Learner Model GQL. <https://docs.lm-uach.org/how-it-works/architecture>.

Tabla 2, Listado de tecnologías con su descripción.

Nombre	Descripción
Next.js	Entorno de trabajo flexible que entrega bloques de construcción para la creación rápida de aplicaciones web (Vercel Inc., 2023)
TypeScript	Superconjunto de JavaScript con seguridad de tipo de datos en mente. TypeScript es JavaScript pero con sintaxis para tipos (Microsoft, 2023).
Auth0	Plataforma adaptable de autenticación y autorización, simple de implementar (Okta Inc., 2023)
Chakra UI	Librería de componentes simple, modular y accesible que entrega los bloques de construcción necesarios para construir aplicaciones React (Adebayo S., 2023)
GraphQL Code Generator	Herramienta plugin de generación de código para pilas que utilizan GraphQL (The Guild, 2023)
React Query	Facilita la obtención, el caché, el sincronizado y actualizado de estados del servidor en react. (Linsley T, 2023)
Valtio	Interfaz de programación de aplicación (API) que facilita el manejo de estados en react mediante una integración simplificada del patrón de diseño proxy (Valtio, 2023)

2.2.2 Detalles técnicos de la plataforma

Como el desarrollo de tutores se concentra en el cliente, la plataforma entrega una plantilla de desarrollo. La plantilla logra simplificar el desarrollo de tutores eliminando la carga que lleva comunicarse con la puerta de enlace o administración de los diferentes servicios mediante consultas predefinidas realizadas en GraphQL. Debido a esta abstracción del proceso de desarrollo, la creación de nuevos tutores se puede concentrar en los aspectos de interfaz. Para mayor detalle de la plataforma, referirse a la documentación encontrada en el sitio: <https://docs.lm-uach.org/>.

2.3 Herramientas utilizadas

El listado de herramientas utilizadas en el transcurso del proyecto de titulación puede ser observado en la Tabla 3.

Tabla 3, Listado de herramientas utilizadas.

Nombre	Función y uso principal
Zoom	Plataforma de comunicación y colaboración, principalmente utilizada para reuniones de trabajo no presenciales durante periodos de cuarentena.
Visual studio	Entorno de desarrollo integrado(IDE), se utilizó para el desarrollo del proyecto.
Notepad + +	Editor de código, se utilizó para comparación y análisis de código.
NextJS	Entorno de trabajo, se utilizó para el desarrollo, depuración y pruebas del proyecto.
Git	Sistema esencial utilizado para el control de versiones.
Github	Servicio de control de versiones en la nube, se utilizó para colaboración centralizada en el desarrollo del proyecto.
Vercel	Servicio el despliegue y alojamiento de aplicaciones utilizado mediante la integración otorgada por github.

3. MARCO TEÓRICO Y ESTADO DEL ARTE

3.1 Tutores inteligentes

Alrededor del año 1970 se desarrollan los tutores de **instrucción asistida por computador (CAI)** mostrando resultados positivos (Alkhatlan y Kalita, 2018). Los tutores CAI reciben una respuesta final a un problema y entregan una retroalimentación básica como felicitaciones, pistas o "intentar nuevamente" (VanLehn, 2011). Finalmente, las investigaciones concluyen que los tutores CAI pueden ser mejorados con la incorporación de técnicas de inteligencia artificial (AI) (Alkhatlan y Kalita, 2018).

Los tutores de **instrucción asistida por computador inteligentes (ICAI)** o bien conocidos como **sistemas de tutoría inteligentes (ITS)** nacen como producto de los resultados informáticos y educacionales obtenidos en los tutores CAI, los resultados obtenidos por Bloom en el año 1984 y los avances tecnológicos de la AI (Alkhatlan y Kalita, 2018). Bloom logra mostrar que los estudiantes bajo la tutela individual, enfocada en el dominio de habilidades y realizada por un **tutor humano (HT)**, presentan tamaño de efecto $d=2$ para la comparación de desempeño entre estudiantes con y sin tutela individual, o bien del percentil 50 al percentil 98 (Alkhatlan y Kalita, 2018; VanLehn, 2011; Kulik y Fletcher, 2016). Esto motiva a los investigadores en AI y CAI explorar diferentes abstracciones de la tutoría realizada por un humano con el fin de predecir el dominio de las habilidades obtenidas por el estudiante en base a sus acciones y retroalimentación del sistema (Alkhatlan y Kalita, 2018).

El tamaño del efecto de los diferentes sistemas es analizado por diferentes metaanálisis comparando los resultados obtenidos a partir del despliegue de los tutores con los resultados de no utilizar tutores (Kulik y Fletcher, 2016). Los tamaños de efecto medios obtenidos por las implementaciones exitosas de las diversas tutorías son: HT con $d=0.8$ o bien del percentil 50 al percentil 79, CAI con $d=0.31$ o bien del percentil 50 al percentil 62, ITS con $d=0.66$ o bien del percentil 50 al percentil 75.

Los ITS con mayor éxito se pueden clasificar según su **granularidad** tales como los tutores **basados en respuesta** que reciben la respuesta final, **basados en pasos** que deconstruyen el procedimiento de resolución de manera fija y **basados en sub pasos** con deconstrucción variable ajustada al estudiante individual (Alkhatlan y Kalita, 2018; VanLehn, 2011). Esta granularidad busca emular el **escalonamiento** del conocimiento que puede aplicar un tutor humano al profundizar el conocimiento obtenido por un estudiante (VanLehn, 2011).

El metaanálisis presentado por VanLehn, 2011, compara los tipos de tutores según su granularidad mostrando mayor complejidad. Principalmente muestra los resultados mediante la hipótesis de la meseta de interacción donde obtiene: *tutor humano = tutor basado pasos = tutor basado en sub pasos > tutor basado en respuesta > sin tutor*.

Los **tutores cognitivos (CT)** son un ITS y se enfocan en la construcción de un currículo basado en investigación siguiendo la teoría cognitiva del **control adaptativo del pensamiento (ACT, ACT-R)** como base teórica, logrando así incorporar la psicología como componente adicional. La teoría ACT-R mantiene los tres principios de la teoría

original ACT los cuales son: La distinción procedural-declarativa, compilación del conocimiento y fortalecimiento. En otras palabras, el conocimiento necesario para realizar una tarea u objetivo se **fortalece/desfortalece** según la frecuencia de la práctica en diferentes oportunidades de los **componentes del conocimiento (KC)** que representan el conjunto de los **conocimientos declarativos y procedurales**. El **conocimiento declarativo** incluye hechos, imágenes y sonidos, mientras que el **conocimiento procedural** consiste en el entendimiento de cómo hacer las cosas dentro de un contexto (John R. Anderson et al., 1996; Ritter et al., 2007).

La Figura 2 logra resumir de manera genérica la arquitectura de un ITS, los módulos son representados por diferentes modelos con tareas específicas (Alkhatlan y Kalita, 2018). Como este trabajo se enfoca en la construcción de un CT, se procederá a definir los diferentes modelos en el contexto de un CT.

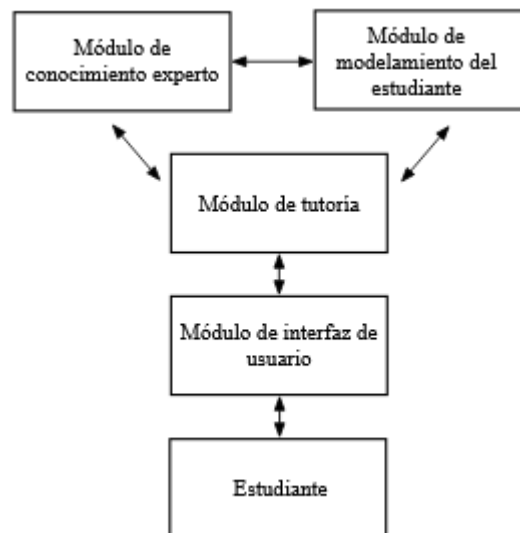


Figura 2, arquitectura de un ITS.²

El **modelo de dominio** o de conocimiento experto, representa los hechos, conceptos, estrategias de resolución y reglas de un dominio en particular a enseñar. Los CT utilizan el modelo cognitivo para entregar principios, estrategias y conocimiento declarativo mediante una retroalimentación precisa durante la resolución paso a paso de un ejercicio (Alkhatlan y Kalita, 2018).

El **modelo del estudiante** busca obtener la representación del dominio del estudiante sobre un KC, prediciendo el conocimiento obtenido con la práctica de este KC en diferentes instancias utilizando la técnica de AI nombrada **trazas de conocimiento**

² AbuEloun, N. N., & Naser, S. S. A. (2017). Mathematics intelligent tutoring system. International Journal of Advanced Scientific Research, 2, 11-16.

bayesiana (BKT) (John R. Anderson et al., 1996; Ritter et al., 2007; Desmarais y d. Baker, 2012).

El **modelo pedagógico** o tutoría, puede ser construido con la descomposición de tareas complejas en un conjunto de KC, enfatizando la práctica de los componentes individuales (Ritter et al., 2007).

El **modelo de interfaz** busca generar la transición del currículo construido al contexto en que el estudiante deberá adquirir y practicar las habilidades mediante la resolución de contenidos específicos. John R. Anderson et al., 1996, logra expandir la teoría ACT-R en ocho principios que entregan una guía de los aspectos a considerar al momento de diseñar una interfaz para un CT.

La interacción de los cuatro modelos también se puede resumir en **dos lazos** (Alkhatlan y Kalita, 2018). Primero se tiene el **lazo interno** que provee el contexto del contenido desplegado por la interfaz al estudiante, y la retroalimentación entregada mediante mensajes de ayuda, mensajes de error y reporte de acciones al sistema. Segundo se tiene el **lazo externo** que se encarga de predecir el conocimiento actual del estudiante y seleccionar la entrega de contenido a resolver en base a la maestría obtenida.

Para el desarrollo de los tutores cognitivos y así incorporar el uso de esta tecnología en la rutina diaria de enseñanza, se necesita la cooperación mutua con el personal de enseñanza con el fin de entender las necesidades locales del currículo (John R. Anderson et al., 1996; Ritter et al., 2007). Mientras que la validación de esta implementación puede realizarse tanto por medios externos como pruebas de diagnóstico o finales, y por medios internos como el análisis de la capacidad de predecir los conocimientos del estudiante dentro del mismo sistema (Desmarais y d. Baker, 2012).

El **modelado basado en restricciones (CBM)** presenta otra manera de construir los tutores. Los tutores basados en CBM representan el conocimiento declarativo en forma de restricciones sobre la respuesta del estudiante o el resultado de las acciones del estudiante. Los **tutores de secuenciado de contenido** presentan un acercamiento diferente al ajustar el contenido a presentar al estudiante luego de evaluar de manera precisa un gran arreglo de habilidades con la menor cantidad posible de evidencia (Desmarais y d. Baker, 2012).

AbuEloun y Naser, 2017, introducen un listado de artículos donde se implementan los tutores en diferente en diferentes áreas y grados de educación. La Figura 3 muestra el funcionamiento de la interfaz de un tutor de álgebra (a), mientras que (b) muestra la retroalimentación entregada al estudiante al cometer un error (John R. Anderson et al., 1996).

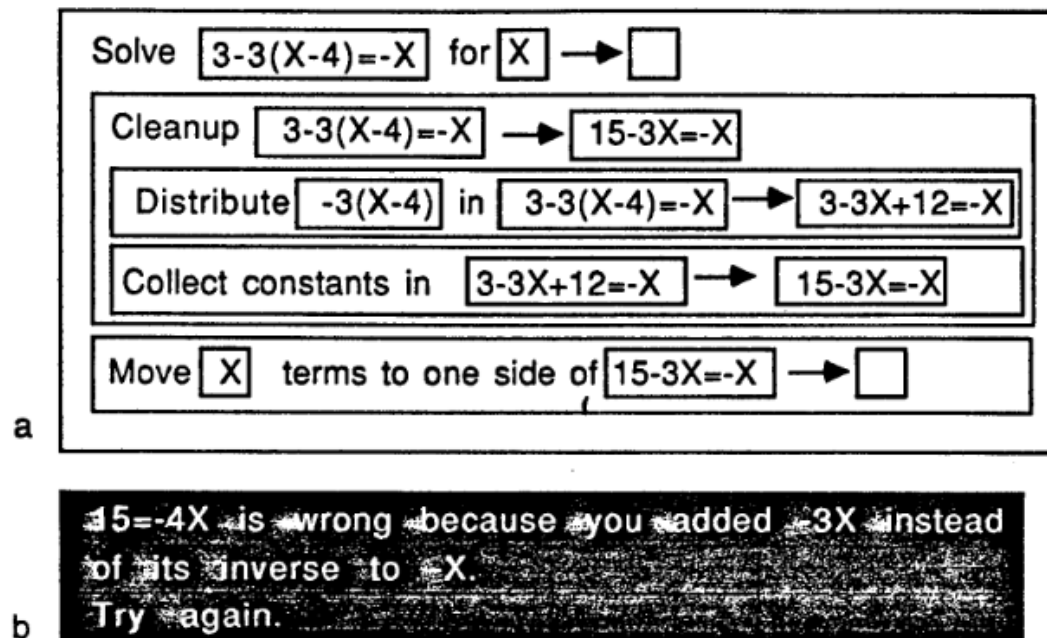


Figura 3, representación de descomposición algorítmica.³

3.2 Tutores inteligentes para matemáticas

A continuación se mostraran tres ITS que han sido utilizados para crear diferentes currículos entre ellos de álgebra:

El **Tutor Cognitivo Álgebra I (CETAI)** es un curso de álgebra de primer año diseñado para estudiantes con un grado de habilidades variable. El currículo incluye textos tradicionales, guías de trabajo y ITS basado en la teoría ACT-R (John F. Pane et al., 2014). CETAI ha sido estudiado en escala obteniendo efectos significativos para enseñanza media (High School) principalmente luego del segundo año de uso (John F. Pane et al., 2014). Otro estudio examina la efectividad del dominio de habilidades debido a que el tutor permite a los estudiantes pasar al siguiente ejercicio luego de múltiples fallos, mostrando mejor éxito en aquellos estudiantes que presentan dificultades (Sales y Pane, 2019).

ALEKS es un tutor de secuenciado de contenido basado en la teoría cognitiva de espacios de conocimiento. Este tipo de ITS diseña el camino de aprendizaje luego de obtener un diagnóstico a diferencia de un CT/CBM que buscan proveer contenido de manera remedial (Desmarais y d. Baker, 2012).

El **Constructor de Sistemas de Tutoría Inteligentes (ITSB)** es una herramienta de autoría para embalar un ITS. Esta herramienta busca simplificar la fabricación de contenido como se observa en la Figura 4. (Naser. 2016).

³ John R. Anderson, A. T. C., Koedinger, K., & Pelletier, R. (1996). Cognitive Tutors: Lessons Learned. United States Army Research Institute for the Behavioral and Social Sciences.

Questions and Answers Data Entry

Enter Question Text 1: Which is of the following is true about the above Java code ?

Enter Question Text 2: Choose all applicable:

Enter Answer Choice 1: Accessing a private variable

Enter Answer Choice 2: Accessing a private method

Enter Answer Choice 3: There is a semantic error

Enter Answer Choice 4: There is no problems in the code

Enter Answer Choice 5:

Enter Answer Choice 6:

Enter Correct Answers: Choice 1: 1 Choice 2: 0 Choice 3: 0 Choice 4: 1 Choice 5: 0 Choice 6: 0

Level of difficulty: 1

Buttons: [Navigation icons] [Close]

Figura 4, Diseño de contenido con la herramienta ITSB.⁴

3.3 Desarrollos actuales

Los desarrollos actuales en el campo de los ITS son: 1) Sistemas de tutoría afectiva (ATS) encargados en reconocer el efecto de las emociones humanas en diferentes maneras, 2) El reconocimiento cultural en la educación, 3) ITS basados en juegos, 4) ITS adaptativos en la nube, 5) Aprendizaje colaborativo, 6) Minería de datos en ITS y 7) Herramientas de autoría de ITS., (Alkhatlan y Kalita, 2018).

3.4 Resumen

Hay alrededor de cincuenta años de experiencia en el desarrollo e investigación de los tutores. Mientras que existen alrededor de veinte años de experiencia con el despliegue de los diversos tutores en el aula de clases.

Los ITS buscan obtener los mismos resultados que un HT en términos de tamaño de efecto al comparar los resultados de los estudiantes que usan un tutor, con los resultados de los estudiantes que no usan el tutor. Luego se logra comparar los tutores según su granularidad y escalonamiento.

Para construir un tutor existe una arquitectura que resume la interacción de los modelos de dominio, estudiante, contenido, pedagógico e interfaz. La interacción de los cinco modelos también se puede resumir según un lazo interno y un lazo externo.

⁴ Naser, S. S. A. (2016). ITSB: An Intelligent Tutoring System Authoring Tool. Journal of Scientific and Engineering Research.

Los CT son un ITS que implementan el modelo cognitivo ACT-R en los modelos de dominio, pedagógico, contenido e interfaz, y utilizan la técnica de AI BKT para el modelo del estudiante.

La evaluación de un tutor se puede realizar de manera externa comparando los resultados de diagnóstico antes de utilizar un tutor y los resultados finales después de utilizar un tutor. Otra forma de evaluación es analizar de manera interna la capacidad de predecir los conocimientos del estudiante.

Finalmente existen implementaciones de los ITS en diferentes áreas y grados de educación entre ellas tutores de álgebra para contenidos remediales.

4. DISEÑO Y SELECCIÓN DE CONTENIDOS

El desarrollo de un tutor cognitivo es una tarea multidisciplinaria donde es importante comprender el currículum que se diseñará mediante la selección de contenido. Esto permitirá entender las estrategias cognitivas a utilizar para que la interfaz sea capaz de brindar el nexo de comunicación entre el estudiante y la plataforma.

4.1 Reuniones de trabajo con expertos

Se realizaron un total de seis sesiones de trabajo que fueron coordinadas por el patrocinante del proyecto. Las reuniones se caracterizaron por ser semanales y de libre discusión, pero en aquellas que fue necesario se logra definir una pauta de trabajo con el fin de optimizar el tiempo de trabajo. La estructura de una reunión comprende tres integrantes: patrocinante, experto y tesista, con una duración entre 45 y 90 minutos. Los expertos que participaron son Javier Arriaza y Juan Concha, ambos expertos trabajan para el Centro de Docencia para Ciencias Básicas en Ingeniería de la Facultad de Ciencias de la Ingeniería en la UACH.

4.2 Material de trabajo inicial

En primera instancia para la selección de contenido se utiliza el libro de Matemática Básica para la Inserción a la Ingeniería (Vol. 6) por Álvarez et al. Este libro fue redactado para ser utilizado en el curso de nivelación de álgebra. A partir de este material se desarrolló un documento redactado en LaTeX con ejercicios de fracción algebraica resueltos con el fin de ser trabajado en la primera reunión de trabajo.

Como resultado de la primera sesión de trabajo selecciona la expresión mostrada en la Figura 5, además de identificar la necesidad de un ejercicio más simple donde se trabaje fracciones con mínimo común múltiplo.

$$\left(\frac{a}{b} + \frac{c}{d}\right) : \left(\frac{a}{b} - \frac{c}{d}\right)$$

Figura 5, Ejercicio seleccionado.

En las siguientes dos reuniones se logra discutir las soluciones de los ejercicios seleccionados de fracciones y potencias enfatizando la cantidad de pasos utilizados y aclarando problemas generados cuando no se aplican las habilidades de manera óptima.

La Figura 6 muestra un ejemplo indicado por el experto donde el estudiante no logra identificar el uso de factorización o factorizar por la base con menor exponente al ser un ejercicio de carácter numérico o con exponente negativo.

2. Evalúe:

$$\begin{aligned}
 &(2^{-3} - 2^{-5})^{-2} \\
 &(2^{-3})^{-2}(1 - 2^{-2})^{-2} \\
 &(2^6)(1 - \frac{1}{4})^{-2} \\
 &(2^6)(\frac{3}{4})^{-2} \\
 &(2^6)(3 \cdot 2^{-2})^{-2} \\
 &(2^6)(3^{-2})(2^4) \\
 &(2^{10})(3^{-2}) \\
 &\frac{1024}{9}
 \end{aligned}$$

2. Evalúe:

$$\begin{aligned}
 &(2^{-3} - 2^{-5})^{-2} \\
 &(2^{-5})^{-2}(2^2 - 1)^{-2} \\
 &(2^{10})(3)^{-2} \\
 &\frac{1024}{9}
 \end{aligned}$$

Figura 6, Evidencia de diferencia en cantidad de pasos.

Finalmente, se decide por estructurar los ejercicios utilizando la estructura de contenido definida en la plataforma Learner Model GQL como guía.

4.3 Estructuración del contenido

La Tabla 4 presenta la estructura principal en que se discuten los ejercicios con los expertos mencionados. Esta tabla logra resumir de manera implícita los requerimientos definidos en los ocho principios de la teoría ACT-R.

Tabla 4, estructura de ejercicios.

Enunciado		Expresión		
Objetivo a desarrollar en el ejercicio		Expresión matemática que se debe desarrollar		
Step	Instrucción	Expresión	Respuesta	KCs
Número de paso	Instrucción precisa de lo que se debe realizar.	Expresión a trabajar en el paso.	Respuesta esperada para la instrucción.	Etiquetado de los componentes de conocimiento.
Step	Hint	Errores comunes		
Número de paso	Listado que contiene tres pistas	Listado de errores comunes.		

4.4 Análisis cognitivo por tipo de ejercicio

Para el análisis cognitivo se procederá a desglosar la asociación del contenido de los tipos de ejercicios desarrollados con los ocho principios de la teoría ACT-R mostrados en la Tabla 5.

Tabla 5, Ocho principios de la teoría ACT-R

Principio	Aspectos relevantes rescatados del principio
1. Representar la competencia estudiantil como un conjunto de producciones.	Modelo preciso de la habilidad objetivo, esto permite definir objetivos apropiados para la interpretación adecuada de la acción del estudiante.
2. Comunicar la estructura subyacente objetiva de resolución del problema.	Resolver un problema involucra la descomposición en objetivos y subobjetivos. Si la estructura objetiva gobernante en la resolución de un problema no es comunicada adecuadamente, esta se debe definir de manera explícita la cual era implícita en la instrucción.
3. Proveer instrucción en el contexto de resolución del problema.	Se debe proveer una sección que permita al estudiante obtener otras referencias a la instrucción. Poner las referencias como parte del ejercicio puede entorpecer el desarrollo realizado por el estudiante.
4. Promover el entendimiento abstracto del conocimiento de resolución del problema.	Es común que el estudiante desarrolle conocimiento muy específico para resolver un único problema. Esto significa que las reglas de producción no son lo suficientemente generales.
5. Minimizar la carga de memoria trabajada.	Trabajar una nueva regla de producción requiere información relevante activa en la memoria. Esto dificulta el aprendizaje de información objetiva. Se debe minimizar la información presentada durante la resolución del problema.
6. Proveer retroalimentación inmediata sobre errores.	Se debe buscar la reducción de tiempo en el estado de error y facilitar la interpretación de resolución del problema del estudiante.
7. Ajustar la granularidad de instrucción de aprendizaje.	Hay evidencia insuficiente indicando que la reducción de reglas de producción no influye en la adquisición de habilidades por parte del estudiante.
8. Facilitar aproximaciones sucesivas de la habilidad específica.	Es común que el estudiante no sea capaz de realizar todos los pasos. Con la práctica el estudiante será capaz de completar más pasos.

La asociación de los principios se realiza en el modelado de contenido (ejercicios). En otras palabras, en el modelado de contenido, el principio “*p1*” indica que una competencia como factorizar se puede descomponer en objetivos menores asociables a habilidades (KC) que varían según el tipo de ejercicio, sin embargo, el principio “*p4*” limita esta descomposición para evitar que se desarrollen conocimientos de habilidades muy específicas las cuales no entregan mayor valor. Mientras que el principio “*p7*” entrega indicaciones tempranas que una competencia con menor descomposición como resolver un ejercicio/instrucción en un solo paso, no entrega mayor beneficio al momento de adquirir habilidades que resolver el mismo ejercicio/instrucción en múltiples pasos. Los principios “*p3*”, “*p6*” y “*p8*” buscan mantener al estudiante en el camino correcto sin bloqueos cognitivos, esto se puede cubrir parcialmente en los mensajes de ayuda (retroalimentación, referencias a la instrucción asociada y aproximaciones como entregar la respuesta en la última ayuda). El principio “*p5*” indica que la instrucción no debe saturar la memoria de trabajo con información de otras instrucciones.

La presentación del contenido (interfaz) también es importante y se menciona en los principios: “*p2*” indicando la presentación de un contexto de resolución bien comunicado, “*p4*” también es aplicable sobre las habilidades del contexto entregado para la resolución y “*p5*” limita el contexto de resolución para evitar saturar la memoria de trabajo. En otras palabras, la interfaz debe ser bien comunicada, no debe necesitar habilidades adicionales muy específicas y no debe mantener preocupado al estudiante con su funcionamiento.

4.4.1 Ejercicios de Fracción algebraica

4.4.1.1 Reducción de expresión

El ejercicio de la Figura 7 tiene como objetivo principal la reducción de una expresión algebraica de fracciones. Los sub-objetivos que pueden ser asociados a esta reducción son:

$$\left(\frac{a}{b} + \frac{c}{d}\right) : \left(\frac{a}{b} - \frac{c}{d}\right)$$

Figura 7, expresión a reducir.

Primero: Identificar que una división puede ser expresada como una fracción.

$$\frac{\left(\frac{a}{b} + \frac{c}{d}\right)}{\left(\frac{a}{b} - \frac{c}{d}\right)}$$

Figura 8, Solución esperada.

Este subobjetivo puede ser presentado como el primer paso con la instrucción:

[Paso 1] *Expresa la siguiente división como una fracción* (ver Figura 8).

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al estudiante al camino correcto son:

1. El lado izquierdo de una división se conoce como dividendo, el lado derecho se conoce como divisor.
2. Una fracción es una forma de expresar un número racional. Un número racional tiene forma P/Q donde P es el numerador y Q el denominador.
3. Para expresar esta división como número racional el dividendo pasa a ser el numerador, mientras que el divisor pasa a ser el denominador.

Segundo: Decidir si se debe resolver el numerador o denominador, la resolución en este caso particular implica operaciones anidadas de suma y resta de fracciones como se muestra en la Figura 9.

$$\left(\frac{ad+bc}{bd}\right), \left(\frac{ad-bc}{bd}\right)$$

Figura 9, Soluciones esperadas.

Este subobjetivo puede ser presentado como los siguientes dos pasos, con las instrucciones:

[Paso 2] *Realiza la suma de fracciones en el numerador.*

[Paso 3] *Realiza la resta de fracciones en el denominador.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al estudiante al camino correcto son:

1. Identifica el numerador/denominador, luego identifica si hay operaciones realizables en el numerador/denominador.
2. Existe una suma/resta de fracciones en el numerador/denominador, aplica la propiedad de suma de fracciones.
3. El numerador es la parte superior de la fracción./El denominador es la parte inferior de la fracción.

Tercero: Aplicar el principio de división de fracciones, luego de identificar fracciones tanto en el numerador y/o en el denominador.

$$\left(\frac{ad+bc}{bd}\right) \left(\frac{bd}{ad-bc}\right)$$

Figura 10, Solución esperada.

Este subobjetivo puede ser presentado como el cuarto paso con la instrucción:

[Paso 4] *Invierte la fracción ubicada en el denominador, luego expresa como multiplicación el resultado con el numerador.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al estudiante al camino correcto son:

1. Aplica la propiedad de división de fracciones.
2. La propiedad de división de fracciones indica que el denominador se invierte y pasa a multiplicar el numerador.
3. La solución deseada para este paso es: ver Figura 10.

Cuarto: Reducción final, como la división de fracciones implica la multiplicación del numerador con el inverso del denominador, se deberá identificar si hay elementos que se pueden cancelar o si se debe factorizar para finalmente identificar posibles cancelaciones.

Al identificar que no quedan factorizaciones o cancelación de términos posibles, se debe entender que la expresión ha sido reducida.

$$\frac{ad + bc}{ad - bc}$$

Figura 11, Solución esperada.

Este subobjetivo puede ser presentado como el quinto paso con la instrucción:

[Paso 5] *Simplifica la siguiente expresión cancelando los términos semejantes, luego multiplica ambas fracciones.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al estudiante al camino correcto son:

1. Aplica la propiedad de multiplicación de fracciones, si hay términos semejantes entre el numerador y denominador agrupalos a la izquierda.
2. La propiedad de multiplicación de fracciones indica que el producto de cada numerador pasa a ser un único numerador y el producto de cada denominador pasa a ser un único denominador.
3. La solución deseada para este paso es: ver Figura 11.

Los sub objetivos para este tipo de ejercicio se pueden agrupar bajo habilidades cognitivas nombradas de manera arbitraria como: “notación de división”, “identificar numerador/denominador”, “notación racional”, “Suma/resta de fracciones”, “División de fracciones”, “Multiplicación de fracciones” y “cancelación términos semejantes”.

4.4.1.2 Fracciones con MCM/MCD

El ejercicio de la Figura 12 tiene como objetivo principal realizar una suma de fracciones aplicando mínimo común múltiplo(MCM), en expresiones en que el denominador presenta bases de exponentes diferentes. Los sub-objetivos que pueden ser asociados a esta suma de fracciones son:

$$\frac{1}{b^2} + \frac{1}{a^2b}$$

Figura 12, expresión a sumar.

Primero: Identificar que para realizar la suma de fracciones se debe encontrar el MCM.

$$b^2, a^2b$$

Figura 13, Contexto de resolución.

Este sub-objetivo puede ser presentado como el primer paso con la instrucción:

[Paso 1] Encuentra el mínimo común múltiplo en los siguientes elementos (ver Figura 13).

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al estudiante al camino correcto son:

1. Para obtener el mínimo común múltiplo puedes comenzar encontrando los factores únicos.
2. El mínimo común múltiplo se obtiene multiplicando todos los factores (comunes y no comunes), elevados a la máxima potencia.
3. En este caso el mínimo común múltiplo es: (ver Figura 14).

$$a^2b^2$$

Figura 14, Solución esperada.

Segundo: Sumar las fracciones con el MCM encontrado, esto implica que el estudiante deberá establecer el denominador común. Luego expresar como sumandos el resultado de la multiplicación del MCM encontrado con el numerador original dividido por el denominador original correspondientes.

$$\frac{a^2 + b}{a^2b^2}$$

Figura 15, Solución esperada.

Este subobjetivo puede ser presentado como el segundo paso, con la instrucción:

[Paso 2] Suma las siguientes fracciones aplicando el mínimo común múltiplo encontrado.

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al estudiante al camino correcto son:

1. Al sumar aplicando mínimo común múltiplo, este se deja como denominador común.
2. Cada elemento del numerador en la fracción resultante estará compuesto del mínimo común múltiplo multiplicado por el numerador y dividido por el denominador de la fracción original correspondiente.
3. El resultado es: (ver Figura 15)

Los sub objetivos para este tipo de ejercicio se pueden agrupar bajo habilidades cognitivas nombradas de manera arbitraria como: “Mínimo común múltiplo.” y “Suma/resta de fracciones.”.

4.4.2 Ejercicios de Potencias

4.4.2.1 Evaluación de expresión numérica 1

El ejercicio de la Figura 16 tiene como objetivo principal evaluar una expresión numérica. Los subobjetivos que pueden ser asociados a esta evaluación son:

$$0,000350 \cdot 5000000 \cdot 0,0004$$

Figura 16, expresión a evaluar.

Primero: Transformar los factores a su notación científica correspondiente. Esto implica que el estudiante debe identificar la presencia de números grandes y decimales que dificultan un cálculo directo.

$$(3,5)(10^{-4})(5)(10^6)(4)(10^{-4})$$

Figura 17, Solución esperada.

Este subobjetivo puede ser presentado como el primer paso con la instrucción:

[Paso 1] *Expresa cada factor en su notación científica.* (ver Figura 17).

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Trabaja los números de manera individual luego expresa el resultado como su multiplicación.
2. Al trabajar un número identifica el primer dígito que dejaras como unidad, luego cuenta la cantidad de dígitos del número original hacia la derecha o izquierda de este hasta llegar a la unidad.
3. Traslada el número hasta la posición de unidad, luego multiplícalo por la base 10 elevada a la cantidad de dígitos contada, esta cantidad será positiva si contaste hacia la derecha y negativa si contaste hacia la izquierda.

Segundo: Multiplicar cada factor con base distinta a 10, luego multiplicar el producto obtenido con la expresión del producto de los elementos de base 10 con exponentes diferentes.

$$(70)(10^{-4})(10^{-4})(10^6)$$

Figura 18, Solución esperada.

Este subobjetivo puede ser presentado como el segundo, con la instrucción:

[Paso 2] *Identifica los números decimales y luego multiplicarlos, expresa el número resultante multiplicando los números de misma base.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Sin evaluar, agrupa los términos con base semejante.
2. Con fin de facilitar el cálculo, puedes agrupar los términos semejantes al lado derecho de la expresión.
3. La solución deseada para este paso es: (ver Figura 18)

Tercero: Aplicar la propiedad de suma de exponentes con misma base.

$$(70)(10^{-2})$$

Figura 19, Solución esperada.

Este subobjetivo puede ser presentado como el tercer paso con la instrucción:

[Paso 3] *Realiza la suma de las potencias en base 10.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Multiplica los términos con exponente.
2. Aplica la propiedad de suma de potencias en términos semejantes.
3. la solución deseada para este paso es: ver Figura 19

Cuarto: Transformar los factores a su notación científica correspondiente.

$$(7)(10)(10^{-2})$$

Figura 20, Solución esperada.

Este subobjetivo puede ser presentado como el cuarto paso con la instrucción:

[Paso 4] *Expresa el número 70 en su notación científica.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Sin multiplicar, identifica si hay números que se pueden expresar como notación científica.
2. Expresa el número 70 con su notación científica.
3. la solución deseada para este paso es: (ver Figura 20)

Quinto: Identificar que la expresión no puede ser reducida y reconocer que la solución puede ser expresada como, número decimal, fracción o notación científica.

$$(7)(10^{-1})$$

Figura 21, Solución esperada.

Este subobjetivo puede ser presentado como el quinto paso con la instrucción:

[Paso 5] *Realiza la suma de las potencias en base 10.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Aplica la propiedad de suma de potencias en los números de misma base.
2. Sin multiplicar, identifica si hay números que se pueden expresar como notación científica.
3. La solución deseada para este paso es: (ver Figura 21)

Los subobjetivos para este tipo de ejercicio se pueden agrupar bajo habilidades cognitivas nombradas de manera arbitraria como: “Notación científica”, “Multiplicación de números decimales” y “Suma/resta de exponentes en misma base”.

4.4.2.2 Evaluación de expresión numérica 2

El ejercicio de la Figura 22 tiene como objetivo principal evaluar una expresión numérica en su notación científica. Los sub-objetivos que pueden ser asociados a esta evaluación son:

$$(2^{-3} - 2^{-5})^{-2}$$

Figura 22, expresión a evaluar.

Primero: Factorizar la expresión dentro del paréntesis por la base con menor exponente. Esto implica que el estudiante sea capaz de reconocer el uso de factorización y que el exponente -5 es menor que el exponente -3.

Este subobjetivo puede ser presentado como el primer paso con la instrucción:

[Paso 1] *Dentro del paréntesis identifica el factor común y factoriza con su menor exponente.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Factoriza los términos semejantes dentro del paréntesis.
2. Identifica el término semejante con el menor exponente y utilízalo para factorizar.
3. La solución deseada para este paso es: ver Figura 23

$$((2^{-5})(2^2 - 1))^{-2}$$

Figura 23, Solución esperada.

Segundo: Identificar que uno de los factores dentro del paréntesis principal es fácil de evaluar, luego resolver el paréntesis principal aplicando la propiedad de multiplicación de potencias.

$$2^{10}3^{-2}$$

Figura 24, Solución esperada.

Este subobjetivo puede ser presentado como el segundo, con la instrucción:

[Paso 2] *Resuelve el paréntesis que contiene la resta, luego multiplica los exponentes.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Identifica si hay operaciones realizables dentro de cada paréntesis sin evaluar exponentes que dificulten el cálculo.
2. Aplica la propiedad de multiplicación de potencias en el exponente con mayor dificultad.
3. la solución deseada para este paso es: ver Figura 24

Tercero: Identificar que la expresión no puede ser reducida y reconocer que la solución puede ser expresada como, número decimal, fracción o notación científica.

$$\frac{1024}{9}$$

Figura 25, Solución esperada.

Este subobjetivo puede ser presentado como el tercer paso con la instrucción:

[Paso 3] *Evalúa la expresión y expresa el resultado como una fracción.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Expresa como un número racional.
2. Luego de expresar como un número racional, calcula las potencias en cada término.
3. La solución deseada para este paso es: ver Figura 25.

Los subobjetivos para este tipo de ejercicio se pueden agrupar bajo habilidades cognitivas nombradas de manera arbitraria como: “Identificación de menor exponente en misma base”, “Factorización por factor común”, “Multiplicación de potencias”, “Evaluación de números enteros con exponente”, “Suma/resta de números enteros” y “Notación racional”.

4.4.2.3 Reducción de expresión algebraica

El ejercicio de la Figura 26 tiene como objetivo principal la reducción de una expresión algebraica con exponentes. Los subobjetivos que pueden ser asociados a esta reducción son:

$$\frac{(a^{b+3} \cdot d^{b+2})^3}{a^{3b} \cdot d^6}$$

Figura 26, expresión a reducir.

Primero: Resolver la expresión presentada en el numerador, esto implica que el estudiante conozca aplicar las propiedades de multiplicación de potencias.

Este subobjetivo puede ser presentado como el primer paso con la instrucción:

[Paso 1] *En el numerador, realiza la multiplicación de potencias.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. La multiplicación de potencias tiene forma: ver Figura 27
2. Cuando la operación de los términos dentro de un paréntesis es la multiplicación se puede aplicar la siguiente propiedad: ver Figura 28
3. Expresa los términos del numerador dentro del paréntesis como la multiplicación de potencias, cada término del numerador dentro del paréntesis queda como la multiplicación de su exponente con el exponente del paréntesis.

$$(a^n)^m = a^{n \cdot m}$$

Figura 27, Propiedad de potencias en matriz cuadrada.

$$(a \cdot b)^n = a^n \cdot b^n$$

Figura 28, Propiedad de potencias de un producto.

Segundo: Multiplicar el numerador por el inverso del denominador, esto implica que el estudiante debe identificar que la cancelación de términos se origina a partir de una operación de potencias.

$$a^{3b+9} \cdot d^{3b+6} \cdot a^{-3b} \cdot d^{-6}$$

Figura 29, Solución esperada.

Este subobjetivo puede ser presentado como el segundo, con la instrucción:

[Paso 2] *Invierte los factores en el denominador.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Puedes obtener el inverso de una expresión elevando por la potencia -1.
2. Recuerda que después de obtener el inverso realiza la multiplicación de potencias correspondiente.
3. El resultado en este escenario es: ver Figura 29

Tercero: Aplicar la propiedad de suma de potencias en igual base e identificar que la expresión ha sido reducida.

Este subobjetivo puede ser presentado como el tercer paso con la instrucción:

[Paso 3] *Realiza la suma/resta de potencias en cada factor de igual base.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Identifican si hay operaciones por realizar en cada exponente.
2. Las operaciones de multiplicación/división/suma/resta dentro de un exponente se pueden realizar como si fuese una expresión normal.
3. Realiza la reducción aplicando las restas correspondientes en cada exponente.

Los subobjetivos para este tipo de ejercicio se pueden agrupar bajo habilidades cognitivas nombradas de manera arbitraria como: “Multiplicación de potencias”, “Invierte denominador” y “Suma/resta de exponentes en misma base”.

4.4.2.4 racionalización de expresión

El ejercicio de la Figura 30 tiene como objetivo principal racionalizar una expresión algebraica. Los subobjetivos que pueden ser asociados a esta reducción son:

$$\frac{1}{\sqrt{a} + \sqrt{b}}$$

Figura 30, Expresión a racionalizar.

Primero: Eliminar las raíces del denominador, el estudiante debe identificar que se puede encontrar el conjugado del denominador obteniendo una suma por su diferencia.

Este subobjetivo puede ser presentado como el primer paso con la instrucción:

[Paso 1] *Identifica el conjugado del denominador, luego deja expresado la racionalización del denominador.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Racionalizar es la técnica que permite eliminar raíces en el numerador y/o denominador en un número racional.
2. Una forma de eliminar raíces en un sumando es encontrar su conjugado para obtener una suma por diferencia.
3. Amplifica este número racional por 1 usando el conjugado encontrando como numerador y denominador.

Segundo: Multiplicar el numerador por el inverso del denominador, esto implica que el estudiante debe identificar que la cancelación de términos se origina a partir de una operación de potencias.

Este subobjetivo puede ser presentado como el segundo, con la instrucción:

[Paso 2] *Multiplifica ambas fracciones, recuerda aplicar suma por diferencia en el denominador.*

Mientras que los mensajes de retroalimentación sucesivos que buscan guiar al usuario al camino correcto son:

1. Identifica si hay operaciones por realizar entre estos números racionales.
2. La multiplicación de dos números racionales tiene la siguiente forma: ver Figura 31
3. El resultado esperado es: ver Figura 32.

$$\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$$

Figura 31, Propiedad de fracciones.

$$\frac{\sqrt{a}-\sqrt{b}}{a-b}$$

Figura 32, Solución esperada.

Los subobjetivos para este tipo de ejercicio se pueden agrupar bajo habilidades cognitivas nombradas de manera arbitraria como: “Racionalización de numerador/denominador con conjugado”, “Amplificación de fracción por 1”, “Multiplicación de fracciones” y “Identificación de suma por diferencia”.

4.5 Aclaraciones y supuestos

Importante destacar que estas habilidades agrupan un conjunto de producciones que generan el conocimiento asociado al aplicar de manera procedural definiciones declarativas. Estas agrupaciones pueden variar al identificar la necesidad de ajustar la granularidad, por ejemplo, agregar una habilidad “factorizar” que agrupe todos los contenidos de factorización es muy general y es muy probable que el estudiante se equivoque con mayor frecuencia.

También es destacable que la lógica de selección del orden de pasos es semejante a la definición del curso normal de eventos en ingeniería de programación. Donde se busca resolver una secuencia de eventos de manera lineal evitando un posible incremento del desarrollo.

Finalmente destacar que el mensaje final de retroalimentación busca entregar la respuesta, para permitir que el estudiante pueda resolver el ejercicio de manera fluida y se espera con normalidad que existan pasos que no sean capaces de resolver.

5. DOCUMENTO DE REQUERIMIENTOS DE PRODUCTO

5.1 Introducción

Este capítulo tiene como objetivo presentar de mejor manera el producto LvlTutor. Además, se explicará a fondo los distintos escenarios de uso, los requisitos que este producto deberá cumplir para que pueda ser lanzado al mercado y la arquitectura que se necesitará para que funcione de forma adecuada. Por último, se entregará la información sobre la estimación, para indicar que tanto recursos humanos y costos tendrá el producto en su desarrollo.

5.2 Propósitos del producto

Es entregar una solución de interfaz para los contenidos de nivelación matemática modelados (ejercicios) de los tópicos de fracciones y potencias. Este producto trabaja con la selección de los ejercicios modelados a partir de tópicos mencionados con apoyo de expertos en el área de ingeniería, de manera que sean implementados en una aplicación que será accedida desde la nube usada por estudiantes entrantes a la educación superior que busquen nivelar sus habilidades a las necesarias para cursar el curso de álgebra de manera exitosa.

5.3 Los principios de LvlTutor

LvlTutor posee tres principios básicos. Estos son:

- i. Intuición.
- ii. Guiar.
- iii. Mejorar.

Todo esto nos permite diseñar soluciones que intuitivamente guíen la mejora desde la organización hasta el valor percibido por el usuario.

5.4 Perfil de usuario final y escenario

Xiaochun estudiante de ingeniería piensa que la álgebra es difícil. Esto es importante porque le baja la motivación y considera la deserción. Cuando conversa con otros compañeros, Xiaochun estudiante de ingeniería ve que tienen rendimientos diferentes entre sí. Cuando obtiene un desempeño bajo en álgebra, Xiaochun siente frustración porque siente que no sirve para la ingeniería. Xiaochun estudiante de ingeniería realiza estudio de álgebra al menos una vez por unidad. Xiaochun estudiante de ingeniería gasta la sesión de estudio completa en intentar resolver los ejercicios propuestos de álgebra por cada unidad.

Xiaochun no tiene dominio en las habilidades de álgebra, entonces es difícil para él tener buen desempeño en la universidad. Termina reprobando el curso de álgebra y considera la deserción universitaria. Xiaochun va a clases, pero el ritmo de la clase no logra resolver sus deficiencias en álgebra.

5.5 Problemas encontrados

Los estudiantes como Xiaochun representan alrededor del 53% de los mechones de ingeniería en la UACH. Debido a que no son capaces de identificar sus deficiencias de manera específica, se enfrentan de cara con ejercicios que “creen” poder resolver luego

de adquirir el conocimiento declarativo durante la clase, sin embargo, carecen del conocimiento procedural y de las habilidades involucradas para transferir del contexto mostrado en la clase al contexto presentado en el texto en que trabajan. Esta ilusión de falsa creencia del saber y el hacer termina colapsando con sentimientos de frustración, rabia, tristeza, entre otros., al no poder avanzar en el estudio según las expectativas personales.

5.6 Solución a los problemas detectados

LvlTutor es una aplicación orientada a mejorar la experiencia del estudiante en la nube durante la resolución de expresiones matemáticas.

LvlTutor presenta los contenidos de álgebra de manera clara, precisa, intuitiva y guiada para así permitir a los estudiantes de la Universidad Austral de Chile adquirir y/o fortalecer sus habilidades que serán mostradas mediante un mejor desempeño académico.

LvlTutor busca suplir al modelo del estudiante mediante trazas del conocimiento, para así permitir a los investigadores de la Universidad Austral de Chile continuar su búsqueda en la mejora de la educación.

5.7 Declaración de producto

LvlTutor podrá guiar intuitivamente al estudiante que haga su uso en la nube durante la resolución de expresiones matemáticas presentes en los tópicos de nivelación de fracciones y potencias algebraicas para así incentivar la mejora.

5.8 Riesgos

El entorno sobre el cual se utilizan los dispositivos móviles puede influir en la percepción e inmersión del estudiante dificultando la completitud de la tarea presente. Las expresiones matemáticas determinarán una barrera inicial para el estudiante. El diseño de interfaz puede quedar obsoleto si las tendencias en los paradigmas de aplicaciones móviles cambian de manera considerable en el mercado.

5.9 Objetivos del producto

- a) Entregar servicios para la resolución de expresiones matemáticas presentes en los tópicos de nivelación de fracciones y potencias algebraicas.
- b) Guiar intuitivamente al estudiante a mejorar o evidenciar la adquisición y/o fortalecimiento de las habilidades usadas durante la resolución de expresiones matemáticas.

5.10 Funciones de LvlTutor

- 1. Permitir la resolución de expresiones matemáticas presentes en los tópicos de nivelación de fracciones y potencias algebraicas previamente seleccionadas.
- 2. Facilitar el ingreso de operaciones matemáticas de manera intuitiva.
- 3. Presentar retroalimentación al estudiante mediante mensajes que permitan guiar al estudiante hacia el camino correcto.
- 4. Almacenar las acciones realizadas por el estudiante durante la resolución de las expresiones matemáticas.

5.11 Arquitectura del sistema

Los componentes desarrollados serán desplegados en la plataforma Learner Model GQL la cual hace uso de un estilo de arquitectura de microservicios aprovechando el uso de GraphQL, además los componentes del ecosistema serán desarrollados haciendo uso de TypeScript, react y el entorno de trabajo proporcionado por Next.js.

5.12 Criterio de publicación

LvlTutor debe permitir la resolución de ejercicios y generación de trazas del conocimiento del estudiante en los contenidos de nivelación para fracciones y potencias algebraicas. Los contenidos serán obtenidos a partir de la plataforma Learner Model GQL. Por lo cual se considera satisfactorio la validación de expertos para el diseño de los tipos de ejercicios y la validación de la solución “LvlTutor” mediante un laboratorio.

5.13 Requerimientos

Las reuniones realizadas para el refinamiento de la selección de contenido permitieron entender las necesidades principales para la implementación de la solución. Luego de realizar la última reunión de refinamiento se termina por verificar los requerimientos detectados para la solución.

5.13.1 Requisitos Funcionales

- (Esencial) Permitir la resolución de expresiones matemáticas.
- (Esencial) Ingreso de expresiones matemáticas.
- (Esencial) Validación de respuestas ingresadas
- (Esencial) Solicitud de mensajes de ayuda.
- (Esencial) Reporte de acciones.

5.13.2 Requisitos No funcionales

- (Esencial) Despliegue correcto de los ejercicios modelados.
- (Esencial) Definición de los componentes de conocimiento asociados en la resolución de expresiones matemáticas seleccionadas.
- (Esencial) Facilidad de uso.
- (Esencial) Intuitivo.
- (Esencial) Documentar las metodologías y códigos desarrollados.
- (Esencial) Validación del producto final.

5.13.3 Requisitos adicionales nacidos durante el proyecto

- (Esencial) Diferentes tipos de validación.
- (Esencial) Ingreso flexible del operador multiplicación en expresiones algebraicas.

5.14 Casos de uso

5.14.1 Resolución de un tipo de ejercicio

Actores: Estudiante

Precondición: El estudiante debe tener acceso a la plataforma Learner Model GQL.

Propósito: Permitir la resolución de expresiones matemáticas.

Descripción: Un estudiante registrado en la plataforma Learner Model GQL podrá realizar acciones que permitan el desarrollo de una expresión matemática paso a paso.

5.14.2 Ingreso de expresiones matemáticas

Actores: Estudiante

Precondición: El estudiante debe tener acceso a la plataforma Learner Model GQL.

Propósito: Facilitar de manera intuitiva el ingreso de operaciones matemáticas.

Descripción: Un estudiante registrado en la plataforma Learner Model GQL podrá ingresar operaciones matemáticas para el desarrollo de expresiones matemáticas presentadas en un elemento de interfaz gráfica.

5.14.3 Validación de respuestas ingresadas

Actores: Estudiante

Precondición: El estudiante debe tener acceso a la plataforma Learner Model GQL.

Propósito: Validar el ingreso de la respuesta ingresada.

Descripción: La respuesta ingresada por el estudiante contará con una validación matemática.

5.14.4 Solicitud de mensaje de ayuda

Actores: Estudiante

Precondición: El estudiante debe tener acceso a la plataforma Learner Model GQL.

Propósito: Guiar al estudiante al camino de resolución recomendado.

Descripción: Un estudiante registrado en la plataforma Learner Model GQL podrá acceder a un elemento de interfaz gráfica que proporcionará un mensaje de ayuda durante la resolución de una expresión matemática.

5.14.5 Reporte de acciones

Actores: Estudiante, Learner Model GQL

Precondición: El estudiante debe tener acceso a la plataforma Learner Model GQL.

Propósito: Generar trazas de conocimiento

Descripción: La plataforma Learner Model GQL registrará acciones generadas por el estudiante.

5.15.6 Diagrama de casos de uso

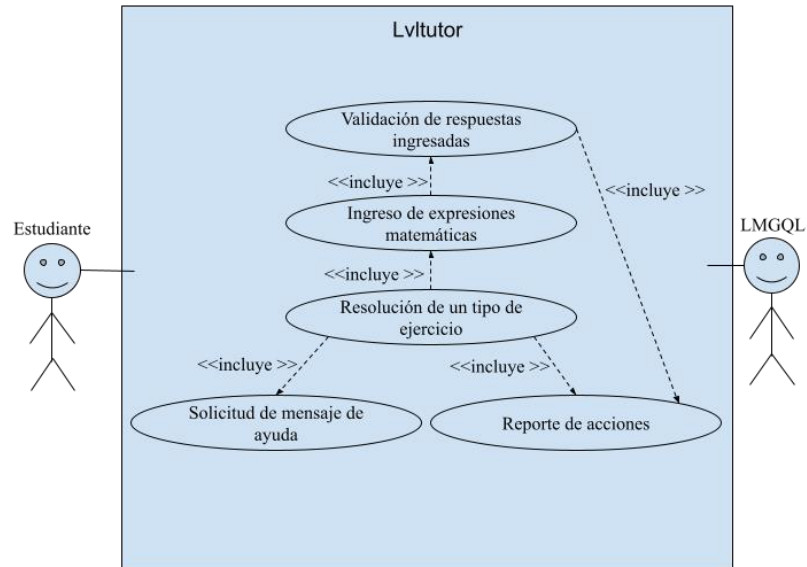


Figura 33, Diagrama de casos de usos.

5.15 Estimación

Para la estimación se utilizó el método de puntos de casos de usos propuesto por Karner en 1993, Además se ajustó la estimación de horas-hombre base de 20 horas a 8.2 horas según los resultados obtenidos por Subriadi et al, (2014). El resultado de la estimación final puede ser observado en la Tabla 6.

Tabla 6, Estimación por puntos de uso

Caso de uso	Horas Hombre
Resolución de un tipo de ejercicio	31.16
Validación de respuestas ingresadas	28.97
Ingreso de operaciones matemáticas	28.77
Solicitud de mensaje de ayuda	31.75
Tiempo total	120.65

Finalmente se debe destacar que esta estimación fue levantada al comienzo del proyecto y no considero el tiempo de desarrollo para el laboratorio de evaluación o de los requerimientos adicionales que fueron implementados de manera ágil pero caótica al no formalizarse con el cliente más allá de lo verbal. Mientras que el diseño de los ejercicios tomó alrededor de dos meses, proceso que incluye el trabajo realizado por expertos educacionales, patrocinante y tesista.

5.16 Gestión de configuraciones de cambios y versiones

Debido a presupuestos, tamaño del equipo y licencias, es que se selecciona el uso de la herramienta Git versión 2.35.1 para para el control de versiones de manera local durante desarrollo de software y github como servicio de control de versiones en la nube.

5.17 Crecimiento de los datos del producto

Las definiciones de las estructuras y modelos de datos junto a la estimación de la carga no son alcance de este proyecto, debido a que son trabajados en la plataforma Learner Model GQL (ver Sáez, 2022 y Nahuelpán, 2021).

5.18 Bosquejos

La Figura 34 muestra el bosquejo que considera la implementación de LvlTutor para dispositivos móviles. Este bosquejo fue descartado por el patrocinante debido al trabajo adicional que implica el desarrollo en una plataforma diferente. Como alternativa el patrocinante propone seguir el esquema presentado en la Figura 35 para el desarrollo paso a paso de un ejercicio. Finalmente siguiendo lo propuesto en este documento se decide mantener un ingreso intuitivo de expresiones junto a una botonera mostrada en la pantalla de resolución de la Figura 34.

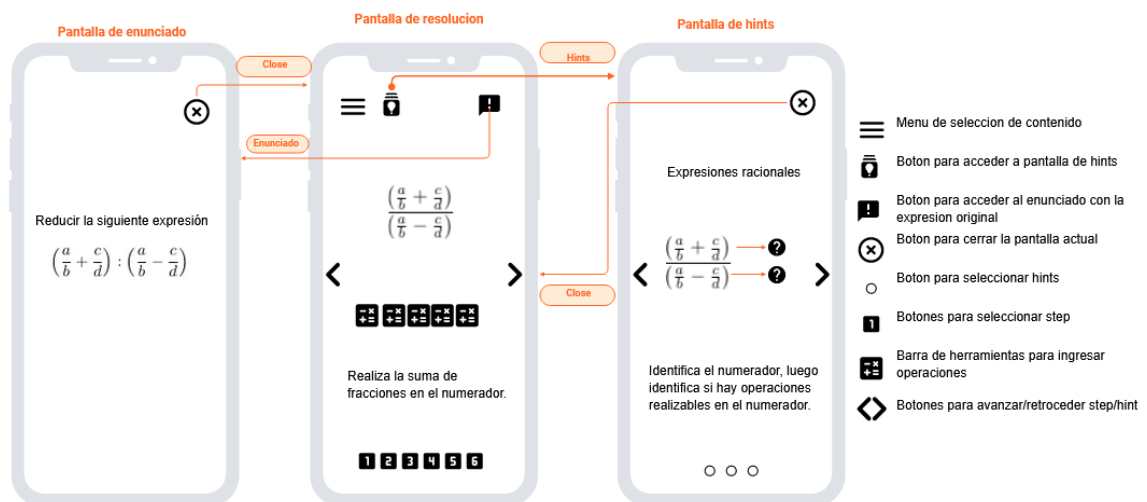


Figura 34, bosquejo de aplicación móvil

Factorización / Factor común

Factorizar la siguiente expresión:

$$6xy^2 - 15x^2y + 21x^2y^2$$

Ingresa factor común:

$6xy^2 - 15x^2y + 21x^2y^2$ () (?)

Factor común incorrecto, ingrese el factor común correspondiente a la expresión dada

Figura 35, Notificación de un nuevo hint al cometer un error.⁵

⁵ Nahuelpán, M. (2021). Tutor cognitivo y modelado de habilidades matemáticas para nivelación de estudiantes de primer año. Trabajo para optar al título de Ingeniero Civil en Informática.

6. DESARROLLO DE LA SOLUCIÓN

Luego de presentar la documentación de diseño, se detecta un problema relacionado con la escritura de las expresiones matemáticas en ASCII al ser poco claro para el estudiante, facilitando casos de ambigüedad cuando no se tienen las habilidades de álgebra bien desarrolladas. En el bosquejo mostrado en la Figura 34 se logra graficar la necesidad de crear una barra de herramientas con el fin de facilitar el ingreso de expresiones, sin embargo, esto es insuficiente para solucionar las dificultades presentadas por la escritura en ASCII. Afortunadamente en el mercado hay diferentes soluciones para presentar la escritura en LaTeX como Mathjax y KaTeX entre otras, mientras que Mathquill es una solución que facilita una transición de la escritura en ASCII a la escritura en LaTeX. Luego de presentar las diferentes alternativas al patrocinante para no generar una mayor divergencia en la gobernanza del proyecto, se decidió utilizar la solución ofrecida por Mathquill. Esto permite comenzar a estudiar las diferentes tecnologías involucradas para luego continuar con el desarrollo de los casos de uso presentados en el documento de requerimientos de producto (PRD).

6.1 Conceptos tecnológicos relevantes para el desarrollo

El entorno de trabajo presentado por Next.JS con TypeScript y React se orienta a trabajar en componentes con estados y definiciones estrictas. La manera en que el código es interpretado es secuencial. La definición de estados introduce actualizaciones dinámicas al cambiar de valor, permitiendo a la plataforma comenzar el proceso de refresco sobre los componentes donde el estado cambió. Mientras que TypeScript busca traer una escritura estandarizada de código para evitar ambigüedades de interpretación.

6.2 Digitalización de contenido

Por convenciones definidas en el proyecto la selección de ejercicios refinada se transformó al formato de ejercicios definido en una estructura JSON, este formato fue definido por Nahuelpán (2021). Esta estructura ha evolucionado a lo largo del proyecto la cual se podrá observar en la Tabla 7 y la Tabla 8. Adicionalmente se creó el componente “ExerciseType.tsx” que almacena las definiciones de las estructuras actualizadas utilizadas en el proyecto.

Tabla 7, Estructura donde se almacena un ejercicio.

Code	Identificador del ejercicio
Meta	Información relevante interna sobre el ejercicio
Title	Título para el contenido
Text	Instrucción general del ejercicio
Type	Etiqueta para identificar el tipo de implementación a utilizar en la carga del ejercicio
Steps	Lista de pasos

Tabla 8, Estructura donde se almacena un paso.

Stepid	Número del paso
KCs	Lista de componentes de conocimiento
Expression	Expresión a resolver
Steptitle	Instrucción del paso
Displayresult	Respuesta ideal
Values	Lista de valores para las variables de la expresión
Hints	Lista de ayudas
Matchingerror	Lista de errores
Answers	Lista de respuestas correctas
Incorrectmsg	Mensaje de error
Correctmsg	Mensaje de respuesta correcta
Summary	Texto que explica lo realizado en el paso

6.3 Metodología de desarrollo de código

Se definió la metodología incremental de desarrollo para guiar la ejecución del proyecto. El orden de implementación de los módulos definidos en diagrama de casos de uso mostrado en la Figura 33 del PRD es: 1.Resolución de un tipo de ejercicio, 2.Ingreso de expresiones matemáticas y 3.Validación de respuestas ingresadas. Los módulos de reporte de acciones e hints ya contaban con implementaciones en el proyecto. Luego de la implementación de la primera iteración se realizó un laboratorio de validación de producto presentado en el siguiente capítulo, sin embargo, por temas prácticos los resultados de la segunda iteración se incluyen en la presentación de los diferentes módulos.

6.4 Resolución de un tipo de ejercicio

Este caso de uso busca presentar el ejercicio completo de manera ordenada. La Figura 36 muestra el despliegue de resolución de un ejercicio comenzando con el despliegue del título, instrucción y expresión iniciales. Luego utilizando el componente “Accordion” se despliegan los pasos del ejercicio cuando no han sido resueltos y la respuesta cuando han

sido resueltos correctamente. Finalmente la Figura 37 muestra el despliegue del resumen al completar el ejercicio. Este es el componente principal donde su código se encuentra en el archivo “Solver2.tsx”.

Fracción Algebraica
Reduce la siguiente expresión:
$$\frac{\left(\frac{a}{b} + \frac{c}{d}\right)}{\left(\frac{a}{b} - \frac{c}{d}\right)}$$

Realiza la suma (o resta) de fracciones en el numerador y expresa el resultado final del numerador.

$\frac{ad+cb}{bd}$ } C. Despliegue de respuesta ingresada

✓ Has realizado las operaciones en el numerador correctamente.

Realiza la suma (o resta) de fracciones en el denominador y expresa el resultado final del denominador.

$\frac{\left(\frac{a \cdot d + b \cdot c}{b \cdot d}\right)}{\left(\frac{a}{b} - \frac{c}{d}\right)}$

B. Despliegue del componente para ingreso de operaciones

Invierte la fracción ubicada en el denominador, luego expresa como multiplicación el resultado con el numerador.

Simplifica la siguiente expresión cancelando los términos semejantes, luego multiplica ambas fracciones.

A. Despliegue de resolución de un ejercicio

Figura 36, Ejemplo de resolución de un ejercicio.

Simplifica la siguiente expresión cancelando los términos semejantes, luego multiplica ambas fracciones

$\frac{ad+bc}{ad-bc}$

✓ Has logrado reducir la expresión correctamente.

Resumen

Expresión: $\frac{\left(\frac{a}{b} + \frac{c}{d}\right)}{\left(\frac{a}{b} - \frac{c}{d}\right)}$

- 1) Luego se resuelve las operaciones en el numerador.
 $\left(\frac{a \cdot d + b \cdot c}{b \cdot d}\right)$
- 2) Se continúa resolviendo las operaciones en el denominador.
 $\left(\frac{a \cdot d - b \cdot c}{b \cdot d}\right)$
- 3) Con el fin de dividir una fracción se multiplica el numerador (la fracción de arriba) con el inverso del denominador (la fracción de abajo).
 $\left(\frac{a \cdot d + b \cdot c}{b \cdot d}\right) \cdot \left(\frac{b \cdot d}{a \cdot d - b \cdot c}\right)$
- 4) Luego se logra reducir la expresión eliminando los términos semejantes. Finalmente se aplica la multiplicación entre ambas fracciones.
 $\frac{a \cdot d + b \cdot c}{a \cdot d - b \cdot c}$

Figura 37, Despliegue del resumen al terminar el ejercicio.

Para la comunicación entre componentes se utilizó la solución ofrecida por Valtio para generar un intermediario en tiempo real. Básicamente funciona definiendo un objeto el cual se desea instanciar con sus valores actuales o con un constructor inicial. Esto permite solucionar problemas generados por la comunicación entre componentes que necesiten una alternativa dentro del servidor para mantener comunicación en tiempo real. La Figura 38 muestra el código con la estructura que contiene las variables de interés dentro de la aplicación definidas en el componente “MQProxy.tsx”.

```

1  import { proxy } from "valtio";
2  import type { Step } from "../ExercicseType";
3
4  interface value {
5    ans: string;
6    att: number;
7    hints: number;
8    lasthint: boolean;
9    fail: boolean;
10   duration: number;
11 }
12
13 interface sharedValues {
14   content: string;
15   step: Step | null;
16   topicId: string;
17   defaultIndex: Array<number>;
18   submit: boolean;
19   submitValues: value;
20   startDate: number;
21   endDate: number;
22   hints: number;
23   error: boolean;
24 }

```

Figura 38, Código de interfaces con variables de interés.

6.5 Ingreso de expresiones matemáticas

Para un público objetivo con dominio en las tecnologías de información, realizar operaciones matemáticas utilizando la escritura ASCII se transforma en una habilidad adquirida. Sin embargo, asumir que estudiantes de primer año con dificultades en álgebra tengan esta habilidad adquirida o tengan dominio en las tecnologías de información sería un error.

Frente al contexto presentado se decidió por aplicar un diseño de la interfaz con un ingreso de operaciones más amigable y cercano. Para lograr este fin, se utiliza la solución entregada por la tecnología Mathquill cuyo funcionamiento se puede observar tanto en la Tabla 9 y como en la Figura 39.

Tabla 9, Expresión transformada con Mathquill.

Expresión en ASCII	Expresión en latex	Resultado visual
((a/b)+(c/d))/((a/b)-(c/d))	\frac{\frac{a}{b}+\frac{c}{d}}{\frac{a}{b}-\frac{c}{d}}	$\frac{\frac{a}{b} + \frac{c}{d}}{\frac{a}{b} - \frac{c}{d}}$



Figura 39, Traducción automática de ASCII a lenguaje matematico.

La Figura 40, Barra de herramientas para ingreso de operaciones.muestra la barra de herramientas desarrollada para el ingreso de operaciones matemáticas con el fin de facilitar la escritura en caso que el usuario no tenga dominio sobre el uso del teclado. Finalmente, el código se encuentra en el archivo “MQ2.tsx” y en su implementación se sincroniza con el intermediario definido en “MQProxy.tsx” .



Figura 40, Barra de herramientas para ingreso de operaciones.

6.6 Validación de respuestas ingresadas

La validación de respuestas en las otras implementaciones de interfaz dentro del proyecto se estaban realizando mediante la comparación de texto. Este alcance es válido cuando se trabajan soluciones sencillas donde hay menor variación en el ingreso de respuestas por

el usuario. Cuando se trabajan con expresiones de cantidad de operandos y operadores mayor, se abre la oportunidad de escribir la solución con las variaciones producidas al aplicar los diferentes axiomas matemáticos en un orden diferente. Descartar la validación de estas variaciones producidas por las permutaciones factibles dentro de una expresión y la asociación con un estudiante que está aprendiendo, pueden generar errores en su aprendizaje que serán más difíciles de identificar.

Como solución a este tema presentado en la validación, se propone validar la expresión mediante una evaluación matemática al reemplazar las variables con valores definidos en los ejercicios. En primera instancia esta solución funcionó bien, sin embargo, luego en el laboratorio de validación se detectó la necesidad de robustecer este acercamiento haciendo uso del error relativo, debido a que una evaluación podría terminar con diferentes valores por error de cálculo en punto flotante. Luego por solicitud del patrocinante se decide en evaluar de manera estricta al contar la cantidad de operadores presentes dentro de la expresión, acercamiento que permite las permutaciones diferentes pero limita que alguien ingrese la solución en una forma diferente. Finalmente se flexibiliza la opción de elegir qué tipo de validación utilizar mediante la lectura de un atributo definido en el ejercicio.

6.6.1 Traductor

Para poder realizar la validación numérica de una expresión escrita en LaTeX se desarrolló un traductor de expresión infija a postfija mediante una derivación del pseudocódigo mostrado en la Figura 41. Este algoritmo fue modificado para incluir la operación unaria “-”. También se desarrolló una derivación más específica para transformar la operación prefija de latex “ $\frac{a}{b}$ ” a una operación infija “a/b”. Finalmente se desarrolló otra derivación para flexibilizar la escritura algebraica al agregar multiplicación entre términos que no representen una palabra reservada como nombre de funciones o operadores. El código del traductor se encuentra en “MQPostfixParser.tsx”

La Figura 42 muestra el código para la preparación realizada sobre la palabra de entrada donde principalmente se reemplazan definiciones de latex a ascii como paréntesis izquierdo definido “ $\left($ ” a “(” o la transformaciones de fracción y de agregar multiplicación. Finalmente se inicializan las variables utilizadas.

La derivación del traductor desarrollado trabaja con la lectura de caracteres dentro de una palabra de entrada. A medida que se leen los caracteres se va construyendo una palabra literal o numérica hasta construir el nombre de una palabra reservada y tratarla como el token trabajado en el pseudocódigo mostrado en la Figura 41. Mientras que los operadores, paréntesis y precedencia se trabajan de la misma forma.

```

/* The functions referred to in this algorithm are simple single argument functions such as sine, inverse or
factorial. */
/* This implementation does not implement composite functions, functions with a variable number of arguments, or
unary operators. */

while there are tokens to be read:
  read a token
  if the token is:
    - a number:
      put it into the output queue
    - a function:
      push it onto the operator stack
    - an operator o1:
      while (
        there is an operator o2 at the top of the operator stack which is not a left parenthesis,
        and (o2 has greater precedence than o1 or (o1 and o2 have the same precedence and o1 is left-
associative))
      ):
        pop o2 from the operator stack into the output queue
      push o1 onto the operator stack
    - a "(":
      while the operator at the top of the operator stack is not a left parenthesis:
        pop the operator from the operator stack into the output queue
    - a left parenthesis (i.e. "("):
      push it onto the operator stack
    - a right parenthesis (i.e. ")"):
      while the operator at the top of the operator stack is not a left parenthesis:
        {assert the operator stack is not empty}
        /* If the stack runs out without finding a left parenthesis, then there are mismatched parentheses. */
        pop the operator from the operator stack into the output queue
      {assert there is a left parenthesis at the top of the operator stack}
      pop the left parenthesis from the operator stack and discard it
      if there is a function token at the top of the operator stack, then:
        pop the function from the operator stack into the output queue
/* After the while loop, pop the remaining items from the operator stack into the output queue. */
while there are tokens on the operator stack:
  /* If the operator token on the top of the stack is a parenthesis, then there are mismatched parentheses. */
  {assert the operator on top of the stack is not a (left) parenthesis}
  pop the operator from the operator stack onto the output queue

```

Figura 41, algoritmo *shunting yard*.

```

163 const MQinfixToPostfix = (word: String) => {
164   let a = word;
165   a = a.replace(/\\right\\)/g, "");
166   a = a.replace(/\\left\\(/g, "(");
167   a = a.replace(/}/g, ")");
168   a = a.replace(/{/g, "(");
169   a = a.replace(/\\cdot/g, "*");
170   a = a.replace(/\\sqrt/g, "sqrt");
171   if (a.search("frac") != -1) a = fracctoInfix(a);
172   a = lazyMath(a);
173   a = a.replace(/\\)/g, ")*(");
174   let l = a.length;
175   let literal = "";
176   let numeric = "";
177   let cOp = "";
178   let stack: Array<string> = [];
179   let output = "";
180   //^:start of string,+:1 or more times,*:0 or more times,$:end of string,flag i:case insensitive
181   let alphabet = new RegExp(/^[a-zA-Z]$/);
182   let number = new RegExp(/^[0-9.]$/);

```

Figura 42, Preparación del algoritmo.

La Figura 43 muestra el fragmento de código que construye las palabras literales. Luego decide agregar la palabra a la pila de operadores en caso de ser una palabra reservada o agregar la palabra a la cadena de salida.


```

183     for (let i = 0; i < l; i++) {
184         let value = a[i];
185         if (!value) continue;
186         if (alphabet.test(value)) {
187             literal = literal + value;
188             if (i == l - 1) output = output + " " + literal;
189         } else if (typeof reservedWords[literal as keyof typeof reservedWords] != "undefined") {
190             //if the literal word formed is a function push into operator stack
191             stack.push(literal);
192             literal = "";
193         } else {
194             //if the literal word is not a function add it to the output
195             if (
196                 literal.length > 0 &&
197                 typeof operator[literal as keyof typeof operator] == "undefined"
198             ) {
199                 output = output + " " + literal;
200                 literal = "";
201             }
202         }

```

Figura 43, Construcción de una palabra reservada.

La Figura 44 muestra el fragmento de código que construye las palabras numéricas y luego decide en agregar la palabra a la cadena de salida.

```

203     if (number.test(value)) {
204         numeric = numeric + value;
205         if (i == l - 1) output = output + " " + numeric;
206     } else {
207         if (numeric.length > 0) {
208             output = output + " " + numeric;
209             numeric = "";
210         }
211     }

```

Figura 44, Construcción de una palabra numérica.

La Figura 45 muestra el fragmento de código que manejan los operadores. En este fragmento se incluye el tratado de la operación unaria “-” y se transforma al operador “\um”. Luego mientras que la pila de operadores no esté vacía se verifica la precedencia y tipo de asociatividad entre el operador actual y el operador almacenado en la cima de la pila, para luego decidir agregar la cima a la palabra de salida. Finalmente se agrega el operador actual a la pila de operadores.

```

212     if (
213         typeof operator[value as keyof typeof operator] != "undefined" ||
214         typeof operator[literal as keyof typeof operator] != "undefined"
215     ) {
216         if (typeof operator[value as keyof typeof operator] != "undefined")
217             cOp = operator[value as keyof typeof operator];
218         else {
219             cOp = literal;
220             literal = "";
221         }
222         if (".".localeCompare(cOp) == 0 && 1 - i > 1) {
223             //detecting if "-" is an unary operator
224             let leftvalue = a[i - 1];
225             if (
226                 i == 0 ||
227                 (leftvalue != undefined &&
228                     i > 0 &&
229                     (typeof operator[leftvalue as keyof typeof operator] != "undefined" ||
230                         "(".localeCompare(leftvalue) == 0))
231             ) {
232                 cOp = "\\um";
233             }
234         }
235         let skip = true;
236         while (
237             stack.length > 0 && //if the stack has an operator
238             skip && //if the top operator has greater precedence
239             (precedence[stack[stack.length - 1] as keyof typeof precedence] >
240                 precedence[cOp as keyof typeof precedence] || //if they have same precedence
241                 (precedence[stack[stack.length - 1] as keyof typeof precedence] ==
242                     precedence[cOp as keyof typeof precedence] && //if current operator is left associative
243                     (typeof associativity[value as keyof typeof associativity] != "undefined"
244                         ? associativity[value as keyof typeof associativity] == "left"
245                         : false)))
246         ) {
247             let leftvalue = stack[stack.length - 1];
248             //if the top operator is not "("
249             if (leftvalue != undefined && "(".localeCompare(leftvalue) != 0)
250                 output = output + " " + stack.pop();
251             else skip = false;
252         }
253         stack.push(cOp);

```

Figura 45, Manejo de operadores.

La Figura 46 muestra la continuación del fragmento de código que manejan los operadores donde se trabajan los casos de paréntesis izquierdo y derecho. Si el paréntesis es izquierdo se agrega a la pila de operadores, mientras que si el paréntesis es derecho se completa la palabra de salida con los operadores encontrados en la cima de la pila hasta identificar el primer paréntesis izquierdo encontrado donde básicamente se ordena las operaciones encontradas dentro de un paréntesis en notación postfija.

La Figura 47 muestra el fragmento de código que manejan los operadores rezagados que necesitan ser agregados a la palabra de salida. Como se indica en el pseudocódigo mostrado en la Figura 41, si el paréntesis en la cima de la pila es izquierdo entonces existen paréntesis sin conjugar (la expresión fue escrita incorrectamente por el usuario).

```

254     } else if (value.localeCompare("(") == 0) {
255         stack.push("(");
256     } else if (value.localeCompare(")") == 0) {
257         let skip = true;
258         while (stack.length > 0 && skip) {
259             let leftvalue = stack[stack.length - 1];
260             //if the top operator is not "("
261             if (leftvalue != undefined && "(".localeCompare(leftvalue) != 0)
262                 output = output + " " + stack.pop();
263             else {
264                 skip = false;
265             }
266         }
267         let leftvalue = stack[stack.length - 1];
268         if (stack.length > 0 && leftvalue != undefined && "(".localeCompare(leftvalue) == 0)
269             stack.pop();
270         if (
271             stack.length > 0 &&
272             typeof reservedWords[stack[stack.length - 1] as keyof typeof reservedWords] != "undefined"
273         )
274             output = output + " " + stack.pop();
275     }
276 }

```

Figura 46, Continuación del manejo de operadores.

```

277     let skip = true;
278     while (stack.length > 0 && skip) {
279         let leftvalue = stack[stack.length - 1];
280         if (leftvalue != undefined && "(".localeCompare(leftvalue) != 0)
281             output = output + " " + stack.pop();
282         else skip = false;
283     }
284     return output;
285 };
286
287 return MQinfixToPostfix(MQinfixInput);
288 };

```

Figura 47, Manejo de operadores rezagados.

La Tabla 10 muestra un ejemplo del cambio en las variables inicializadas durante la ejecución del código:

Tabla 10, Ejemplo de ejecución del código

Sean la entrada: T = "sqrt(a*b)", salida: s = "", literal: l = "", stack: st = [], operador: op = "", se tiene:			
i=0	T[0] = "s", l = "s"	i=5	T[5] = "a", l = "a"
i=1	T[1] = "q", l = "sq"	i=6	T[5] = "*", s = " a", l = "", op = "*", st = ["sqrt", "(", "*"]
i=2	T[2] = "r", l = "sqr"	i=7	T[7] = "b", l = "b"
i=3	T[3] = "t", l = "sqrt"	i=8	T[8] = ")", s = " a b * sqrt"
i=4	T[4] = "(", l = "", st = ["sqrt", "("]	fin	

6.6.2 Calculadora

El funcionamiento de la calculadora postfija es más sencillo. Primero se recibe la palabra de entrada y se reemplazan las variables con un valor numérico definido en el ejercicio, en caso en que no estén definidos se utiliza el valor uno, lo que indica la importancia de definir bien los valores en caso de existir valores críticos. Segundo se define una pila donde se van agregando los valores según la lectura de la palabra de entrada hasta encontrar un operador, para así aplicar la operación sobre los primeros dos elementos encontrados en la cima de la pila de valores y almacenar el resultado en la misma pila luego de expulsar los elementos trabajados. Finalmente se continúa con el mismo sistema hasta terminar de leer la palabra, mientras que para la operación unaria “-” definida como “\um” solo multiplica la cima de la pila de operadores con “-1”. El código de la calculadora se puede encontrar en el archivo “MQPostfixSolver.tsx”.

6.6.3 Comparación estricta

El acercamiento para la comparación estricta primero considera la cantidad de cada tipo de elemento encontrado en la palabra postfija. Luego se realiza la comparación de estas cantidades entre la respuesta ingresada por el usuario y la respuesta almacenada en el ejercicio. Este acercamiento se diferencia a la comparación de texto al ignorar las permutaciones en que se puede ingresar una respuesta, sin embargo, requiere de la comparación del error relativo por que de la misma manera se ignora las permutaciones donde no se aplican los axiomas matemáticos de manera correcta. Esta solución fue implementada para evitar que el usuario ingrese la misma expresión a ser resuelta como solución.

Como sugerencia para permitir un cierto nivel de flexibilidad dado que igual se aplica la comparación del error relativo, se debe comparar las cantidad en porcentajes para así permitir la comparación con respuestas ingresadas correctamente pero en forma diferente, de lo contrario se deberá considerar agregar cada forma de la respuesta a la lista de respuestas. Por ejemplo si $4a+4b$ es una respuesta, otra forma seria $4(a+b)$. El código de la comparación estricta se puede encontrar en el archivo “MQPostfixstrict.tsx”.

6.7 Solicitud de mensaje de ayuda

Para el caso de uso de solicitud de mensaje de ayuda se decide utilizar el componente desarrollado por Nahuelpan (2021). Este componente recibe la respuesta ingresada por el estudiante, luego verifica si es diferente a la anterior. También recibe la lista de ayudas a desplegar cuando se verifica una respuesta incorrecta diferente a la anterior. Dentro del proceso para el modelo del estudiante, este componente reporta las acciones cuando se hace uso de una ayuda. El código se encuentra en el archivo “Hint.js” y la Figura 48 muestra una referencia para el despliegue de un hint.

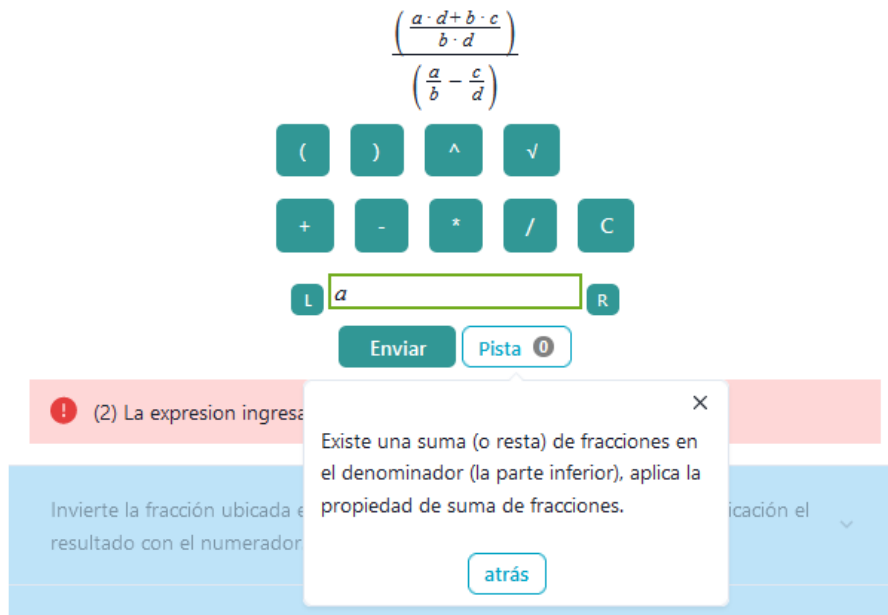


Figura 48, Despliegue de una ayuda.

6.8 Reporte de acciones

Para el caso de uso de reporte de acciones se decide utilizar el componente *action* definido por Saez (2022). Este componente define una mutación en graphql para reportar las diferentes acciones realizadas por el estudiante. Por solicitud del patrocinante, las acciones que se deben reportar son: Cargar el contenido, abrir/cerrar un paso del ejercicio, respuesta enviada por el estudiante y contenido completado. Mientras que el reporte de acciones para las ayudas ya fue implementado en el componente *Hint* desarrollado por Nahuelpán (2021). El código se puede encontrar en el archivo “Action.ts”.

6.8 Diagramas estructurales y de comportamiento

Para facilitar el entendimiento de la solución desarrollada, Figura 49 muestra el diagrama estructural de componentes para LvlTutor. Mientras que la figura 50 muestra el diagrama de secuencia para la carga de un ejercicio y la figura 51 muestra el diagrama de secuencia para la resolución de un ejercicio.

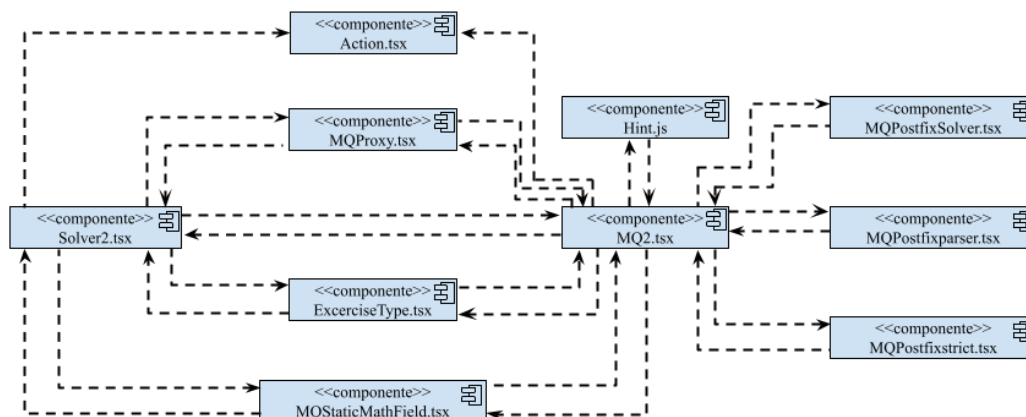


Figura 49, Diagrama de componentes de LvlTutor.

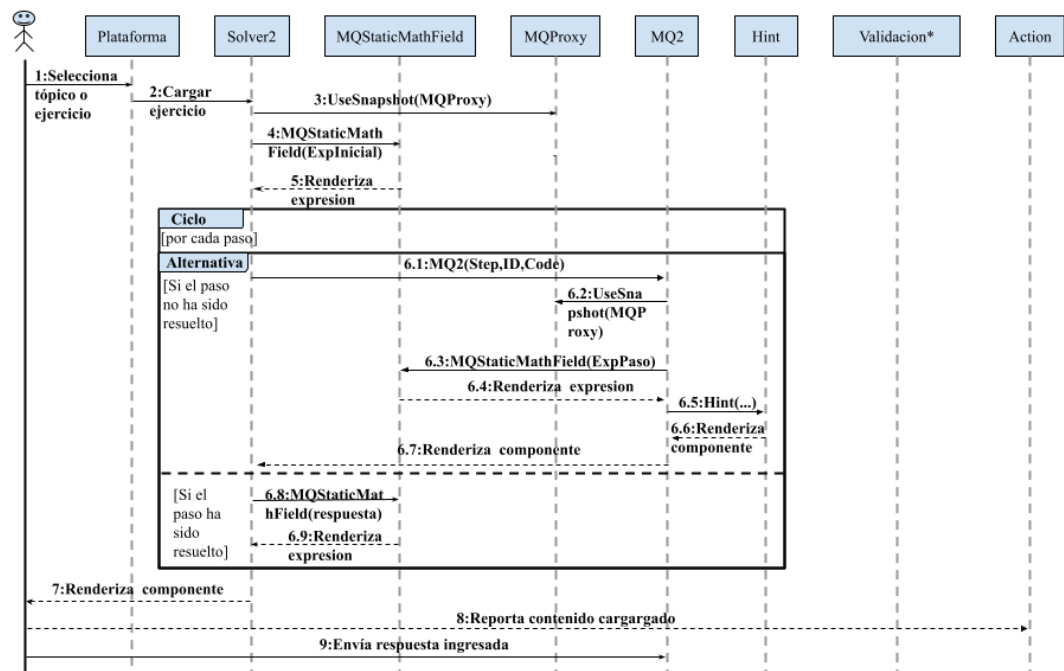


Figura 50, Diagrama de secuencia para el cargado de un ejercicio.

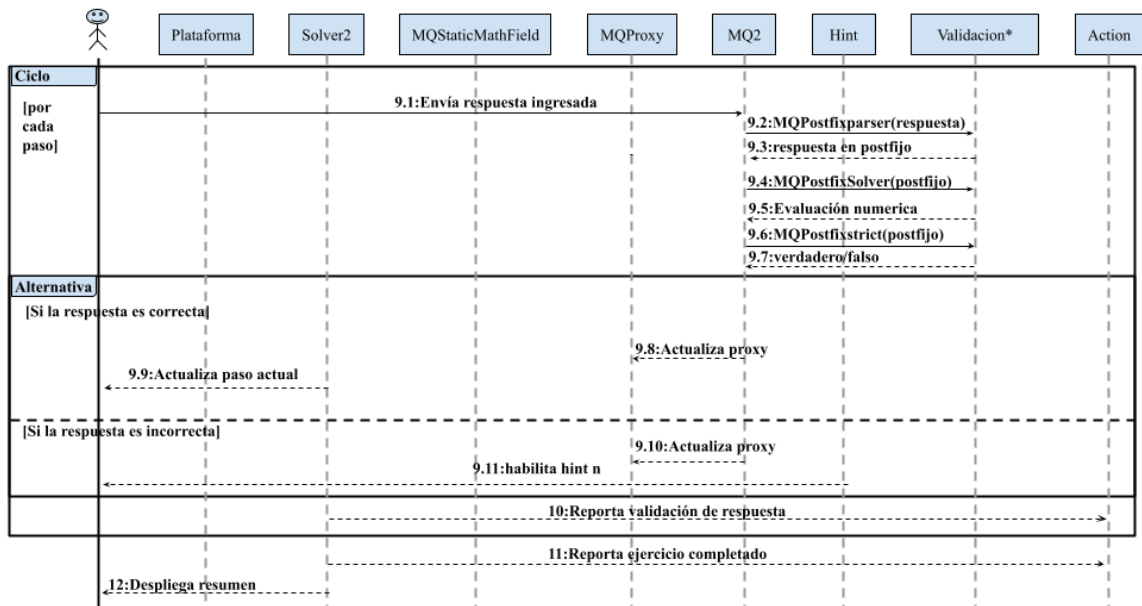


Figura 51, Diagrama de secuencia para la resolución de un ejercicio.

7. VALIDACIÓN DE INTERFAZ

La validación de interfaz se realizó mediante un laboratorio inspirado en el esquema presentado por la evaluación experimental y la configuración de grupos mediante la estrategia de cuadrados latinos para medir aspectos de interés durante el uso de la solución por usuarios reales.

7.1 Evaluación experimental

La evaluación experimental propone controlar aspectos específicos del comportamiento interactivo. La evaluación comienza definiendo un perfil de usuario representativo, luego se propone una hipótesis de interés a probar. La hipótesis definida será guiada mediante el uso de elementos de control relacionados con variables independientes como estilo de interfaz o cantidad de ítems en un menú y variables dependientes como tiempo transcurrido o número de errores. Después de obtener los resultados se propone que los cambios de comportamiento se deben a condiciones diferentes, ya sean definidas o imprevistas (Dix A., et al, 2003). El perfil de usuario fue definido en el capítulo cinco.

7.3 Definición de hipótesis

La finalidad de este experimento es la evaluación del producto desarrollado y su eficacia en el contexto de la presentación y facilidad de uso de la interfaz. La escritura en latex sin conocer de látex es una de las funcionalidades principales implementadas con el fin de homologar visualmente lo escrito por un usuario en la máquina, con lo presentado por un profesor de matemáticas en el aula. Otra funcionalidad de interés con necesidad de evaluación es la barra de herramientas y navegación junto la validación de respuestas. El fin de la barra de herramientas y navegación es facilitar aún más la escritura entregando elementos visuales en formato de botones cercanos al contexto de resolución. Finalmente se busca observar el comportamiento de la validación de respuestas ingresadas.

Para este experimento es importante definir el uso de la escritura tradicional con lápiz y papel, además de, el uso de la escritura de expresiones matemáticas en texto plano (caracteres ASCII) como elementos de contraste. Frente al contexto presentado se presenta la siguiente hipótesis: ***“La escritura en MQ2 requiere una menor cantidad de intentos que la escritura en ASCII”.***

7.4 Definición de variables

a) Variables de control independiente: Tipo de ingreso en ASCII, tipo de ingreso en MQ2.

b) Variables de control dependiente: tiempo de carga de página hasta ingreso del primer carácter, tiempo desde el ingreso del primer carácter a la publicación de respuesta de la respuesta correcta, cantidad de intentos utilizados.

7.5 Diseño experimental

Para el laboratorio se diseñaron dos fases, la primera fase busca comparar el tipo de ingreso, mientras que la segunda fase busca presentar los ejercicios diseñados en LvlTutor. Luego de terminar cada fase se realizaron preguntas de carácter abierto para fomentar la comunicación y reducir posible rigidez generadas al participar en un laboratorio. Al

terminar el tiempo asignado al laboratorio se realizó una encuesta de **escala de usabilidad del sistema (SUS)**, acompañada con una encuesta de satisfacción de diseño propio.

En la primera fase los participantes realizan el copiado de las expresiones mostradas en Tabla 11 mediante ambos tipos de entrada (MQ2 y ASCII), pero en orden diferente. El orden de resolución comienza con las expresiones fáciles, luego expresiones intermedias y terminan con las expresiones difíciles. Otra alteración del orden es realizada mediante la estrategia de cuadrados latinos para eliminar el ruido de experiencia adquirida al copiar las expresiones. La alteración del orden define cuatro grupos diferentes mostrados en la Tabla 12 que garantizan que cada participante tenga la experiencia de pasar por todas las expresiones a copiar, copie cada una usando ambas versiones (ASCII y MQ2) de manera no seguida y alternando entre ASCII y MQ2. La numeración de las expresiones mostradas en la Tabla 11 se salta los números cinco, diez y quince que corresponden a expresiones descartadas para el diseño de grupos.

Tabla 11, Expresiones definidas

n	Expresión fácil	n	Expresión intermedia	n	Expresión difícil
1	$a + b$	6	$(a + b) \cdot c$	11	$(a + b) \cdot (a^2 \cdot b - a)$
2	$\frac{a}{b}$	7	$(a + b)^c$	12	$\frac{a^2 + b^2}{c^{a+b}}$
3	$a \cdot b$	8	$\frac{c}{a+b}$	13	$\frac{1}{\sqrt{a} + \sqrt{b}}$
4	a^b	9	\sqrt{a}	14	$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$

La segunda fase busca evaluar el curso normal de resolución de los seis tipos de ejercicios desarrollados (o aquellos que los estudiantes alcanzaron desarrollar). La experiencia de usuario se verá reflejada formalmente en la encuesta SUS, mientras que informalmente se buscó obtener retroalimentación dirigida a los puntos de interés mediante la encuesta de satisfacción.

Tabla 12, Orden de expresiones por grupo de cuadrado latino.

	GRUPO	A	GRUPO	B	GRUPO	C	GRUPO	D
DIFICULTAD	EXPRESION	INPUT	EXPRESION	INPUT	EXPRESION	INPUT	EXPRESION	INPUT
FACIL	1	MQ2	2	ASCII	3	MQ2	4	ASCII
FACIL	2	ASCII	3	MQ2	4	ASCII	1	MQ2
FACIL	3	MQ2	4	ASCII	1	MQ2	2	ASCII
FACIL	4	ASCII	1	MQ2	2	ASCII	3	MQ2
FACIL	1	ASCII	2	MQ2	3	ASCII	4	MQ2
FACIL	2	MQ2	3	ASCII	4	MQ2	1	ASCII
FACIL	3	ASCII	4	MQ2	1	ASCII	2	MQ2
FACIL	4	MQ2	1	ASCII	2	MQ2	3	ASCII
INTERMEDIA	6	MQ2	7	ASCII	8	MQ2	9	ASCII
INTERMEDIA	7	ASCII	8	MQ2	9	ASCII	6	MQ2
INTERMEDIA	8	MQ2	9	ASCII	6	MQ2	7	ASCII
INTERMEDIA	9	ASCII	6	MQ2	7	ASCII	8	MQ2
INTERMEDIA	6	ASCII	7	MQ2	8	ASCII	9	MQ2
INTERMEDIA	7	MQ2	8	ASCII	9	MQ2	6	ASCII
INTERMEDIA	8	ASCII	9	MQ2	6	ASCII	7	MQ2
INTERMEDIA	9	MQ2	6	ASCII	7	MQ2	8	ASCII
DIFICIL	11	MQ2	12	ASCII	13	MQ2	14	ASCII
DIFICIL	12	ASCII	13	MQ2	14	ASCII	11	MQ2
DIFICIL	13	MQ2	14	ASCII	11	MQ2	12	ASCII
DIFICIL	14	ASCII	11	MQ2	12	ASCII	13	MQ2
DIFICIL	11	ASCII	12	MQ2	13	ASCII	14	MQ2
DIFICIL	12	MQ2	13	ASCII	14	MQ2	11	ASCII
DIFICIL	13	ASCII	14	MQ2	11	ASCII	12	MQ2
DIFICIL	14	MQ2	11	ASCII	12	MQ2	13	ASCII

7.5.1 Encuesta SUS

La escala de usabilidad del sistema (SUS) es una métrica compuesta de diez preguntas que busca entregar una vista subjetiva sobre las evaluaciones de usabilidad (Brooke, J., 1995). Esta encuesta fue construida cuidadosamente para fomentar respuestas extremas y se presenta al terminar de utilizar el producto luego de sugerir a los participantes que contesten impulsivamente sin pensar mucho en la respuesta. Para calcular el puntaje SUS se calcula la sumatoria de los puntajes después de transformarlos a una escala de [0,4] realizando las operaciones: (puntaje ítem impar)-1 y 5-(puntaje ítem par). Luego se multiplica el puntaje obtenido por 2.5, obteniendo así el puntaje general en una escala de [0,100] que no se debe confundir como porcentajes. El listado de preguntas SUS es:

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 el apoyo de un técnico para poder utilizar este sistema.
5. Encontré que las diversas funciones de este sistema estaban bien integradas.
6. Pensé que había demasiada inconsistencia en este sistema.
7. Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.
8. He encontrado el sistema muy complicado de usar.
9. Me sentí muy segura/o usando el sistema.
10. Necesité aprender muchas cosas antes de ser capaz de usar este sistema.

7.5.2 Encuesta de satisfacción

La encuesta de satisfacción fue construida con el fin de obtener retroalimentación específica sobre los elementos de interés. La puntuación para las primeras cinco preguntas

comienza en (1,desacuerdo) hasta (5, de acuerdo). Mientras que la pregunta seis es de carácter abierto donde el usuario explica con sus palabras las sugerencias.

1. ¿Encuentras la barra de herramientas de utilidad?
2. ¿Cómo describirías tu experiencia en el ingreso de expresiones matemáticas?
3. ¿Encuentras los enunciados claros?
4. ¿Encuentras las ayudas claras?
5. ¿El resumen ayudó en la reflexión?
6. ¿Si tuvieras que hacer uso de esta herramienta que te gustaría mejorar?

7.5.3 Guión del laboratorio

Introducción [3 minutos]

Junto a agradecer su participación les damos la bienvenida a este laboratorio de usabilidad de producto. Como objetivo buscamos validar el producto desarrollado y así detectar posibles mejoras para futuras mantenciones. El producto que se evaluará en esta ocasión es Lvtutor busca:

- 1. Permitir la resolución de expresiones matemáticas presentes en los tópicos de nivelación de fracciones y potencias algebraicas previamente seleccionadas.*
- 2. Entregar una barra de herramientas que permita ingresar operaciones matemáticas de manera intuitiva.*
- 3. Presentar ayuda al estudiante mediante mensajes que permitan guiar el camino correcto.*
- 4. Almacenar las acciones de manera anónima que son realizadas por el estudiante durante la resolución de las expresiones matemáticas.*

Para poder lograr la meta de este laboratorio, además de su participación, necesitaremos una buena comunicación. En otras palabras si encuentran temas que causan algún nivel de frustración o tienen dificultades para hacer uso de la herramienta, comunicar en el momento para así definir posibles mejoras a futuro.

Primera parte [25+1 minutos]

La primera parte busca evaluar el ingreso de expresiones matemáticas sin ser resuelta con diferentes tipos de entrada. El tiempo destinado al ingreso de expresiones es de veinte minutos, mientras que a las preguntas de discusión abierta se le destinaron cinco minutos. Las preguntas definidas en la discusión abierta son:

- 1. ¿Encuentras la barra de herramientas de utilidad?*
- 2. ¿Cómo describirías tu experiencia en el ingreso de expresiones matemáticas?*

Segunda parte [25+1 minutos]

La segunda parte busca evaluar la experiencia de usuario en la resolución de expresiones matemáticas de manera guiada. El tiempo designado para la resolución de ejercicios es de veinte minutos, mientras que se designó cinco

minutos para las encuesta sus y de satisfacción, finalizando con una discusión abierta sobre la experiencia en el laboratorio.

7.6 Implementación de aplicación para el laboratorio

La Figura 52 muestra el bosquejo diseñado para el ingreso de expresiones utilizado en la primera fase. Este bosquejo busca presentar un formulario y barra de herramientas tanto para el ingreso en ASCII o MQ2, mientras que la única variación se presenta en el campo de ingreso.

Definiciones	Ejemplo
Resta: -	a-b
Suma: +	a+b
Multiplicacion: *	a*b
Division: /	a/b
Exponente: ^	a^b
Raiz cuadrada: raiz(valor)	raiz(a)

Importante considerar el uso de parentesis

Ejemplo 1
7-4+2=5 7-(4+2)=1

Ejemplo 1
1^2+3-3 1^(2+3)=1

Figura 52, Bosquejo de ingreso de expresiones.

Para el desarrollo de código se siguió el tutorial ofrecido por Prisma y presentado por Mahmoud A, 2021. Se decidió utilizar este tutorial debido al uso de las mismas tecnologías utilizadas en el proyecto. Este acercamiento fue realizado para definir y utilizar una base de datos personalizada a las necesidades presentes del laboratorio, además de practicar conocimientos adquiridos en GraphQL.

7.7 Ejecución del laboratorio

Para la ejecución del laboratorio se estimó una duración de cincuenta y cinco minutos la cual se terminó extendiendo a un tiempo no mayor de noventa minutos (se perdió el registro explícito). La aplicación desarrollada fue desplegada en el equipo de desarrollo sobre el cual se configuró una red de acceso a las máquinas de laboratorio para que el usuario pueda acceder desde el navegador donde la página de inicio ya fue cargada con un identificador asignado por la aplicación internamente al cargar la página principal. Además del identificador se asignó un grupo perteneciente al conjunto [a, b, c, d].

La ejecución del laboratorio presentó numerosos problemas en la primera fase donde se busca comparar los tipos de escritura, comenzando con la pérdida de uno de los grupos al tener que cargar nuevamente el navegador, luego se detectó que la validación de respuestas para ASCII estaba mostrando el mensaje de respuesta correcta independiente del tipo de respuesta ingresada. Finalmente existieron usuarios que decidieron concentrarse en hacer ingeniería inversa para detectar cómo funciona dicha vulnerabilidad. Esto significó la pérdida de los datos del grupo b.

En la segunda fase, donde se evaluó el curso normal de la resolución de los tipos de ejercicios, se ejecutó de mejor manera. Se detectó un problema para el ingreso de potencias donde la configuración del teclado en las máquinas generaba el símbolo numérico del superíndice para las potencias cuando se esperaba que ingresara el símbolo “^”. Esta fase tardó más tiempo del considerado y se observó la existencia de usuarios sin dominio en las habilidades de álgebra.

7.9 Resultados

7.9.1 Datos de la primera fase

Para el procesamiento de los datos se realizaron evaluaciones con los componentes desplegados en producción con el fin de descartar ruido en las validaciones realizadas en el laboratorio. Los datos se obtuvieron de la base de datos trabajada en el laboratorio y luego transformados a formato json mediante código en python trabajado en la herramienta Jupyter. Las Figura 53, 54 y 56 muestran los datos de los ingresos realizados con ASCII, mientras que las Figura 54, 55 y 56 muestran los datos de los ingresos realizados con MQ2. Las definiciones de las etiquetas de las columnas se pueden observar en la Tabla 13.

Tabla 13, Definiciones de etiquetado

Expresión	Renderizado de la expresión original con Mathquill.
Input	Expresión en ASCII o renderizado con Mathquill para el ingreso de la respuesta del usuario según el tipo de ingreso con ASCII o Mathquill respectivamente.
EvExp	Evaluación numérica de la expresión original.
EvIn	Evaluación numérica de la expresión ingresada por el usuario.
= =	Comparación $EvExp == EvIn$, entregando el valor 1 si son iguales o el valor 0 si no.
ExpCorrecta	Validación de la cantidad de elementos remanentes en la pila de la calculadora para la expresión original, entregando el valor 1 si la expresión fue evaluada completamente y el valor 0 si no.
InCorrecto	Validación de la cantidad de elementos remanentes en la pila de la calculadora para la expresión ingresada por el estudiante, entregando el valor 1 si la expresión fue evaluada completamente y el valor 0 si no.
Err	Comparación del error relativo $Err(EvExp, EvIn) < 0.005$. Entregando 1 si Err es menor a 0.005 y 0 si no.
Estricto	Realiza el conteo de la cantidad de elementos únicos entre la expresión original y la expresión ingresada por el usuario. Luego compara que la cantidad entre los pares únicos de ambas expresiones sean iguales. Si la comparación es igual entrega el valor true, si no entrega el valor false.
Tiempo(S)	Tiempo en que el estudiante tardó en ingresar la expresión correcta o en ingresar tres intentos desde el tiempo en ingresar el primer carácter.

Principalmente se busca mostrar el último ingreso por el usuario para cada copiado de expresión y el comportamiento de la solución al hacer uso de los diferentes tipos de evaluación. Finalmente como el experimento cuenta con una población de tamaño igual a cuatro y debido a que solo se cuenta con los datos de tres participantes, es que este experimento no es suficiente para ser representativo y por ende los resultados solo sirven para explorar y obtener una percepción temprana de los comportamientos de la solución y del estudiante. La Tabla 14 resume la cantidad de respuestas enviadas por cada participante.

Tabla 14, Cantidad de envío de respuestas del participante (q) por tipo de entrada y usuario.

Expresión	Usuario D ASCII	Usuario D MQ2	Usuario A ASCII	Usuario A MQ2	Usuario C ASCII	Usuario C MQ2
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	8	3	4	2	3	2
$\frac{1}{\sqrt{a} + \sqrt{b}}$	1	3	4	2	3	1
$\frac{a^2 + b^2}{c^{a+b}}$	3	2	3	2	3	1
$\frac{a}{b}$	4	1	2	2	3	2
$\frac{c}{a+b}$	2	1	2	2	2	1
$(a+b) \cdot c$	1	3	2	1	1	3
$(a+b) \cdot (a^2 \cdot b - a)$	3	1	3	2	3	2
$(a+b)^c$	1	1	2	2	1	2
\sqrt{a}	1	3	3	2	2	1
$a+b$	3	2	2	2	1	3
$a \cdot b$	1	3	2	2	3	3
a^b	2	1	2	2	1	1

EXPRESION	INPUT	EVEXP	EVIN	==	EXPCORRECTA	INCORRECTO	ERR	ESTRICTO	TIEMPO(S)
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	0.5*10^-4 *23*10^2	0.11	0.11	1	1	1	1	true	26.11
$\frac{1}{\sqrt{a} + \sqrt{b}}$	1/raiz(b)-raiz(a)	0.33	3.00	0	1	1	0	false	30.65
$\frac{a^2+b^2}{c^{a+b}}$	a^2+b^2/c^b+c	17.00	18.00	0	1	1	0	false	34.99
$\frac{a}{b}$	a-b=943943	0.25	1.00	0	1	0	0	false	32.25
$\frac{c}{a+b}$	a/b+c	0.20	1.25	0	1	1	0	true	12.68
$(a+b) \cdot c$	(a+b)*c	5.00	5.00	1	1	1	1	true	7.01
$(a+b) \cdot (a^2 \cdot b - a)$	(a-b)*(a^2*a+a)	0.00	-6.00	0	1	1	0	false	30.75
$(a+b)^c$	(b+c)^a	5.00	5.00	1	1	1	1	true	23.23
\sqrt{a}	raiz(a)=0	1.00	1.00	1	1	0	1	false	7.08
$a+b$	a-b	5.00	-3.00	0	1	1	0	false	28.35
$a \cdot b$	a*b	4.00	4.00	1	1	1	1	true	6.53
a^b	a-b=448	1.00	1.00	1	1	0	1	false	6.8

Figura 53, Ingreso con ASCII para el Usuario D.

EXPRESION	INPUT	EVEXP	EVIN	==	EXPCORRECTA	INCORRECTO	ERR	ESTRICTO	TIEMPO(S)
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	$0.5 \cdot 10^{-4} \cdot 10^2 \cdot 23$	0.11	0.12	0	1	1	1	true	33.51
$\frac{1}{\sqrt{a} + \sqrt{b}}$	$\frac{1}{\sqrt{b} + \sqrt{c}}$	0.33	0.33	1	1	1	1	false	22.75
$\frac{a^2 + b^2}{c^{a+b}}$	$\frac{a^2 + b^2}{c^{a+b}} = 110$	17.00	17.00	1	1	0	1	false	50.07
$\frac{a}{b}$	$\frac{a}{b}$	0.25	0.25	1	1	1	1	true	28.3
$\frac{c}{a+b}$	$\frac{c}{a+b}$	0.20	0.20	1	1	1	1	true	11.3
$(a+b) \cdot c$	$(b-c)^c$	5.00	3.00	0	1	1	0	false	53
$(a+b) \cdot (a^{2 \cdot b} - a)$	$(a-c)(a^{2b} - a)$	0.00	0.00	1	1	1	0	false	22.3
$(a+b)^c$	$(b+a)^c$	5.00	5.00	1	1	1	1	true	8.57
\sqrt{a}	\sqrt{a}	1.00	1.00	1	1	1	1	true	18.73
$a+b$	$a+b$	5.00	5.00	1	1	1	1	true	13.6
$a \cdot b$	$\frac{a}{9} = 543243$	4.00	1.00	0	1	0	0	false	15.85
a^b	a^b	1.00	1.00	1	1	1	1	true	2.06

Figura 54, Ingreso con MQ2 para el Usuario D.

EXPRESION	INPUT	EVEXP	EVIN	==	EXPCORRECTA	INCORRECTO	ERR	ESTRICTO	TIEMPO(S)
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	0.5*(10^-4)*23*10	0.11	0.01	0	1	1	0	false	56.15
$\frac{1}{\sqrt{a} + \sqrt{b}}$	1/(raiz(a)+raiz(b))	0.33	0.20	0	1	1	0	false	49.15
$\frac{a^2+b^2}{c^{a+b}}$	((a^2)+(b^2))/((C^a+b))	17.00	3.40	0	1	1	0	true	103.56
$\frac{a}{b}$	a/b	0.25	0.25	1	1	1	1	true	9.52
$\frac{c}{a+b}$	c/a+b	0.20	5.00	0	1	1	0	true	13.45
$(a+b) \cdot c$	(a+b)*c	5.00	5.00	1	1	1	1	true	14.59
$(a+b) \cdot (a^{2 \cdot b} - a)$	(a+b)*((a^2*b)-a)	0.00	15.00	0	1	1	0	true	74.53
$(a+b)^c$	(a+b)^c	5.00	5.00	1	1	1	1	true	35.21
\sqrt{a}	raiz(a)	1.00	1.00	1	1	1	1	false	15.38
$a+b$	a+b	5.00	5.00	1	1	1	1	true	10.85
$a \cdot b$	a*b	4.00	4.00	1	1	1	1	true	8.11
a^b	a^b	1.00	1.00	1	1	1	1	true	24.04

Figura 55, Ingreso con ASCII para el Usuario A.

EXPRESION	INPUT	EVEXP	EVIN	==	EXPCORRECTA	INCORRECTO	ERR	ESTRICTO	TIEMPO(S)
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	0.11	0.11	1	1	1	1	true	47.34
$\frac{1}{\sqrt{a} + \sqrt{b}}$	$\frac{1}{\sqrt{a} + \sqrt{b}}$	0.33	0.33	1	1	1	1	true	25.87
$\frac{a^2+b^2}{c^{a+b}}$	$\frac{a^2+b^2}{c^{a+b}}$	17.00	17.00	1	1	1	1	true	89.78
$\frac{a}{b}$	$\frac{a}{b}$	0.25	0.25	1	1	1	1	true	6.56
$\frac{c}{a+b}$	$\frac{c}{a+b}$	0.20	0.20	1	1	1	1	true	9.45
$(a+b) \cdot c$	$(a+b) \cdot c$	5.00	5.00	1	1	1	1	true	10.64
$(a+b) \cdot (a^{2 \cdot b} - a)$	$(a+b) \cdot (a^{2 \cdot b} - a)$	0.00	0.00	1	1	1	0	true	31.83
$(a+b)^c$	$(a+b)^c$	5.00	5.00	1	1	1	1	true	14.51
\sqrt{a}	\sqrt{a}	1.00	1.00	1	1	1	1	true	147.19
$a+b$	$a+b$	5.00	5.00	1	1	1	1	true	5.26
$a \cdot b$	$a \cdot b$	4.00	4.00	1	1	1	1	true	4.96
a^b	a^b	1.00	1.00	1	1	1	1	true	52.05

Figura 56, Ingreso con MQ2 para el Usuario A.

EXPRESION	INPUT	EVEXP	EVIN	==	EXPCORRECTA	INCORRECTO	ERR	ESTRICTO	TIEMPO(S)
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	0.5 * 10^-4 * 23 * 10^2	0.11	0.11	1	1	1	1	true	68.48
$\frac{1}{\sqrt{a} + \sqrt{b}}$	1/raiz(a)+raiz(b)	0.33	5.00	0	1	1	0	false	30.87
$\frac{a^2 + b^2}{c^{a+b}}$	a^2 + b^2 / c^a+b	17.00	21.00	0	1	1	0	true	61.75
$\frac{a}{b}$	a/b	0.25	0.25	1	1	1	1	true	20.72
$\frac{c}{a+b}$	c/a+b	0.20	5.00	0	1	1	0	true	10.56
$(a + b) \cdot c$	(a + b)*c=8475	5.00	5.00	1	1	0	1	false	9.77
$(a + b) \cdot (a^{2 \cdot b} - a)$	(a + b) * (a^2*b - a)	0.00	15.00	0	1	1	0	true	76.58
$(a + b)^c$	(a+b)^c=3847	5.00	5.00	1	1	0	1	false	15.45
\sqrt{a}	raiz(a)	1.00	1.00	1	1	1	1	false	12.82
$a + b$	a + b	5.00	5.00	1	1	1	1	true	2.4
$a \cdot b$	a + b	4.00	5.00	0	1	1	0	false	77.34
a^b	a^b = 995	1.00	1.00	1	1	0	1	false	16.87

Figura 57, Ingreso con ASCII para el Usuario C.

EXPRESION	INPUT	EVEXP	EVIN	==	EXPCORRECTA	INCORRECTO	ERR	ESTRICTO	TIEMPO(S)
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	0.11	0.11	1	1	1	1	true	19.92
$\frac{1}{\sqrt{a} + \sqrt{b}}$	$\frac{1}{\sqrt{b} + \sqrt{a}}$	0.33	0.33	1	1	1	1	true	17.96
$\frac{a^2 + b^2}{c^{a+b}}$	$\frac{a^2 + b^2}{c^{a+b}}$	17.00	17.00	1	1	1	1	true	19.68
$\frac{a}{b}$	$\frac{a}{b}$	0.25	0.25	1	1	1	1	true	42.9
$\frac{c}{a+b}$	$\frac{c}{a+b}$	0.20	0.20	1	1	1	1	true	15.18
$(a+b) \cdot c$	$(a+b) \cdot c$	5.00	5.00	1	1	1	1	true	35.72
$(a+b) \cdot (a^{2 \cdot b} - a)$	$(a+b) \cdot (a^{2 \cdot b} - a)$	0.00	0.00	1	1	1	0	true	54.87
$(a+b)^c$	$(a+b)^c$	5.00	5.00	1	1	1	1	true	23.4
\sqrt{a}	$\sqrt{a} = 8283$	1.00	1.00	1	1	0	1	false	11.05
$a + b$	$a + b$	5.00	5.00	1	1	1	1	true	22.45
$a \cdot b$	$a \cdot b$	4.00	4.00	1	1	1	1	true	26.87
a^b	a^b	1.00	1.00	1	1	1	1	true	4.26

Figura 58, Ingreso con MQ2 para el Usuario C.

7.9.2 Resultados segunda fase

La Tabla 15 muestra el puntaje calculado en la encuesta de **Escala de usabilidad del sistema (SUS)** para los cuatro participantes del laboratorio. Sobre estos resultados se calculó la media con un valor de 53.75 y la desviación estándar con un valor 8.93.

Tabla 15, resultados de encuesta sus

	Participante 1	Participante 2	Participante 3	Participante 4
Resultados SUS (puntaje):	42.5	52.5	52.5	67.5

La Tabla 16 muestra los resultados de la encuesta de satisfacción donde se busca caracterizar la recepción de los diferentes aspectos de interés en la evaluación de LvlTutor, permitiendo al estudiante comentar libremente sobre que se debiese mejorar. Para calcular este puntaje se restó una unidad a los puntajes individuales ingresados a cada pregunta, luego se multiplicaron los valores por veinte y cinco obteniendo un puntaje en la escala [0-100] para cada ítem. Finalmente se obtiene la media de los puntajes calculados en los primeros cinco ítems.

Tabla 16, resultados de encuesta de satisfacción

	Participante 1	Participante 2	Participante 3	Participante 4
Resultados satisfacción (puntaje):	40	80	60	55

La Tabla 17 muestra la retroalimentación entregada en la pregunta abierta de la encuesta de satisfacción.

Tabla 17, retroalimentación del usuario

Pregunta: ¿Si tuvieras que hacer uso de esta herramienta, que te gustaría mejorar?

Participante 1:	Adaptabilidad al usuario, coherencia en instrucciones/enunciados. y una mejora en las ayudas (son muy ambiguas).
Participante 2:	Ejemplos visuales de los ejercicios que se están planteando, sirven como una guía por si se consigno escrito no queda claro.
Participante 3:	Las ayudas, ya que no son muy claras para todos.
Participante 4:	La expresión de algunas ayudas, que sean más explícitas a medida que la persona necesita más ayuda y que sean fácil de comprender.

7.9.3 Código de preparación de los datos

La Figura 59 muestra el código utilizado para transformar los datos al formato JSON trabajado en la aplicación utilizada para procesar los datos. El código de la aplicación utilizada para el procesamiento se puede encontrar en el repositorio github: “<https://github.com/zxFiro/DatosValidacionMQlab>”

```
import pandas as pd
acc = pd.read_csv("allActions.csv")
#Usuario 395=>grupo A,Usuario 400=>grupo D,Usuario 402=>grupo C
#Por fallas en la ejecucion del experimento no se cuenta con datos de un usuario grupo B
A_acc=acc[(acc["userId"] == 395)]
D_acc=acc[(acc["userId"] == 400)]
C_acc=acc[(acc["userId"] == 402)]
```

```
#Listado de expresiones originales para filtrar
Expresiones=[
    "0.5\\cdot10^{-4}\\cdot23\\cdot10^2",
    "\\frac{1}{\\sqrt{a}+\\sqrt{b}}",
    "\\frac{a^2+b^2}{c^{a+b}}",
    "\\frac{a}{b}",
    "\\frac{c}{a+b}",
    "\\left(a+b\\right)\\cdot c",
    "\\left(a+b\\right)\\cdot\\left(a^2\\cdot b-a\\right)",
    "\\left(a+b\\right)^c",
    "\\sqrt{a}",
    "a+b",
    "a\\cdot b",
    "a^b"]
```

```
import json
def dataByExp(df):
    answerList={"list":[]}
    dfa=df[(df["view"] == "EXPCOPY")]
    for e in Expresiones:
        dfIn={"ascii":{},"mathquill":{}}
        dfb=dfa[dfa["expression"]==e]

        dfAS=dfb[dfb["inputType"]=="ASCII"]
        dfAS=dfAS[["expression","expstep","createdAt","value"]]
        res1=dfAS.to_dict("split")
        dfIn["ascii"]=res1

        dfMQ=dfb[dfb["inputType"]=="MATHQUILL"]
        dfMQ=dfMQ[["expression","expstep","createdAt","value"]]
        res2=dfMQ.to_dict("split")
        dfIn["mathquill"]=res2

    answerList["list"].append(dfIn)
    return answerList
```

```
import os
cwd = os.getcwd()
path = cwd
dfList=[dataByExp(A_acc),dataByExp(D_acc),dataByExp(C_acc)]
csvL=["A395.json","D400.json","C402.json"]
for e1,e2 in zip(dfList,csvL):
    with open(e2, "w") as fp:
        fp.write(str(e1))
```

Figura 59, Código python trabajado en la herramienta Jupyter.

7.10 Análisis de resultados

Primero se debe considerar las siguientes fallas de implementación: 1) Error de cálculo en punto flotante mostrado en la Figura 60, 2) No se implementó la validación de raíz en ASCII como se muestra en la Figura 61. La primera falla solo se observó para el caso singular mostrado en la Figura 60. La segunda falla se presentó en el copiado con el input ASCII de las expresiones 9 y 14, obteniendo un comportamiento mixto. Para la expresión 9 el resultado fue validado como correcto si el usuario ingresaba la respuesta “raiz(a)”, sin embargo, esta validación se debe a la manera en cómo se implementó el código cuando hay variables/palabras sin definir, entregándoles un valor “1”. Esto significa que para la variable “a” definida con un valor “1”, mientras que las letras “r”, “i” y “z” de la palabra “raiz” se consideran como variables independientes con valor “1” por defecto al no estar definidas, entonces, la evaluación numérica para la expresión “raiz(a)” sería “1*1*1*1*1”, resultado que no varía del cálculo real “raiz(1)”, las multiplicaciones se agregan con el método “*lazymath*” implementado que permite escribir expresión sin agregar multiplicación entre variables/paréntesis. Para la expresión 14, el comportamiento anómalo presentado para la

validación de la expresión 9 no se repitió, debido a que la variable “b” se le asignó el valor “4” obteniendo la comparación $O:1/(1+2) == U:1/(1+4)$. La expresión original fue validada correctamente debido a que fue escrita en latex donde si se tiene considerada la implementación de la raíz cuadrada (como $\sqrt{\text{función}}$ a diferencia de $\text{raiz}(\text{función})$).

EXPRESION	INPUT	EVEXP	EVIN	==
$0.5 \cdot 10^{-4} \cdot 23 \cdot 10^2$	$0.5 \cdot 10^{-4} \cdot 10^2 \cdot 23$	0.11	0.12	0

Figura 60, Error de cálculo en punto flotante.

Escribe la siguiente expresion en el campo de entrada.

$$\frac{1}{\sqrt{a} + \sqrt{b}}$$

1/(raiz(a)+raiz(b))

Enviar

! La expresion ingresada no es correcta.

Ayudas para ingreso de operaciones

Suma	a+b
Resta	a-b
Multiplicacion	a*b
Division	a/b
Exponente	a^b
Raiz cuadrada	raiz(b)

Importante el uso de parentesis

Ejemplos

7-4+2=5	7-(4+2)=1
1^2*3=3	1^(2*3)=1

Figura 61, Error en la validación de raíz en ASCII.

Segundo, se pudo observar que dos participantes intentaron realizar ingeniería inversa como se muestra en la Figura 62. Este comportamiento comenzó luego de que un participante reportó la existencia de expresiones donde se validaba como correcto los ingresos incorrectos. Luego de analizar el comportamiento para la validación de la raíz en ASCII, se detectó que habían expresiones donde no se le había definido el valor a la variable “c”. Esto significa que los participantes podrían haber intercambiado las variables “a” y “c” sin afectar el resultado final. Finalmente, esta situación permitió detectar una nueva falla de implementación cuando la expresión no se evalúa completamente al agregar símbolos que rompen la evaluación en la calculadora al no tener operaciones definidas, este comportamiento también puede ser observado en la figura 59.

$$(a + b) \cdot c \quad (a + b) * c = 8475$$

Figura 62, Ingreso de un participante al intentar probar el funcionamiento.

Tercero, se tiene que considerar tanto la pérdida de los datos de uno de los grupos y el tamaño de la población $n=4$. Esto significa que la muestra es muy pequeña como para obtener mayores conclusiones.

Cuarto, hacer uso de los diferentes tipos de evaluación (evaluación numérica, error relativo y evaluación estricta) permite enrobustecer la validación de respuestas generando un efecto de triple redundancia que tiene el potencial de reducir la cantidad de falsos positivos en la evaluación de respuestas ingresadas.

Quinto, la Tabla 18 muestra que los tiempos para cada usuario no tienen mayor variación entre los tipos de entrada, sin embargo, dos de los tres usuarios tardaron menos con el ingreso en ASCII. Este comportamiento se puede ver como positivo para las escrituras con ASCII y MQ2, debido a que no existen diferencias considerables entre ambos tipos de escritura para un aspecto temporal.

Tabla 18, resumen del tiempo (en segundos) que un usuario tardó en pasar a la siguiente expresión

	Usuario D ASCII	Usuario D MQ2	Usuario A ASCII	Usuario A MQ2	Usuario C ASCII	Usuario C MQ2
Media	20.8	23.34	34.55	37.12	33.71	24.67
Desviación estándar	11.73	15.7	30.52	42.99	28.79	14

Sexto, el histograma mostrado en la Figura 63 da a entender que los usuarios pasan a la siguiente expresión con una menor cantidad de intentos con el tipo de ingreso MQ2. Esto respalda la hipótesis *"La escritura en MQ2 requiere una menor cantidad de intentos que la escritura en ASCII"*. Sin embargo los resultados son considerados preliminares y no concluyentes por el tamaño pequeño de la muestra. Una causa posible del incremento en el número de intentos ingresados en formato ASCII es el uso de paréntesis que se observa como deficiente, y MQ2 mejora esto con al autocompletar los paréntesis.

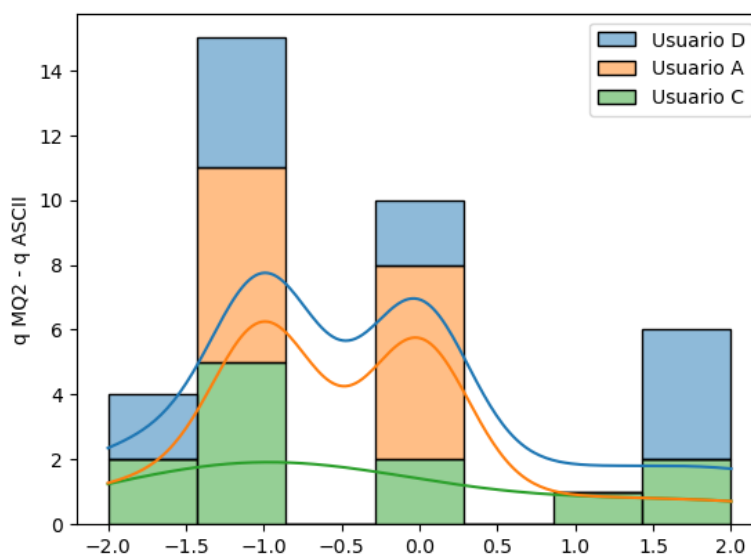


Figura 63, histograma para la diferencia de cantidad de acciones utilizadas (q) entre los ingresos MQ2 y ASCII. Valores negativos reflejan menos acciones con MQ2.

Finalmente, los resultados de las encuestas SUS y satisfacción dan resultados que evalúan la implementación como insatisfactoria. La razón de estos resultados se puede observar en la pregunta abierta de la encuesta de satisfacción, donde se reporta que la escritura de los enunciados y ayudas para los tipos de ejercicio pueden generar confusión, la necesidad de expresiones más explícitas y la necesidad de material de referencia.

8. INTEGRACIÓN Y DESPLIEGUE

Para la integración del producto “Lvlututor” con la plataforma Learner Model GQL se utilizó la plantilla de desarrollo entregada por el proyecto. Esto significó mover los componentes desarrollados a un directorio llamado “lvlututor” donde adicionalmente se creó un conector para facilitar la importación. El código del conector se muestra en la Figura 64. Luego se procedió a verificar el funcionamiento del reporte de acciones mediante el cliente de administración de la plataforma.

```
1 |import dynamic from "next/dynamic";
2
3 |import type { ExType } from "../Tools/ExerciseType";
4
5 |const Lvlututor = dynamic({
6 |  () => {
7 |    return import("../Tools/Solver2");
8 |  },
9 |  { ssr: false },
10 |});
11
12 |export const Plain = ({ topicId, steps }: { topicId: string; steps: ExType }) => {
13 |  return (
14 |    <
15 |      {steps?.type == "lvlututor" ? (
16 |        <Lvlututor key={topicId} topicId={topicId} steps={steps} />
17 |      ) : (
18 |        "potato"
19 |      )}
20 |    </>
21 |  );
22 |};
23
24 |export default Plain;
```

Figura 64, Conector entregado para el producto “Lvlututor”.

Otros aspectos de integración incluyeron la corrección de código en *Javascript a TypeScript* y la separación de código no esencial a las funcionalidades de los componentes principales. La corrección de código significó crear un componente donde se definen las interfaces (estructura donde se definen los tipos de las variables) para la estructura JSON que almacena los ejercicios desarrollados.

8.1 Github

Para facilitar el proceso de integración de las diversas implementaciones de los tutores se creó una rama adicional en el servicio *Github* donde se almacena el control de versiones de la plataforma Learner Model GQL. Esto significó cambiar el origen del proyecto en la configuración local de *Git* a la nueva dirección y cambiar el proceso de desarrollo requiriendo realizar un llamado para actualizar el repositorio local antes de realizar las modificaciones. Adicionalmente se incorporó el uso de la tecnología “prettier” la cual tiene como objetivo “ordenar y limpiar visualmente” el código. Esto significa que el proceso de edición de código se resume en: 1)llamar al repositorio, 2)modificar el código, 3)aplicar *prettier*, 4) cometer los cambios en *git* y 5) enviar los cambios al origen configurado en *github*.

8.2 Despliegue de los ejercicios

La solución entregada es utilizada para desplegar los tipos de ejercicios diseñados en el capítulo 4. La Figura 65 muestra el despliegue del segundo tipo de ejercicio del contenido de fracciones en el segundo paso, mientras que la Figura 66 muestra el despliegue del ejercicio al ser completado. La encuesta mostrada al final de la Figura 66 corresponde a un componente agregado por el equipo de desarrollo. Debido a los resultados obtenidos en el laboratorio, el patrocinante realizó modificaciones al contenido de algunas instrucciones/ayudas diseñadas para ser más claras.

Para desplegar un ejercicio en la plataforma, se deberá seguir la estructura JSON mostrada con los contenidos definidos en la tabla 9 del capítulo 6 y asignar al atributo type el valor “lvltutor”.

Para seleccionar el tipo de evaluación, se debe definir en cada paso un atributo llamado “validation” con los valores: “evaluate”, “countElements”, “evaluateAndCount. Al no especificar este atributo, se realizará una comparación de cadenas de texto entre las respuestas.

Suma de fracciones

Suma las siguientes fracciones:

$$\frac{1}{b^2} + \frac{1}{a^2 \cdot b}$$

Encuentra el mínimo común múltiplo entre los denominadores y escríbelo

Suma las siguientes fracciones aplicando el mínimo común múltiplo encontrado.

$$\frac{1}{b^2} + \frac{1}{a^2 \cdot b}$$

(

)

x^y

\sqrt{x}

$\sqrt[y]{x}$

+

-

*

/

C

L

$\frac{a^2+b}{a^2b^2}$

R

Enviar

Pista 0

Figura 65, despliegue del desarrollo de un paso con Lvltutor.

Suma de fracciones

Suma las siguientes fracciones:

$$\frac{1}{b^2} + \frac{1}{a^2 \cdot b}$$

Encuentra el mínimo común múltiplo entre los denominadores y escríbelo

$$a^2 b^2$$

Has encontrado el mínimo común múltiplo de los denominadores.

Suma las siguientes fracciones aplicando el mínimo común múltiplo encontrado.

$$\frac{a^2 + b}{a^2 b^2}$$

Has realizado la suma de fracciones aplicando mínimo común múltiplo correctamente.



Resumen

Expresión: $\frac{1}{b^2} + \frac{1}{a^2 \cdot b}$

1) Se obtiene el mínimo común múltiplo entre dos denominadores.

$$a^2 \cdot b^2$$

2) Finalmente se realiza la suma de fracciones haciendo uso del mínimo común múltiplo encontrado.

$$\frac{a^2 + b}{a^2 \cdot b^2}$$

¿Cómo fue tu experiencia con este ejercicio que el sistema eligió para ti?



Siguiente Ejercicio

Figura 66, despliegue del completado de un ejercicio con Lvlutor.

Como limitación, se tiene la resolución lineal del orden de los pasos de un ejercicio. Esto significa que un usuario no podrá saltar pasos si ingresa una solución correspondiente a un paso mayor al subsiguiente. El motivo original de esta limitación se debe a que hay diferentes KC en los pasos y no es bueno asumir su dominio.

8.3 Reusabilidad y extensibilidad del componente de evaluación numérica

Es posible que en futuras mantenciones de la plataforma nazca el requerimiento de utilizar y/o extender la evaluación numérica en otros componentes. Como este requerimiento fue ajustado al componente MQ2, se procede a mostrar una alternativa para reutilizar el código o realizar modificaciones en las funcionalidades implementadas.

8.3.1 Integración en otros componentes

El traductor postfijo recibe como entrada una cadena de texto con escritura de expresiones en ascii/latex con orden infijo. Como salida entrega una cadena de texto donde se separan las variables/números/operadores con un espacio y en orden postfijo. La calculadora recibe como entrada una cadena de texto con el formato definido en la salida del traductor postfijo. La Figura 67 muestra el despliegue de un ejemplo de uso para el traductor y calculadora, mientras que la Figura 68 muestra el código del ejemplo mostrado.

Traduccion postfija: 3 1 * 2 /

Evaluacion numerica: 1.5

3*1/2

Figura 67, Ejemplo de uso del traductor y evaluación numérica.

```
1  import { Input, Text, VStack } from "@chakra-ui/react";
2  import { useRef, useState } from "react";
3  import DMQPostfixparser from "@/utils/DMQPostfixparser";
4  import MQPostfixSolver from "@/utils/MQPostfixSolver";
5
6  const Ejemplo1 = () => {
7    const [traduccion, setTraduccion] = useState("");
8    const [evaluacion, setEvaluacion] = useState("");
9    const text = useRef("");
10   const handleChange = (event :React.ChangeEvent<HTMLInputElement>) => {
11     text.current=DMQPostfixparser(event.target.value);
12     setTraduccion(text.current)
13     setEvaluacion(MQPostfixSolver(text.current,["{}"]))
14   }
15
16
17   return(
18     <VStack>
19       <Text>Traduccion postfija: {traduccion}</Text>
20       <Text>Evaluacion numerica: {evaluacion}</Text>
21       <Input placeholder='Pruebas' onChange={handleChange}/>
22     </VStack>
23   )
24 }
25 export default Ejemplo1;
```

Figura 68, Código del ejemplo de uso para uso del traductor y evaluación.

Limitaciones, por el momento la traducción y calculadora postfija solo tiene implementado las funciones y operadores mostrados en la Figura 69. Adicionalmente se tiene implementado el operador unitario “-” (ejemplo: 2^-1).

```
//here we define the functions, but it can be expanded to other cases
const reservedWords = {
  sqrt: "sqrt",
  sin: "sin",
  cos: "cos"
};

const operator = {
  "^": "^",
  "*": "*",
  "/": "/",
  "+": "+",
  "-": "-",
};
```

Figura 69, Listado de funciones y operadores implementados.

8.3.2 Incorporación de nuevas funciones u operadores al traductor y calculadora

Actualmente para agregar una nueva función o operador se debe modificar el componente llamado “MQPostfixparser.tsx”. Si se desea agregar una función, ésta se debe definir en la constante llamada “reservedWords” mostrada en la Figura 69. Para agregar un nuevo operador este se debe definir en la constante llamada “operator”, también mostrada en la Figura 69. También se debe agregar el orden de precedencia y tipo de asociatividad en las constantes llamadas “precedense” y “associativity”, ambas mostradas en la Figura 70. Como limitación se tiene el procesamiento de funciones de manera “ $f(\text{expresión})$ ” o “ $f(\text{expresión } A)(\text{expresión } B)$ ”. Si se define una función de comportamiento diferente, se deberá considerar su preprocesamiento para cambiar su forma.

```
const precedense = {
  sin: 17,
  cos: 17,
  sqrt: 17,
  "√": 17,
  "\\um": 14,
  "^": 13,
  "*": 12,
  "/": 12,
  "+": 11,
  "-": 11,
};

const associativity = {
  "^": "right",
  "*": "left",
  "/": "left",
  "+": "left",
  "-": "left",
};
```

Figura 70, constantes para definir precedencia y asociatividad.

9. CONCLUSIONES

Como promesa de este proyecto se tiene la creación de nuevos contenidos de fracciones y potencias remediales en álgebra para el tutor cognitivo desplegado en la plataforma Learner Model GQL. Originalmente la creación de nuevo contenido significaba realizar el esfuerzo del diseño de los tipos de ejercicios y el desarrollo del código asociado con la interfaz encargada de su despliegue. Con la entrega de "*Lvltutor*" se cuenta con una alternativa que permite solo enfocarse en el diseño de los tipos de ejercicios. Actualmente se cuenta con alrededor de 200 ejercicios para diferentes tópicos diseñados por practicantes que utilizan la solución entregada.

Para el proceso de diseño de los nuevos ejercicios se utilizó el modelo cognitivo definido por la teoría ACT-R. Esta teoría entrega una abstracción de un acercamiento educacional para la creación o fortalecimiento de nuevas habilidades mediante la distinción del conocimiento declarativo y procedural, además de entregar una estrategia que utiliza la definición de componentes del conocimiento (KC) para el modelado del estudiante. Sin embargo, la aplicación de esta teoría no fue suficiente. Esto se debe a una deficiencia reportada mediante la retroalimentación entregada por los participantes del laboratorio de evaluación donde indican la poca claridad de las instrucciones y ayudas entregadas absolviendo de responsabilidad la teoría aplicada. Esto da indicaciones que además del uso de teoría cognitiva es necesario utilizar otras estrategias comunicacionales que faciliten un mejor entendimiento de las instrucciones/ayudas.

Los requerimientos permitieron la detección temprana de la necesidad de desarrollar una evaluación de respuestas numéricas. Esta alternativa permite simplificar el trabajo necesario en el diseño de ejercicios al momento de considerar las respuestas aceptadas por el sistema. Un requerimiento de diseño con resultados positivos en la evaluación del laboratorio fue el uso de la botonera que permite facilitar el ingreso de operadores y funciones básicas.

El desarrollo de la evaluación numérica para el componente MQ2 significó desarrollar la traducción de notación infija con elementos del lenguaje latex a notación postfija para luego ser evaluados en una calculadora simple desarrollada. Los resultados de pruebas preliminares mostraron la necesidad adicional de agregar la operación multiplicación en las cadenas de entrada por la costumbre del usuario al no agregar el operador al escribir variables algebraicas. Finalmente se pudo detectar tempranamente la necesidad de implementar una comparación numérica basada en el cálculo del error relativo y luego agregar un elemento de comparación estricta al contar la cantidad de elementos.

La ejecución del laboratorio de evaluación desarrollado permitió comparar los tipos de ingreso en texto plano (ASCII) vs ingreso con el componente MQ2(componente que permite la escritura en *LaTeX* sin su conocimiento), además de la resolución secuencial de los ejercicios diseñados. Los resultados de evaluación mostraron ser deficientes al obtener un puntaje medio de 53.75 en la encuesta de usabilidad del sistema (SUS). El resultado SUS obtenido fue parcialmente explicado con la retroalimentación entregada por los estudiantes al comunicar la poca claridad de los enunciados y ayudas. Los resultados de cantidad de intentos utilizados en el copiado de expresiones, indican que los estudiantes cometen menos errores al utilizar el componente MQ2 y que los estudiante tienen dificultades al identificar

el uso de paréntesis en texto plano. Esto significa que si se decide utilizar un input en ASCII, puede que sea necesario agregar el uso de paréntesis como otra habilidad a ser modelada por el sistema. Utilizar la escritura con la entrada que utiliza ASCII o el componente MQ2 no tiene mayor variación de tiempo. Finalmente se debe considerar que el laboratorio solo obtuvo una participación de cuatro estudiantes, de los cuales solo se procesaron los datos de tres debido a la pérdida de un grupo de evaluación por problemas técnicos.

Como mejoras del proyecto se recomienda separar completamente la validación de respuestas del componente MQ2 y facilitar la incorporación de nuevas funciones/operadores en los componentes de evaluación numérica. Otra mejora para la solución “Lvltutor” es permitir la selección del tipo de componente a utilizar en cada paso para permitir diferentes tipos de resolución. Teniendo estas mejoras implementadas, se podría comenzar con la evaluación de un nuevo servicio que permita generar una interfaz flexible para la creación de nuevos ejercicios sin salir de la plataforma Learner Model GQL.

Para finalizar este capítulo se procederá a listar los logros obtenidos, 1) Desarrollo de marco teórico y estado del arte sobre sistemas de tutoría inteligentes, 2) Modelado de ejercicios (dos de fracciones y cuatro de potencias) en conjunto con expertos, 3) Diseño de una solución de software, 4) Desarrollo de producto en base a la solución diseñada, 5) Evaluación del producto, 6) Integración del producto con la plataforma Learner Model GQL.

10. REFERENCIAS

- AbuEloun, N. N., & Naser, S. S. A. (2017). Mathematics intelligent tutoring system. *International Journal of Advanced Scientific Research*, 2, 11-16.
- Adebayo S. (10 de octubre de 2023). Chakra UI. Chakra UI. Recuperado el 10 de octubre de 2023 <https://chakra-ui.com/>.
- Álvarez, P., Arriaza, J., Cisternas, R., Olmos, Y., Ríos, C., & C., T. (n.d.). *Matemática Básica para la Inserción a la Ingeniería* (Vol. 6).
- Brooke, John. (1995). SUS: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.
- Dix A., Finlay J., Abowd G, Beale R. (4 de octubre de 2022). *evaluation techniques*. hcibook. <https://www.hcibook.com/e3/chaps/ch9/resources/>.
- John F. Pane, B. A. G., Mcaffrey, D. F., & Karam, R. (2014). Effectiveness of Cognitive Tutor Algebra I at Scale. *Educational Evaluation and Policy Analysis*, 36, 127-144.
- John R. Anderson, A. T. C., Koedinger, K., & Pelletier, R. (1996). *Cognitive Tutors: Lessons Learned*. United States Army Research Institute for the Behavioral and Social Sciences.
- Kulik, J. A., & Fletcher, J. (2016). Effectiveness of Intelligent Tutoring Systems: A Meta-Analytic Review. *Review of Educational Research*, 86, 42-78.
- Linsley T (11 de octubre de 2023). Overview | TanStack Query Docs. TanStack. Recuperado el 11 de octubre de 2023 <https://tanstack.com/query/v3/>.
- Microsoft (10 de octubre de 2023). TypeScript: JavaScript With Syntax For Types. TypeScript. Recuperado el 10 de octubre de 2023 <https://www.typescriptlang.org/>.
- Nahuelpán, M. (2021). Tutor cognitivo y modelado de habilidades matemáticas para nivelación de estudiantes de primer año. Trabajo para optar al título de Ingeniero Civil en Informática.
- Naser, S. S. A. (2016). ITSB: An Intelligent Tutoring System Authoring Tool. *Journal of Scientific and Engineering Research*.
- Okta Inc. (10 de octubre de 2023). Auth0: Secure access for everyone. But not just anyone. Auth0. Recuperado el 10 de octubre de 2023 <https://auth0.com/>.
- Oyarzun, M. (2023). Tutor cognitivo para actividades de nivelación del álgebra. Trabajo para optar al título de Ingeniero Civil en Informática.
- Ritter, S., John R. Anderson, K. R. K., & 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.
- Sáez, P. (10 de octubre de 2023). Architecture - Learner Model GQL. Learner Model GQL. <https://docs.lm-uach.org/how-it-works/architecture>.
- Sales, A. C., & Pane, J. F. (2019). The role of mastery learning in an intelligent tutoring system: principal stratification on a latent variable. *The Annals of Applied Statistics*, 13, 420-443.
- TheGuild. (10 de octubre de 2023). Introduction – GraphQL Code Generator. GraphQL Code Generator. Recuperado el 10 de octubre de 2023 <https://the-guild.dev/graphql/codegen/docs/getting-started>.
- Valtio (11 de octubre de 2023). Getting Started — Valtio, makes proxy-state simple for React and Vanilla. Valtio. Recuperado el 11 de octubre de 2023 <https://valtio.pmnd.rs/docs/introduction/getting-started>.

Vercel, Inc. (10 de octubre de 2023). What is Next.js? | Learn Next.js. Next.js. Recuperado el 10 de octubre de 2023 <https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs>.