



Universidad Austral de Chile

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

FORTALECIMIENTO DE TUTOR COGNITIVO MEDIANTE EL DESARROLLO DE EJERCICIOS INTERACTIVOS CON ENFOQUE SOCIO- MEDIOAMBIENTAL PARA UN APRENDIZAJE INTEGRAL

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:

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

PROFESOR CO-PATROCINANTE:

DANIEL EDUARDO SÁNCHEZ IBÁÑEZ
INGENIERO ACÚSTICO
MAGÍSTER EN INGENIERÍA MECÁNICA
MAGÍSTER EN EDUCACIÓN MATEMÁTICA
DOCTOR EN MATEMÁTICA APLICADA

PROFESOR INFORMANTE:

ELIANA SCHEIHING GARCÍA
INGENIERO CIVIL MATEMÁTICO
D.E.A. EN MATHEMATIQUE
DOCTEUR EN STATISTIQUE

ENZO JAVIER MARÍN ROSAS
VALDIVIA – CHILE
2023

AGRADECIMIENTOS

ÍNDICE

ÍNDICE	II
ÍNDICE DE TABLAS	III
ÍNDICE DE FIGURAS	IV
RESUMEN.....	VII
ABSTRACT	VIII
1. INTRODUCCIÓN	1
1.1 Definición del problema y oportunidad	1
1.2 Objetivos	3
2. MARCO TEÓRICO.....	4
2.1 Revisión sistemática.....	5
2.2 Desarrollo de la revisión sistemática.....	5
2.3 Principales hallazgos	11
2.4 Conclusiones de la revisión sistemática.....	13
3. METODOLOGÍA DE DESARROLLO	14
3.1 Scrum	14
3.2 Adaptación de metodología	14
3.3 Tecnologías de desarrollo	15
4. DISEÑO Y MODELADO DEL CONTENIDO	16
4.1 Diseño de ejercicios	16
4.2 Definición de componentes.....	21
4.3 Análisis cognitivo y diseño de resolución de ejercicios	21
4.4 Diseño de interfaz de los ejercicios.....	44
4.5 Modelamiento de ejercicios	48
5. IMPLEMENTACIÓN.....	66
5.1 Sprint 1	66
5.2 Sprint 2	79
5.3 Sprint 3	88
5.4 Sprint 4	98
5.5 Sprint 5	105
5.6 Sprint 6.....	110
6. VALIDACIÓN.....	114
6.1 Planificación.....	114
6.2 Preparación del sistema.....	116
6.3 Resultados	118
7. CONCLUSIONES Y TRABAJO FUTURO	128
7.1 Trabajo futuro	128
8. REFERENCIAS.....	130
9. ANEXOS	132
ANEXO A: Tablas de artículos seleccionados.	132
ANEXO B: Encuesta por ejercicio	146
ANEXO C: Encuesta SUS	147
ANEXO D: Encuesta satisfacción	149
ANEXO E: Resultados de encuestas	150

ÍNDICE DE TABLAS

Tabla	Página
Tabla 1: Artículos seleccionados	8
Tabla 2: Ejemplo de instrumento para la recopilación de información	9
Tabla 3: Resultados de la extracción de datos del primer artículo	10
Tabla 4: Identificación de tópicos	16
Tabla 5: (Continuación)	17
Tabla 6: Descripción de ejercicios seleccionados	19
Tabla 7: (Continuación)	20
Tabla 8: KCs del Ejercicio 1	49
Tabla 9: KCs del Ejercicio 2	50
Tabla 10: KCs del Ejercicio 3	50
Tabla 11: (Continuación)	51
Tabla 12: KCs del Ejercicio 4	51
Tabla 13: KCs del Ejercicio 5	52
Tabla 14: Definición de atributos en la estructura general de los ejercicios	53
Tabla 15: Definición de los atributos para el campo <i>presentation</i>	54
Tabla 16: Definición de los atributos para el campo <i>learningObjetives</i>	54
Tabla 17: Definición de los atributos para el campo "table"	55
Tabla 18: Definición de atributos para la estructura de las preguntas	56
Tabla 19: Definición de los atributos de la estructura para los pasos de una pregunta....	57
Tabla 20: Definición del campo <i>meta</i> para el componente <i>mathComponent</i>	59
Tabla 21: Definición del campo <i>meta</i> para el componente <i>selectionComponent</i>	60
Tabla 22: Definición del campo <i>meta</i> para el componente <i>graphComponent</i>	61
Tabla 23: Definición del campo <i>metaComponent</i> para el componente <i>selectPoint</i>	61
Tabla 24: Definición del campo <i>metaComponent</i> para el componente <i>linearFit</i>	61
Tabla 25: Verbos para el reporte de acciones	110
Tabla 26: Planificación de sesiones de evaluación	114
Tabla 27: Asignación de ejercicios mediante la técnica <i>latin square design</i>	115
Tabla 28: Promedio de la pregunta 1 por grupo y ejercicio	119
Tabla 29: Promedio de la pregunta 2 por grupo y ejercicio	120
Tabla 30: Promedio de la pregunta 3 por grupo y ejercicio	120
Tabla 31: Promedio de la pregunta 4 por grupo y ejercicio	121
Tabla 32: Promedio por grupo de la pregunta 3 del test de satisfacción.....	126

ÍNDICE DE FIGURAS

Figura	Página
Figura 1. Ejemplo de ejercicio del tópico de factorización en la plataforma de TC.....	2
Figura 2. Agrupación de palabras claves para búsqueda bibliográfica.....	6
Figura 3. Número de artículos seleccionados por año.....	8
Figura 4. Estructura de los ejercicios	17
Figura 5: Diseño de pregunta 1 del Ejercicio 1.....	22
Figura 6: Diseño de pregunta 2 del Ejercicio 1.....	23
Figura 7: Diseño de pregunta 3 del Ejercicio 1.....	24
Figura 8: Diseño de pregunta 4 del Ejercicio 1.....	25
Figura 9: Diseño de pregunta 5 del Ejercicio 1.....	25
Figura 10: Diseño de pregunta 7 del Ejercicio 1	26
Figura 11: Diseño de pregunta 1 del Ejercicio 2.....	27
Figura 12: Diseño de pregunta 2 del Ejercicio 2.....	27
Figura 13: Diseño de pregunta 3 del Ejercicio 2.....	28
Figura 14: Diseño de pregunta 4 del Ejercicio 2.....	29
Figura 15: Diseño de pregunta 1 del Ejercicio 3.....	30
Figura 16: Diseño de paso 1 y 2 de la pregunta 2 del Ejercicio 3.....	31
Figura 17: Diseño del paso 3 de la pregunta 2 del Ejercicio 3.....	32
Figura 18: Diseño del paso 4 de la pregunta 2 del Ejercicio 3.....	33
Figura 19: Diseño del paso 5 de la pregunta 2 del Ejercicio 3.....	34
Figura 20: Diseño de los pasos 1 y 2 de la pregunta 3.....	35
Figura 21: Diseño de pregunta 4 del Ejercicio 3.....	36
Figura 22: Diseño de pregunta 1 del Ejercicio 4.....	37
Figura 23: Diseño de paso 1 de la pregunta 2 del Ejercicio 4.....	38
Figura 24: Diseño de pasos 2 y 3 de la pregunta 2 del Ejercicio 4.....	39
Figura 25: Diseño de pregunta 3 del Ejercicio 4.....	40
Figura 26: Diseño de pregunta 1 del Ejercicio 5.....	40
Figura 27: Diseño de paso 1 y 2 de la pregunta 2 del Ejercicio 5.....	41
Figura 28: Diseño de los pasos 3 al 6 de la pregunta 2 del Ejercicio 5.....	42
Figura 29: Definición de pasos del 7 al 9 de la pregunta 2 del Ejercicio 5.....	43
Figura 30: <i>Mockup</i> general del diseño de los ejercicios	45
Figura 31: <i>Mockup</i> de la selección de alternativa	45
Figura 32: <i>Mockup</i> de la selección de alternativa con respuesta errónea.	46
Figura 33: <i>Mockup</i> de la selección de alternativa con respuesta correcta.....	46
Figura 34: <i>Mockup</i> componente matemático con respuesta vacía	47
Figura 35: Mockup del despliegue de un <i>hint</i>	47
Figura 36: <i>Mockup</i> del componente gráfico.....	48
Figura 37: JSON de la estructura general de los ejercicios.....	55
Figura 38: Estructura JSON de la tabla del Ejercicio 1.	56
Figura 39: Estructura JSON del arreglo de preguntas de un ejercicio	58
Figura 40: Estructura JSON del componente " <i>mathComponent</i> ".	60
Figura 41: Estructura JSON del componente " <i>selectionComponent</i> "	60

Figura 42: Ejemplo de estructura JSON del componente <i>selectPoint</i>	62
Figura 43: Ejemplo de estructura JSON del componente <i>linearFit</i>	64
Figura 44: Estructura JSON del atributo <i>graphSettings</i>	65
Figura 45: Estructura de archivos del proyecto de tutores cognitivos	66
Figura 46: Estructura de archivos del proyecto.....	67
Figura 47: Estructura de archivos de los componentes.....	68
Figura 48: Pieza de software correspondiente al Ejercicio 1	69
Figura 49: Despliegue de los pasos de las preguntas	70
Figura 50: Componente de selección.	70
Figura 51: Despliegue de alerta al ingresar respuesta incorrecta.	71
Figura 52: Despliegue de alerta al ingresar respuesta correcta.	71
Figura 53: Código componente <i>ResAlert</i>	72
Figura 54: Código del <i>custom hook useAlert</i>	73
Figura 55: Código del componente de selección.	75
Figura 56: Definición de <i>interface SelectionMeta</i>	76
Figura 57: Código para importar un componente	76
Figura 58: Código para el despliegue de pasos	77
Figura 59: Definición del <i>enum "components"</i>	78
Figura 60: Pieza de software correspondiente a la pregunta 3 del Ejercicio 1	80
Figura 61: Campo " <i>componentToAnswer</i> " de la pregunta 3 del Ejercicio 1.....	81
Figura 62: Código de la implementación del componente <i>Mathfield</i>	82
Figura 63: Código para importar componente <i>Mathfield</i>	83
Figura 64: código del componente <i>MathComponent</i>	84
Figura 65: código de la función <i>checkAnswer</i>	85
Figura 66: Retroalimentación visual por campo del componente <i>MathComponent</i>	86
Figura 67: Código de la función <i>handleMathFieldChange</i>	87
Figura 68: Ejemplo de uso del componente <i>mathComponent</i>	87
Figura 69: Interfaz del componente <i>selectPoint</i>	89
Figura 70: Interfaz del componente <i>linearFit</i>	90
Figura 71: <i>Interfaces</i> para la configuración del gráfico.....	91
Figura 72: <i>Interfaces</i> para el componente <i>SelectPoint</i>	92
Figura 73: Código del componente <i>selectPoint</i>	94
Figura 74: Definición de componente <i>LinearFit</i>	95
Figura 75: Interfaz <i>linearFitMeta</i>	95
Figura 76: Código para el despliegue de los datos del componente <i>LinearFit</i>	96
Figura 77: Código para la creación de los <i>sliders</i> del componente <i>LinearFit</i>	96
Figura 78: Código para la definición y despliegue de la función lineal.....	97
Figura 79: Visualización de <i>Hint</i> inicial.	99
Figura 80: Notificación al desbloquear nuevo <i>hint</i>	99
Figura 81: Visualización del último <i>hint</i>	100
Figura 82: Código del custom hook <i>useHint</i>	102
Figura 83: Definición de componente <i>HintButton</i>	103
Figura 84: Retorno del componente <i>HintButton</i>	104
Figura 85: Código del <i>hook useStore</i> para el manejo de los estados globales.....	106
Figura 86: Inicialización de estados para el control de las preguntas y pasos.	107
Figura 87: Ejemplo de uso del atributo " <i>isBlocked</i> ".	108

Figura 88: Ejemplo de uso de la función " <i>unlockNextStep</i> ".	109
Figura 89: Importación y definición del <i>hook useAcion</i>	111
Figura 90: Ejemplo de reporte de acción.	112
Figura 91: Reporte de acciones en el sistema.	112
Figura 92: Página de inicio para las sesiones de evaluación.....	116
Figura 93: Pagina de instrucciones para comenzar sesiones de evaluación.....	117
Figura 94: Presentación de ejercicio para la sesión de evaluación.	118
Figura 95: Resultados promedio de la pregunta 1 por ejercicio.....	119
Figura 96: Resultados promedio de la pregunta 2 por ejercicio.....	120
Figura 97: Resultados promedio de la pregunta 3 por ejercicio.....	121
Figura 98: Resultados promedio de la pregunta 4 por ejercicio.....	122
Figura 99: Nube de etiquetas de aspectos positivos.....	123
Figura 100: Nube de etiquetas de aspectos negativos.....	124

RESUMEN

El presente proyecto de título tiene como propósito fortalecer la plataforma de Tutores Cognitivos, la cual tiene como objetivo brindar apoyo en el aprendizaje de matemáticas a estudiantes de primer año de ingeniería. Lo anterior, se llevará a cabo mediante la incorporación de nuevos ejercicios interactivos con un enfoque socio-medioambiental. Estos ejercicios buscan mejorar la experiencia de aprendizaje de los estudiantes universitarios de primer año en carreras de ingeniería al proporcionar problemas matemáticos contextualizados que integren diversos temas.

Para lograr este propósito, inicialmente se realizó una revisión sistemática de la literatura científica sobre tutores cognitivos de matemáticas que implementen ejercicios contextualizados. Los principales hallazgos indican que estos sistemas utilizan reglas para modelar el proceso de resolución y el método cartesiano como base para estructurar los pasos de desarrollo de los ejercicios.

En términos de metodología de desarrollo, se empleó una versión adaptada de SCRUM. En esta adaptación, el rol de *Product Owner* fue desempeñado por el profesor patrocinante, los roles de *Scrum Master* y *Scrum Team* por el estudiante tesista, mientras que el rol de *Stakeholder* fue ocupado por el profesor co-patrocinante. Esta dinámica involucró reuniones semanales con los profesores patrocinante y co-patrocinante, durante las cuales se definieron los requerimientos y se llevaron a cabo las validaciones necesarias.

Durante dichas reuniones se diseñaron y modelaron cinco ejercicios con enunciados y aplicaciones del mundo real. Se definieron los pasos de resolución y los componentes interactivos requeridos, incluyendo un componente de selección de respuesta, un componente matemático y componentes gráficos.

Posteriormente, se procedió a la fase de implementación de estos ejercicios en la plataforma. Esto involucró la implementación de los componentes interactivos y la implementación e integración de los nuevos ejercicios dentro del sistema de Tutores Cognitivos.

Finalmente, se realizó una validación con estudiantes de primer año para evaluar la usabilidad del sistema y la satisfacción con los ejercicios. Los resultados de la evaluación indican una alta aceptación y una percepción positiva de los ejercicios contextualizados, lo que respalda su utilidad en el proceso de aprendizaje de matemáticas.

ABSTRACT

The purpose of this degree project is to strengthen the Cognitive Tutors platform, which aims to provide support in the learning of mathematics to first year engineering students. This will be done by incorporating new interactive exercises with a socio-environmental approach. These exercises seek to improve the learning experience of first-year university students in engineering careers by providing contextualized mathematical problems that integrate diverse topics.

To achieve this purpose, we initially conducted a systematic review of the scientific literature on cognitive mathematics tutors that implement contextualized exercises. The main findings indicate that these systems use rules to model the resolution process and the Cartesian method as a basis for structuring the development steps of the exercises.

In terms of development methodology, an adapted version of SCRUM was used. In this adaptation, the role of Product Owner was played by the sponsoring professor, the roles of Scrum Master and Scrum Team by the thesis student, while the role of Stakeholder was occupied by the co-sponsoring professor. This dynamic involved weekly meetings with the sponsoring and co-sponsoring professors, during which the requirements were defined and the necessary validations were carried out.

During these meetings, five exercises with real-world statements and applications were designed and modeled. The resolution steps and the required interactive components were defined, including an answer selection component, a mathematical component and graphical components.

Subsequently, we proceeded to the implementation phase of these exercises in the platform. This involved the implementation of the interactive components and the implementation and integration of the new exercises within the Cognitive Tutors system.

Finally, a validation was conducted with first year students to evaluate the usability of the system and satisfaction with the exercises. The evaluation results indicate a high acceptance and positive perception of the contextualized exercises, which supports their usefulness in the mathematics learning process.

1. INTRODUCCIÓN

Cada año ingresan a la Facultad de Ciencias de la Ingeniería (FCI), de la Universidad Austral de Chile (UACh), una gran cantidad de estudiantes con diferentes niveles educativos adquiridos al finalizar la educación media. Este desnivel ha provocado que una gran parte de los estudiantes tengan dificultades en el aprendizaje de los tópicos impartidos en las primeras asignaturas de matemática del ciclo de bachillerato en ingeniería. Para solventar esta problemática, la FCI ha puesto a disposición de sus nuevos estudiantes algunas estrategias de nivelación, antes del inicio del primer semestre, en las cuales se les proporcionan clases presenciales y/o material para que los estudiantes puedan prepararse para su inminente inicio de clases. Sin embargo, el poco tiempo que disponen para las nivelaciones y la gran cantidad de contenidos a cubrir, hacen que difícilmente se logre disminuir la brecha de los diversos niveles educativos adquiridos desde la educación media entre los nuevos estudiantes.

Con el objetivo de abordar esta problemática y apoyar en la etapa de nivelación a los nuevos estudiantes de la facultad, se ha desarrollado el proyecto de Tutores Cognitivos (TC), en el marco del proyecto fondecyt titulado “*Motivational Design of Cognitive Tutors for Supporting Math Learning of First-Year Undergraduate Engineering Students*”. En éste, se desarrolló una plataforma de sistemas educativos junto a una serie de implementaciones del tutor inteligente específico para los tópicos matemáticos de factorización, potencias, ecuaciones (lineales y cuadráticas), entre otros. Los ejercicios implementados en la plataforma se corresponden con los contenidos del Libro de Nivelación del Centro de Docencia de Ciencias Básicas para ingeniería (Álvarez et al., s.f.). No obstante, la mayoría de estos ejercicios se perciben abstractos y descontextualizados, siendo posiblemente un factor relevante en la desmotivación de los estudiantes por resolver ejercicios matemáticos que, consecuentemente, dificulta el razonamiento matemático y la comprensión de éstos por parte de los estudiantes.

Es por lo anterior que este trabajo busca aportar a la plataforma de Tutores Cognitivos con una nueva implementación que dotará a la plataforma de nuevos ejercicios interactivos y de aplicación contextualizados, que abordarán diversos tópicos matemáticos y que permitirán a los estudiantes adquirir y/o reforzar conocimientos, al mismo tiempo que los relacionan con contextos socio y medioambientales reales.

1.1 Definición del problema y oportunidad

Debido a la potencial desigualdad de la educación recibida con la que los estudiantes entran a la FCI, muchos estudiantes han tenido dificultades al resolver ejercicios matemáticos que se les presentan en las primeras asignaturas de sus carreras, ya sea en la comprensión de los conceptos como en la aplicación de los mismos. Por esta razón, parece ser importante que los aprendizajes de matemáticas se puedan vincular con la realidad, de manera que los estudiantes puedan mejorar su razonamiento matemático, la comprensión de los contenidos y su capacidad para aplicarlos en un contexto real.

En este aspecto, la plataforma de Tutores Cognitivos cuenta, actualmente, con una amplia variedad de ejercicios que abarcan diversos tópicos matemáticos como factorización, ecuaciones, potencias, geometría y fracciones. Estos ejercicios se presentan en forma de expresiones algebraicas y, dependiendo del caso, se solicita al estudiante que realice una acción específica, como por ejemplo resolver una ecuación, factorizar una expresión dada o convertir un número a notación científica, entre otras. Además, estos ejercicios están estructurados en una serie de pasos secuenciales de resolución, los cuales el estudiante debe completar correctamente para poder avanzar al siguiente paso.

A pesar de que los ejercicios actualmente disponibles en la plataforma de Tutores Cognitivos son útiles para reforzar los conocimientos básicos de matemáticas, éstos presentan algunas limitaciones. En primer lugar, están diseñados, en general, para cada tópico matemático de manera individual, lo que significa que no cuentan con ejercicios que integren a más de un tópico. Además, estos ejercicios son explícitos, con un alto nivel de abstracción, es decir, no se presentan dentro de un contexto ni enunciado que contribuya para que el aprendizaje sea, potencialmente, más significativo (ver Figura 1). Por último, la interactividad en la implementación de estos ejercicios es limitada, ya que, salvo en la implementación del tutor de ecuaciones, en la que el estudiante responde arrastrando el componente con la respuesta correcta, las otras implementaciones solo ofrecen un cuadro de texto para que el estudiante ingrese su respuesta.

Factorización / Factorización por factor común compuesto

Resolver el siguiente ejercicio de factorización por factor común compuesto

$$abcx - dx + abcy - dy$$

Agrupar los sumandos

Aceptar

Ayuda 0

Ingresar factor común a cada grupo

Ingresar factor común de toda la expresión

Figura 1. Ejemplo de ejercicio del tópico de factorización en la plataforma de TC.

Así, parece ser necesario de una nueva implementación de ejercicios que integren y contextualicen los aprendizajes de matemáticas de nivelación, de manera que los estudiantes puedan comprender, aplicar y relacionar el aprendizaje con problemas de la vida real, favoreciendo de esta manera el desarrollo de sus habilidades y el pensamiento crítico. Entonces, para este trabajo, esta implementación contemplará nuevos componentes interactivos, de manera que los estudiantes puedan interactuar de manera más dinámica con los ejercicios y el tutor cognitivo.

En cuanto a la elección del enfoque de los nuevos ejercicios, este será el socio y medioambiental, ya que éste está relacionado con las competencias que declara la nuestra universidad, relacionadas con la concientización ecológica y medioambiental, lo cual incentiva a orientar el material en este tema y a la vez nos da la oportunidad de incorporar a la plataforma el distintivo sello UACh: “conocimiento y naturaleza”.

1.1.1 Impacto del proyecto

Este proyecto busca potenciar y mejorar la plataforma de Tutores Cognitivos, para que los estudiantes tengan una experiencia de aprendizaje más significativa, que les permita comprender y aplicar los conceptos de matemáticas de una manera más fluida y efectiva. Además, se espera poder fomentar el interés y la motivación de los estudiantes por la resolución de ejercicios matemáticos y sus potenciales aplicaciones.

Por último, se espera poder contribuir al objetivo general del proyecto de Tutores Cognitivos, que es la disminución en la tasa de reprobación de estudiantes en las asignaturas matemáticas iniciales de ciencias básicas, como álgebra y geometría, y mejorar la retención de los estudiantes en las carreras de ingeniería.

1.2 Objetivos

1.1.2 Objetivo general

Fortalecer la plataforma de Tutores Cognitivos mediante el modelamiento e implementación de ejercicios interactivos con enfoque socio-medioambiental, en beneficio a potenciar el razonamiento matemático de estudiantes de primer año.

1.1.3 Objetivos Específicos

1. Determinar las características de los Tutores Cognitivos de matemática existentes que contengan material contextualizado por medio de una revisión sistemática.
2. Elaborar un conjunto de ejercicios mediante la recolección y adaptación de contenidos que aborden diversos tópicos de nivelación matemática.
3. Modelar adecuadamente los ejercicios para la plataforma de Tutores Cognitivos.
4. Implementar los ejercicios, ya recolectados y modelados, en la plataforma de Tutores Cognitivos.
5. Validar la propuesta mediante la evaluación subjetiva de los ejercicios, con estudiantes de primer año.

2. MARCO TEÓRICO

Los sistemas de tutoriales inteligentes (ITS, su sigla en inglés) son sistemas computacionales que buscan brindar instrucción y apoyo inteligente en el proceso de enseñanza mediante la interacción entre el sistema y el estudiante (Arias, Jiménez, & Ovalle, 2009). Entre los diferentes tipos de ITS, se encuentra una rama específica conocida como “tutores cognitivos”. Estos se basan en la teoría cognitiva y el modelo cognitivo para apoyar el aprendizaje mediante la práctica de resolución de problemas y la provisión de una guía de resolución paso a paso (Aleven, McLaren, & Sewall, 2009).

Los tutores cognitivos utilizan un modelo cognitivo basado en reglas de resolución de problemas por parte del estudiante, lo que permite al sistema realizar un seguimiento del desarrollo de las habilidades del estudiante a lo largo del tiempo y seleccionar problemas de forma individualizada (Aleven et al., 2009). Estos sistemas han demostrado ser efectivos en la mejora del rendimiento académico en diferentes áreas, como las matemáticas y las ciencias, acercándose a la efectividad de la tutoría humana (VanLEHN, 2011).

Existen diversos contextos sociales y medioambientales en los que se pueden aplicar las matemáticas. Por ejemplo, en el contexto social, las matemáticas pueden ayudar a analizar y comprender estadísticas relacionadas con el bienestar de la población, como las tasas de desempleo o el índice de pobreza. En el contexto medioambiental, las matemáticas pueden ayudar a analizar datos relacionados con el cambio climático, como la temperatura media global o el nivel medio del mar (Catalano, 2010). Estos tipos de contextos, son de gran utilidad para que los estudiantes puedan relacionar sus aprendizajes con aspectos reales, aplicar sus habilidades y activar su interés por la matemática.

Este tipo de problemas matemáticos, que presentan un enunciado y contexto, también conocidos en inglés como “*word problems*”, son un tipo de problemas común en la enseñanza de matemáticas, en donde se plantea un ejercicio matemático mediante un texto o enunciado. Este tipo de problemas resultan ser valiosos en la investigación psicológica sobre el razonamiento complejo, porque implican tanto demandas de procesamiento del lenguaje como de razonamiento matemático (Walkington, Clinton, Ritter, & Nathan, 2015)

Del Olmo, González, Diago, Arnau, & Arevalillo, (2023) mencionan que la implementación de Tutores Cognitivos destinados a enseñar y aprender la resolución de los “*word problems*” deben cumplir dos requisitos: (1) Supervisar la validez de las respuestas del estudiante y (2) recopilar la información sobre el desempeño del estudiante. El primer requisito debe permitir la identificación de éxitos y errores y proporcionar ayuda durante la resolución del problema. El segundo requisito permite al Tutor Cognitivo modelar el conocimiento del estudiante sobre la resolución de problemas y usar este conocimiento como base para las decisiones educativas que el sistema debe tomar.

Además, Del Olmo et al. (2023) mencionan que la necesidad de monitorear el proceso de resolución requiere el diseño de sistemas con granularidad fina (alto nivel de detalle), que, al mismo tiempo, sean flexibles a las decisiones tomadas por el alumno.

En particular, la resolución de “*word problems*” algebraicos, en donde puede existir la posibilidad de que el mismo problema se pueda resolver a través de diferentes rutas de solución, va a requerir proporcionar al Tutor Cognitivo la capacidad de determinar la validez de un paso e identificar la ruta de solución entre muchas posibles, para articular sugerencias basado en el estado del proceso de resolución (Del Olmo et al., 2023).

2.1 Revisión sistemática

Con el fin de conocer el estado del arte de los tutores cognitivos que implementen ejercicios textuales (“*word problems*”) y contextualizados para el aprendizaje de matemáticas, se ha llevado a cabo una revisión sistemática de la literatura científica disponible.

2.2 Desarrollo de la revisión sistemática

2.1.1 Objetivo y pregunta de investigación

El objetivo de esta revisión sistemática es buscar, encontrar y analizar literatura relacionada con tutores cognitivos de matemáticas en los que se hayan modelado e implementado ejercicios textuales y contextualizados, en inglés llamados “*word problems*”. De esta manera se plantearon las siguientes preguntas de investigación:

1. ¿Cuáles son las características principales de los tutores cognitivos de aprendizaje de matemáticas que implementen “*word problems*” o ejercicios contextualizados?
2. ¿Cómo se modelan o implementan los ejercicios contextualizados o “*word problems*” en los tutores cognitivos de matemática?

2.2.2 Metodología de la revisión

Como metodología para el desarrollo de esta revisión sistemática se emplearon tres etapas. En primer lugar, se hizo una definición y refinamiento de la cadena de búsqueda en base a las preguntas de investigación. En segundo lugar, se realizó la búsqueda en donde se obtuvieron un total de 70 artículos científicos de los cuales se escogieron los artículos más relevantes, de acuerdo a los criterios de selección. Por último, se realizó el análisis de la información y extracción de datos, con lo cual se elaboró un resumen de los resultados obtenidos. A continuación, se detalla el proceso de la revisión sistemática.

2.2.3 Fuentes de revisión

Las fuentes bibliográficas utilizadas para realizar la búsqueda fueron la librería digital de Web of Science¹ (WoS) y SCOPUS².

En cuanto a la estrategia de selección se consideró solo búsquedas en el idioma inglés, para obtener una mayor cantidad de resultados. Además, la selección se centró únicamente en la revisión de los resúmenes (*abstract*) de los artículos y solo en los publicados desde el año 2000 en adelante, es decir, se excluyeron los artículos con más de 22 años de antigüedad.

2.2.4 Palabras claves

Relacionadas con este trabajo de investigación, se seleccionaron las siguientes palabras claves:

tutoring systems, intelligent tutoring, Smart tutor, Digital tutor, Cognitive tutor, math, mathematics, pre-algebra, algebra, geometry, problems, exercises, content, material, word, contextualized, story, real world, real case, real life, textual, contextual, motion.

Una vez definidas las palabras claves se procedió a agruparlas, por su similaridad de sus conceptos, en los siguientes grupos (ver Figura 2):

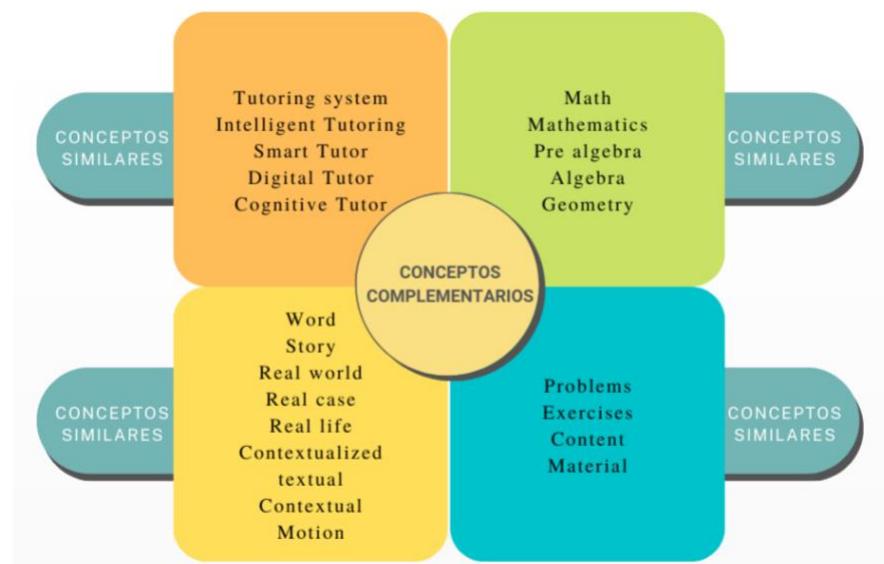


Figura 2. Agrupación de palabras claves para búsqueda bibliográfica.

¹ <https://www.webofscience.com/wos/woscc/basic-search>

² <http://www.scopus.com/>

2.2.5 Cadena de búsqueda

La cadena de búsqueda finalmente quedó definida de la siguiente manera:

(“*tutoring systems*” OR “*intelligent tutoring*” OR “*Smart tutor*” OR “*Digital tutor*” OR “*Cognitive tutor*”)

AND

(*math* OR *mathematics* OR “*pre-algebra*” OR *algebra* OR *geometry*)

AND

(*problems* OR *exercises* OR *content* OR *material*)

AND

(*word* OR *contextualized* OR *story* OR “*real world*” OR “*real case*” OR “*real life*” OR *textual* OR *contextual* OR *motion*)

2.2.6 Criterios de selección

Una vez realizada la búsqueda, se seleccionaron los artículos que tienen relación con el modelado o implementación de ejercicios textuales o “*word problems*” en un Tutor inteligente, así como también artículos que mencionan o indiquen cómo estos entregan *feedback* a los estudiantes. Específicamente los criterios de selección de artículos científicos fueron los siguientes:

- Donde se describan o indiquen detalles de la implementación de los “*word problems*”.
- Que describan tutores cognitivos con “*word problems*” y donde se modelen estos ejercicios.
- Que describan competencias o habilidades asociadas a los “*word problems*”.
- Incluyan detalles de cómo abordar el espacio de soluciones posibles.
- Incluyan detalles del *feedback* que se entrega al estudiante.
- Incluyan temas de la comprensión lectora al resolver problemas.

2.2.7 Extracción de información

Al inicio del proceso de búsqueda se obtuvieron un total de 70 artículos, 38 desde WoS y 32 desde SCOPUS, de los cuales se hizo un proceso de selección en donde se leyeron detenidamente los “*abstract*” de cada artículo y se fueron escogiendo según los criterios previamente listados. De esta selección específica se obtuvieron 14 artículos, los cuales fueron leídos por completo y de los cuales solo 10 artículos fueron finalmente seleccionados para este trabajo (ver Tabla 1 y Figura 3).

Tabla 1: Artículos seleccionados.

Año	Título Paper
2000	<i>Analyzing and generating mathematical models: An Algebra II Cognitive Tutor design study</i>
2002	<i>An intelligent tutoring system incorporating a model of an experienced human tutor</i>
2007	<i>LIM-G: Learner-initiating instruction model based on cognitive knowledge for geometry word problem comprehension</i>
2013	<i>Domain-specific knowledge representation and inference engine for an intelligent tutoring system</i>
2013	<i>Fundamentals of the design and the operation of an intelligent tutoring system for the learning of the arithmetical and algebraic way of solving word problems</i>
2015	<i>How Readability and Topic Incidence Relate to Performance on Mathematics Story Problems in Computer-Based Curricula</i>
2015	<i>Intensive scaffolding in an intelligent tutoring system for the learning of algebraic word problem solving</i>
2017	<i>Rule-based Cognitive Modeling and Model Tracing for Symbolization in a Math Story Problem Tutor</i>
2018	<i>Constraint-based Student Modeling in Probability Story Problems with Scaffolding Techniques</i>
2021	<i>Systematic literature review of STEM self-study related ITSs</i>

Cabe destacar que, la gran mayoría de los artículos descartados, si bien abordaban Tutores cognitivos con “*word problems*”, estos se centraban en la evaluación del desempeño del tutor y la efectividad de los mismos en el aprendizaje de los estudiantes, pero no detallaban cómo estos tipos de ejercicios eran modelados y/o implementados en un ITS.

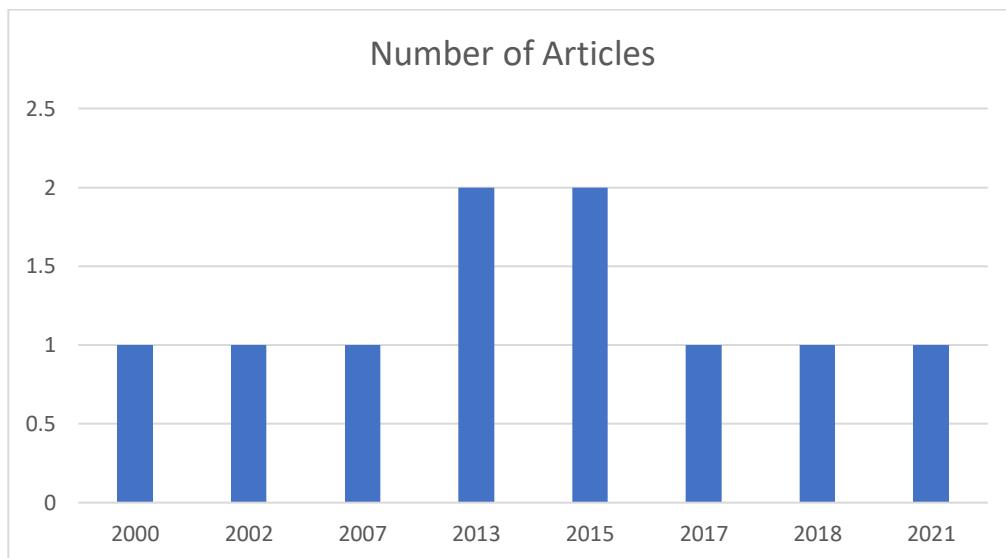


Figura 3. Número de artículos seleccionados por año.

2.2.8 Resultados

Para recopilar la información de cada artículo seleccionado, se creó una tabla de relleno/extracción de datos que sigue el formato presentado en la Tabla 2.

Tabla 2: Ejemplo de instrumento para la recopilación de información

Código del Artículo
Nombre del artículo
¿Cuál es el contexto del artículo?
¿Qué datos o detalles útiles nos brinda este artículo?
Describir los elementos o características claves del modelamiento de los “word problems”
Describir como implementan los “word problems” en un tutor cognitivo
¿El artículo describe competencias y/o habilidades asociadas a la solución de los “word problems”?
¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. para ayudar a la resolución del problema?
¿El artículo describe los tipos de feedback entregados a los estudiantes?
¿Cómo el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?
Otros hallazgos relevantes

De esta manera se fue completando la Tabla 2 con cada artículo seleccionado. A continuación se muestra un ejemplo, del completado de la tabla, para el primer artículo seleccionado:

Tabla 3: Resultados de la extracción de datos del primer artículo

Código del Artículo
[A01]
Nombre del artículo
Rule-based Cognitive Modeling and Model Tracing for Symbolization in a Math Story Problem Tutor.
¿Cuál es el contexto del artículo?
El contexto de este artículo gira en torno a MAST, el cual es un Sistema de tutor inteligente para problemas verbales (word problems) de probabilidad.
¿Qué datos o detalles útiles nos brinda este artículo?
Este artículo realiza un análisis del modelado cognitivo en el MATH Story problem Tutor (MAST). Además, la simbolización de los "word problems" (convertir las palabras en problemas simbólicos) y el seguimiento de los errores son las principales contribuciones de este artículo a MAST.
Describir los elementos o características claves del modelamiento de los "Word problems"
El sistema tiene que emplear la representación del conocimiento de sus dominios seleccionados, además de mecanismos de inferencia. Para lograr estos objetivos básicamente utilizan un modelo cognitivo que tiene como objetivo desarrollar una simulación de la resolución humana de problemas en término de componentes de conocimiento que imiten a los humanos. Un modelo cognitivo basado en reglas se compone normalmente de 3 módulos principales: (a) una base de conocimientos, (b) una memoria de trabajo y (c) un motor de inferencia. La base de conocimiento incluye un conjunto de reglas de producción if-then. Una regla if-then se compone de una o varias condiciones y una o varias acciones que se realizan cuando se cumplen las condiciones. La memoria de trabajo almacena una serie de hechos que el motor de inferencia compara con las reglas de la base de conocimientos para decidir qué regla(s) ejecutar o disparar. En el contexto de la resolución de problemas, las reglas de producción representan pasos de solución según los hechos del problema en la memoria de trabajo en un dominio específico.
Describir como implementan los "Word problems" en un tutor cognitivo
No aplica.
¿El artículo describe competencias y/o habilidades asociadas a la solución de los "Word problems"?
No aplica.
¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?
Básicamente utilizan un árbol para extraer palabras claves del enunciado. Estas palabras claves se utilizan para identificar los parámetros del problema necesarios para el proceso de simbolización.
¿El artículo describe los tipos de feedback entregados a los estudiantes?
No aplica.
¿Cómo el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?
No aplica.
Otros hallazgos relevantes
No aplica.

El resto de tablas rellenadas de los artículos seleccionados se describen en el ANEXO A.

2.3 Principales hallazgos

Una vez analizados los artículos, previamente seleccionados, podemos dar respuesta a las preguntas de investigación mediante los hallazgos encontrados.

De esta manera, podemos describir los hallazgos encontrados según las preguntas de investigación.

1. ¿Cuáles son las características principales de los Tutores cognitivos de aprendizaje de matemáticas que implementen “word problems” o ejercicios contextualizados?

Principales hallazgos según 1:

Dentro de los artículos seleccionados se encontró que los tutores cognitivos para el aprendizaje de matemáticas, en particular para la resolución de “*word problems*” o ejercicios contextualizados utilizan un modelo cognitivo basado en reglas como herramienta para simular la resolución humana de problemas. Estos modelos se componen normalmente de tres módulos principales: (a) Una base de conocimientos, (b) una memoria de trabajo y (c) un motor de inferencia. La base de conocimiento incluye un conjunto de reglas de producción del tipo “*if-then*”. Una regla *if-then* se compone de una o varias condiciones y una o varias acciones que se realizan cuando se cumplen las condiciones. La memoria de trabajo almacena una serie de hechos que el motor de inferencia compara con las reglas de la base de conocimientos para decidir qué regla(s) ejecutar. En el contexto de la resolución de problemas, las reglas de producción representan pasos de solución según los hechos del problema, en la memoria de trabajo, en un dominio específico.

En cuanto al *feedback* entregado al estudiante, en general los tipos de ayuda entregados que destacan son la ayuda a pedido, es decir cuando el estudiante solicita ayuda cuando se encuentra estancado en un paso. La retroalimentación de apoyo se centra en proporcionar al estudiante, cuando éste comete un error, un conjunto de pistas incrementales hasta que sea capaz de resolver el problema correctamente. Además, hay que destacar la implementación en algunos ITS de sugerencias visuales en forma de palabras claves resaltadas en el problema al detectar un error del estudiante. Por último, cabe mencionar el uso de ayudas o *tips* conceptuales relacionados al problema que se está resolviendo, con el fin de cubrir lagunas conceptuales que presenten los estudiantes.

En cuanto a la segunda pregunta de investigación:

2. ¿Cómo se modelan o implementan los ejercicios contextualizados o “*word problems*” en los tutores cognitivos de matemática?

Principales hallazgos según 2:

En cuanto al modelamiento de los ejercicios textuales, se encontró que el método cartesiano (MC) es el método más utilizado como esqueleto sobre el cual se estructura el proceso de resolución de problemas contextualizados. El MC se compone de los siguientes siete pasos:

- 1) La lectura analítica del enunciado del problema para transformarlo en una lista de cantidades y relaciones entre cantidades.
- 2) Elegir una cantidad (o varias) que se designa con una letra (o varias letras diferentes).
- 3) Escribir expresiones algebraicas para designar las demás cantidades, utilizando la letra (o letras) introducida en el segundo paso y las relaciones encontradas en la lectura analítica realizada en el primer paso.
- 4) Escribir una ecuación (o tantas ecuaciones independientes como el número de letras introducidas en el segundo paso), a partir de la observación de que dos expresiones algebraicas (no equivalentes) escritas en el tercer paso designan la misma cantidad.
- 5) Transformar la ecuación en su forma canónica.
- 6) La aplicación de la fórmula o el algoritmo de solución a la ecuación en su forma canónica.
- 7) La interpretación del resultado en términos del planteamiento del problema.

Estos siete pasos metodológicos ayudan al estudiante a comprender los conceptos básicos de la resolución de problemas algebraicos. Además, cabe mencionar que ciertos artículos describen distintas herramientas para la representación del conocimiento algebraico de los problemas textuales, tales como árboles o hipergrafos³. Estos últimos resultan ser especialmente útil para el primer paso del MC, el cual implica realizar una lectura analítica del problema mediante la cual se traduce el enunciado a una serie de variables y relaciones entre éstas. En este sentido los hipergrafos se pueden utilizar para representar la lectura analítica de un problema textual y también para representar el espacio de soluciones posibles de un problema algebraico.

Finalmente, se encontró que los tutores cognitivos para el aprendizaje de matemáticas generalmente proporcionan una estructura en la que el estudiante tiene que completar cada paso para luego pasar al siguiente. En cuanto a la construcción de las expresiones algebraicas, cabe destacar que los ITS generalmente utilizan un componente gráfico similar a una calculadora que contiene un botón para cada operador aritmético, y otros para cada variable que el estudiante haya definido previamente.

³ En matemáticas y ciencias de la computación, un hipergrafo es una generalización de un grafo, cuyas aristas aquí se llaman hiperaristas, y pueden relacionar a cualquier cantidad de vértices, en lugar de solo un máximo de dos como en el caso de los grafos

2.4 Conclusiones de la revisión sistemática

Fue posible reconocer que el modelamiento e implementación de ejercicios textuales y/o contextualizados en un tutor cognitivo es un tema complejo, en el cual se deben considerar numerosos factores para lograr una correcta implementación que permitan lograr una adecuada comprensión y resolución de los problemas por parte de los estudiantes.

Además, como resultado de la revisión sistemática se encontró que los tutores cognitivos para el aprendizaje de matemáticas, en particular para la resolución de “*word problems*” o ejercicios contextualizados, utilizan un modelo cognitivo basado en reglas como herramienta para simular la resolución humana de problemas. Además, se identificó que el método cartesiano (MC) es el método más utilizado como esqueleto sobre el cual estructurar el proceso de resolución de los problemas contextualizados. Por otro lado, se evidenció que los hipergrafos se pueden utilizar para representar la lectura analítica de un problema textual y también para representar el espacio de soluciones posibles de un problema algebraico. Finalmente, se determinó que los tutores cognitivos para el aprendizaje de matemáticas suelen utilizar un componente gráfico similar a una calculadora para la construcción de las expresiones algebraicas.

3. METODOLOGÍA DE DESARROLLO

En este trabajo de título, se adaptó la metodología ágil SCRUM como base para el desarrollo del proyecto. La elección de esta metodología se debe a su capacidad para permitir un desarrollo flexible, iterativo e incremental, lo que resulta fundamental para lograr un proceso de desarrollo eficiente y adaptativo. Esto permite recibir retroalimentaciones de manera temprana en cada iteración y asegurar la entrega continua de valor.

3.1 Scrum

Scrum es una metodología de trabajo iterativa e incremental para la gestión de proyectos, utilizado principalmente en el desarrollo ágil de software (Drumond, s.f). La metodología *Scrum* hace hincapié en el software funcional, la colaboración y la flexibilidad para cambiar juntos con las realidades de negocio emergentes como páginas web.

Scrum se basa en la división del trabajo en iteraciones llamadas *Sprints*, que son períodos de tiempos cortos, generalmente de una a cuatro semanas. Cada *Sprint* se inicia con una planificación (*Sprint Planning*) en las que se definen las tareas a realizar y se establecen los objetivos específicos para ese periodo (Rehkopf, s.f). Durante el *Sprint*, se llevan a cabo las actividades de desarrollo, pruebas y revisión.

Uno de los principios claves de *Scrum* es la entrega continua de valor al final de cada *Sprint*. Esto implica que, al finalizar cada iteración, se debe contar con una versión funcional del software. Esta entrega incremental permite recibir el *feedback* de los usuarios y *Stakeholders* de manera temprana, lo que facilita la detección de posibles errores, mejoras y ajustes necesarios.

En la metodología *Scrum* existen cuatro roles fundamentales: *Product Owner* (dueño del producto), *Scrum Master* (líder del proyecto), *Scrum Team* (equipo de desarrollo) y los *Stakeholders* (interesados en el proyecto)

3.2 Adaptación de metodología

Debido a que este proyecto no cuenta con un equipo de trabajo numeroso, se adaptó la metodología para trabajar con un equipo reducido. Para esto se definieron los roles de la siguiente manera: el rol de *Product Owner* será ocupado por el profesor patrocinante, los roles de *Scrum Master* y *Scrum Team* serán ocupados por el estudiante tesista, mientras que el rol de *Stakeholders* será ocupado por el profesor co-patrocinante quien es profesor de matemáticas de la FCI.

3.2.1 Sprint

Debido a la naturaleza del proyecto, que no cuenta con un equipo de desarrollo, no se llevaron a cabo reuniones entre el *Scrum Master* y el *Scrum Team*. En cambio, el

estudiante tesista asumió ambos roles y trabajó en estrecha colaboración con el profesor patrocinante (*Product Owner*) y co-patrocinante (*Stakeholder*) para planificar las tareas y actividades de cada *Sprint*.

A pesar de esta adaptación, se mantuvo el enfoque iterativo e incremental de SCRUM. Se estableció una dinámica de reuniones semanales, tanto con el profesor patrocinante como con el profesor co-patrocinante, en las cuales se presentan los avances realizados, se discuten las dificultades encontradas, se validan los progresos con el *Stakeholder* y se definen las tareas a realizar para la siguiente iteración.

3.3 Tecnologías de desarrollo

Para el desarrollo de este proyecto, se utilizarán las tecnologías que han sido empleadas a lo largo del proyecto general de tutores cognitivos. Estas tecnologías se describen a continuación:

3.3.1 ReactJS

ReactJS es una biblioteca JavaScript para el desarrollo de interfaces de usuarios web y nativas. Proporciona una colección de fragmentos de código JavaScript reutilizables, utilizados para crear interfaces de usuario (UI) llamadas componentes (Open Source Community, s.f.).

3.3.2 NextJs

Next.js es un *framework* de desarrollo web de código abierto basado en React.js. Proporciona funcionalidades adicionales para la creación de aplicaciones web como enrutamiento, renderizado del lado del servidor (SSR), generación de sitios estáticos, entre otros (Next.Js by Vercel, s.f.).

3.3.3 ChakraUI

Chakra UI es una biblioteca de componentes simples, modulares y accesibles para crear aplicaciones React (Adebayo, s.f.). Chakra UI proporciona una amplia gama de componentes básicos que son fáciles de usar y personalizar para construir aplicaciones web. Estos componentes incluyen botones, formularios, tarjetas, barras de navegación y más, lo que facilita la creación de interfaces de usuario estilizadas y funcionales.

3.3.4 TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado por Microsoft que se basa en JavaScript. Añade características de tipado estático y permite una mejor herramienta de desarrollo para construir aplicaciones escalables y robustas (TypeScript. s.f.).

4. DISEÑO Y MODELADO DEL CONTENIDO

Este capítulo abordará los objetivos específicos 2 y 3, los cuales corresponden a la elaboración de los ejercicios, la definición del proceso de resolución para cada uno de ellos (solución paso a paso), la especificación de los componentes utilizados en cada paso y el modelado de los ejercicios.

Durante esta etapa, se llevaron a cabo reuniones semanales con el profesor patrocinante y el profesor co-patrocinante, en las cuales se presentaron los avances en el diseño y modelado de los ejercicios, así como para recibir sugerencias y retroalimentación por parte de los profesores.

La retroalimentación recibida en estas reuniones permitió refinar y mejorar el diseño y modelado de los ejercicios. Además, en estas se evaluaron aspectos como la claridad de los enunciados, la coherencia de la estructura de solución paso a paso, la pertinencia de los componentes utilizados y la secuencia lógica de los pasos de resolución. En cada iteración se realizaron ajustes y modificaciones de acuerdo a las recomendaciones de los profesores, asegurando así la claridad y la calidad de los ejercicios.

4.1 Diseño de ejercicios

Para el diseño de los ejercicios, se llevó a cabo una reunión inicial con el profesor co-patrocinante. En esta reunión, se analizaron los diferentes tópicos disponibles en la plataforma y se identificaron tipos de ejercicios que podrían abarcar otros tópicos del tutor ya implementado, permitiendo así generar ejercicios que aborden diversas áreas de manera integrada. Además, se realizó una exhaustiva búsqueda en diversas fuentes, tanto en internet como en material proporcionado por el profesor co-patrocinante, con el fin de obtener ejercicios de ejemplo que sirvieran como referencia para la creación de los ejercicios a implementar en el tutor cognitivo.

Con la información recopilada, se procedió a construir una síntesis (descrita en la Tabla 4) que contiene los principales tópicos identificados, así como una descripción del contenido de los ejercicios relacionados. Esta síntesis proporcionó una visión general de los temas a abordar y sirvió como guía durante el proceso de diseño y desarrollo de los ejercicios.

Tabla 4: Identificación de tópicos

Tópico	Descripción
Potencias	<ul style="list-style-type: none">Ejercicio sobre notación científica en el contexto medioambiental
Fracciones	<ul style="list-style-type: none">Ejercicio sobre resistencia en serie y en paralelo, en el contexto de sensores electrónicos.Ejercicio sobre velocidad, tiempo y distancia, en el contexto de transporte vehicular.

Tabla 5: (Continuación)

Tópico	Descripción
Ecuaciones	<ul style="list-style-type: none"> • Ejercicios sobre evaluación de ecuaciones lineales y cuadráticas, en donde se integren fracciones (mediante porcentajes), en contexto medioambiental.
Sistemas de ecuaciones	<ul style="list-style-type: none"> • Ejercicio donde se compare la plantación de dos tipos de árboles y se establezca un sistema de ecuaciones

4.1.1 Desarrollo de set de ejercicios

Para el desarrollo del nuevo set de ejercicios, se le asignó al estudiante tesista la tarea de crear 2 ejercicios por cada tópico mencionado en la Tabla 4, esto con el fin de contar con una amplia variedad de opciones para luego seleccionar los mejores ejercicios a implementar en el tutor cognitivo.

El proceso de desarrollo de los ejercicios involucró una búsqueda exhaustiva en diversas fuentes, tanto en internet como en materiales proporcionados por el profesor copatrocincante. En esta búsqueda se realizó una revisión minuciosa de ejercicios existentes, tanto del contexto de referencia como de otros contextos, esto con el objetivo de obtener ideas y buenas prácticas. A partir de esta búsqueda, se identificó una estructura común generalmente usada en ejercicios con contexto (con un enunciado, una descripción visual o tabulada y una secuencia de preguntas, tal como se presenta en la Figura 4), la cual fue utilizada como base para generar el set de ejercicios.



Figura 4. Estructura de los ejercicios

Además, se realizó una exploración de diversos contextos en los cuales se pudieran aplicar los ejercicios matemáticos, asegurando así su relevancia y su aplicabilidad en situaciones reales socio y medioambientales. Este proceso culminó con un set de once ejercicios que cubrían los tópicos mencionados en la Tabla 4. Estos ejercicios fueron cuidadosamente diseñados y desarrollados por el estudiante tesista teniendo en cuenta el contexto definido para este proyecto y las recomendaciones recibidas en las reuniones con el profesor patrocinante y co-patrocinante.

Una vez creados los ejercicios, se procedió a realizar un primer procedimiento de solución paso a paso para cada uno de ellos. En la siguiente reunión de equipo se presentaron los ejercicios junto con sus soluciones, donde se recibieron comentarios y retroalimentación por parte de ambos profesores. Durante esta reunión se hizo hincapié en la importancia de utilizar *datos reales* en el desarrollo de los ejercicios y de asegurar que tuvieran un propósito de aplicación real, de manera que la relación entre el contexto y la resolución del ejercicio fuera significativa y relevante para los estudiantes.

A lo largo de las siguientes iteraciones, se continuó mejorando y refinando el set de ejercicios, centrándose en aspectos como la claridad del enunciado y las preguntas, el proceso de solución paso a paso y la explicación previa a cada paso. Esto último debido a que, en conjunto con los profesores, se definió que las soluciones de los ejercicios debían ser un proceso continuo de aprendizaje en lugar de un simple proceso de resolución. Por lo tanto, se acordó guiar constantemente al estudiante, siempre y cuando sea posible, mediante explicaciones y ejemplos previos a la resolución de cada paso, con el objetivo de promover una mejor comprensión de los ejercicios.

4.1.2 Selección de ejercicios

Una vez desarrollado el set de ejercicios, se procedió a seleccionar los ejercicios más relevantes. Esta selección fue realizada principalmente según el criterio del profesor de matemáticas (profesor co-patrocinante), quien escogió aquellos ejercicios que eran potencialmente más pertinentes y efectivos para el aprendizaje de los estudiantes de primer año de ingeniería. Como resultado de este proceso, se seleccionaron un total final de 5 ejercicios, los cuales se detallan en la Tabla 6.

Tabla 6: Descripción de ejercicios seleccionados

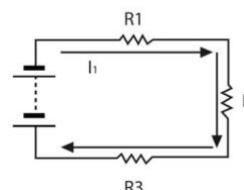
Tópico Principal	Descripción de ejercicio										
Potencia	<p>El gobierno de Chile realiza un seguimiento de la cantidad de emisiones de gases de efecto invernadero (GEI) en gramos de CO₂ de los principales sectores contribuyentes en la emisión y absorción de GEI. La siguiente tabla muestra los valores reales en chile de las emisiones y absorción de los cuatro principales sectores en 2013:</p> <table border="1"> <thead> <tr> <th>Sector</th><th>CO₂ neto</th></tr> </thead> <tbody> <tr> <td>Energía</td><td>$8,18231 \times 10^{13}$ gr</td></tr> <tr> <td>Procesos industriales</td><td>$4,3602 \times 10^{12}$ gr</td></tr> <tr> <td>Usos de la tierra (Agricultura, Silvicultura y otros)</td><td>$-3,9915 \times 10^{13}$ gr</td></tr> <tr> <td>Residuos</td><td>3×10^8 gr</td></tr> </tbody> </table> <p>*Los números positivos representan emisiones de GEI, mientras que los negativos representan absorciones de GEI.</p> <p>A partir de la tabla anterior, nos gustaría identificar:</p> <ol style="list-style-type: none"> 1. ¿Qué sector emite más CO₂, contribuyendo mayormente a la contaminación? 2. ¿Qué sector absorbe CO₂, disminuyendo la contaminación? 3. Convierte los valores de CO₂ de los sectores en potencias de 10^9 4. ¿Cuál es el total de emisiones? 5. ¿Cuál es el balance total entre las emisiones y absorciones de CO₂? 6. ¿Cuál es el porcentaje que aporta el sector de energía en las emisiones totales? <p>Realiza los cálculos en gigagramos (Unidad de masa equivalente a mil millones de gramos o 10^9 gramos.)</p>	Sector	CO ₂ neto	Energía	$8,18231 \times 10^{13}$ gr	Procesos industriales	$4,3602 \times 10^{12}$ gr	Usos de la tierra (Agricultura, Silvicultura y otros)	$-3,9915 \times 10^{13}$ gr	Residuos	3×10^8 gr
Sector	CO ₂ neto										
Energía	$8,18231 \times 10^{13}$ gr										
Procesos industriales	$4,3602 \times 10^{12}$ gr										
Usos de la tierra (Agricultura, Silvicultura y otros)	$-3,9915 \times 10^{13}$ gr										
Residuos	3×10^8 gr										
Fracciones	<p>Se tiene un circuito de tres sensores en serie conectados a una fuente de alimentación de 12V. El circuito se utiliza para medir la humedad en una sala de cultivos y cualquier fallo en la medición podría tener consecuencias negativas en el crecimiento de las plantas.</p>  <p>Si la resistencia de cada sensor es de 100 ohm, es decir $R_1=R_2=R_3 = 100$ ohm</p> <ol style="list-style-type: none"> 1. ¿Cuál es la resistencia total del circuito de sensores en serie? 2. Si uno de los sensores falla, ¿Cómo afectaría esto a al circuito? 3. Si se cambia el circuito a un circuito en paralelo ¿Cuál sería la resistencia equivalente? 4. Determinar el valor 'x' (en ohm) de tres nuevas resistencias para mantener una resistencia equivalente en paralelo igual a la resistencia en serie de 300 ohm. 										

Tabla 7: (Continuación)

Tópico Principal	Descripción de ejercicio																										
Ecuaciones Lineales	<p>En una investigación realizada en la Universidad de Chile se ha observado la superficie del glaciar Grey a lo largo del tiempo. En la siguiente tabla podemos observar las mediciones realizadas de la superficie del glaciar Grey entre los años 1979 y 2020.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Años</th> <th>Área (km²)</th> </tr> </thead> <tbody> <tr><td>1979</td><td>265,05</td></tr> <tr><td>1984</td><td>260,98</td></tr> <tr><td>1986</td><td>260,11</td></tr> <tr><td>1989</td><td>258,76</td></tr> <tr><td>1998</td><td>247,23</td></tr> <tr><td>2001</td><td>245,26</td></tr> <tr><td>2005</td><td>240,46</td></tr> <tr><td>2007</td><td>240,38</td></tr> <tr><td>2011</td><td>236,87</td></tr> <tr><td>2014</td><td>236,15</td></tr> <tr><td>2016</td><td>234,14</td></tr> <tr><td>2020</td><td>231,70</td></tr> </tbody> </table> <p style="text-align: center;">Áreas en km² por año del glaciar Grey</p> <ol style="list-style-type: none"> 1. ¿Según los datos de la tabla, ha disminuido o ha aumentado la superficie del glaciar de Grey a lo largo del tiempo? 2. Identificar las variables dependientes e independientes y representar los datos en un gráfico cartesiano 3. Calcular la superficie de hielo estimada para el año 2023. 4. Estimar en qué año la superficie del glaciar llegaría al 50% del tamaño medido en 1979. 	Años	Área (km ²)	1979	265,05	1984	260,98	1986	260,11	1989	258,76	1998	247,23	2001	245,26	2005	240,46	2007	240,38	2011	236,87	2014	236,15	2016	234,14	2020	231,70
Años	Área (km ²)																										
1979	265,05																										
1984	260,98																										
1986	260,11																										
1989	258,76																										
1998	247,23																										
2001	245,26																										
2005	240,46																										
2007	240,38																										
2011	236,87																										
2014	236,15																										
2016	234,14																										
2020	231,70																										
Ecuación cuadrática	<p>El rendimiento de combustible de un automóvil se obtiene de acuerdo a la velocidad con la que se desplaza, si ‘x’ es la velocidad media en kilómetros por hora (km/h) y el rendimiento está dado por la siguiente expresión:</p> $y = -\frac{1}{40}x^2 + \frac{7}{2}x, \text{ para } 0 < x < 120$ <ol style="list-style-type: none"> 1. Calcula el rendimiento para las velocidades de 20, 60 y 100 en km/h 2. ¿A qué velocidad se obtiene el máximo rendimiento? 3. ¿Cuál es el máximo rendimiento? 																										
Sistemas de ecuaciones lineales	<p>En un terreno de 1 hectárea se desea plantar dos tipos de árboles, eucaliptos y pinos radiata. Se sabe que un eucalipto necesita 9 metros cuadrados de espacio y tiene un costo de plantación de \$528, aproximadamente, mientras que un pino radiata necesita 7 metros cuadrados de espacio y tiene un costo de plantación de \$544, aproximadamente.</p> <ol style="list-style-type: none"> 1. Sin considerar los precios, ¿Qué tipo de árbol usarías si quisieras tener más árboles en esta plantación? 2. Si contamos con un presupuesto de \$640.000, ¿cuántos árboles de cada tipo se deben plantar para maximizar el espacio? 																										

4.2 Definición de componentes

Una vez que se estableció una versión inicial de los pasos necesarios para resolver cada pregunta de los ejercicios, se procedió a definir los componentes que se utilizarán para responder cada paso. Durante este proceso, se identificaron tres componentes principales: un componente matemático, un componente de selección y un componente gráfico.

El componente matemático debe permitir generar pasos secuenciales en los cuales el estudiante pueda ingresar tanto respuestas completas como respuestas parciales. Esto significa que se pueda proporcionar al estudiante una parte estática (ya completada) de la respuesta y permitirle completar otras ciertas partes específicas de la expresión matemática, definidas en la elaboración del ejercicio. De esta manera, se puede guiar al estudiante en la resolución del paso, brindándole apoyo y orientación durante todo el proceso de resolución.

En cuanto al componente de selección de respuesta, se decidió incluirlo debido a que los ejercicios con contexto a menudo tienen preguntas abiertas de identificación o comprensión, lo cual puede llevar a un amplio rango de posibles soluciones. Por esta razón, se decidió que el estudiante deba responder a este tipo de preguntas seleccionando la respuesta correcta de entre varias opciones proporcionadas.

Finalmente, se decidió la inclusión de un componente gráfico para aquellos casos en los que se requiera representar visualmente la información del ejercicio matemático planteado. Este componente permitirá a los estudiantes interactuar con el gráfico para responder las preguntas correspondientes del ejercicio asociado.

4.3 Análisis cognitivo y diseño de resolución de ejercicios

Con los ejercicios a implementar ya seleccionados y con una primera versión de la solución paso a paso de cada ejercicio realizada por el estudiante tesista, se llevó a cabo un análisis cognitivo sobre el procedimiento de resolución de cada ejercicio. En este análisis, se trabajó en conjunto con los profesores para refinar la secuencia de solución paso a paso de cada ejercicio.

Además, se determinó una secuencia de solución paso a paso para abordar cada pregunta de manera individual, en lugar de considerar el ejercicio completo como una única entidad. Este enfoque permitió un análisis más detallado y preciso de cada pregunta, asegurando que se aborden adecuadamente los conceptos claves y los desarrollos necesarios para llegar a la solución correcta de cada una de ellas.

A continuación, se muestra el diseño de solución paso a paso realizado para cada ejercicio definido previamente en la Tabla 6.

4.3.1 Diseño de solución paso a paso del Ejercicio 1

En el caso del Ejercicio 1, se han definido las dos primeras preguntas como preguntas de identificación de información, con el objetivo de verificar que el estudiante comprende la información proporcionada en el enunciado del ejercicio.

Para abordar estas preguntas, se ha establecido un paso único denominado "Selecciona la alternativa correcta" que presenta varias opciones de respuesta, pero solo una correcta. Además, se ha agregado un recordatorio previo al primer paso para reforzar los conocimientos del estudiante. La Figura 5 y la Figura 6 muestran las soluciones correspondientes a estas preguntas.

1. ¿Qué sector emite más CO₂, contribuyendo mayormente a la contaminación?

Recordatorio: La notación científica describe números basados en potencia de 10. Su forma es:

$$m \times 10^n$$

Donde 'm' es un número decimal cuya parte entera tiene una sola cifra distinta de cero y 10ⁿ es una potencia de exponente entero.

Paso 1: Selecciona la alternativa correcta



Figura 5: Diseño de pregunta 1 del Ejercicio 1.

2. ¿Qué sector absorbe CO₂, disminuyendo la contaminación?

Paso 1: Selecciona la alternativa correcta



Figura 6: Diseño de pregunta 2 del Ejercicio 1.

Para abordar la tercera pregunta, se han identificado inicialmente cuatro pasos que implican la transformación de los números que representan los valores de emisiones de CO₂ de cada sector a una misma potencia, con el fin de facilitar las operaciones. Sin embargo, considerando la necesidad de brindar explicaciones y ejemplos al estudiante para guiarlo en el proceso de resolución, se proporcionará una breve explicación previa junto con un ejemplo correspondiente al primer paso, que consiste en transformar los valores de emisión del sector de energía (ver Figura 7). De esta manera, se reduce el número de pasos a tres.

En estos pasos, se proporcionará parte de la solución de manera fija e incompleta, lo que permitirá al estudiante completar la respuesta utilizando un "placeholder" (cuadro de texto) (ver Figura 7). Esto tiene como objetivo facilitar la participación activa del estudiante en el proceso de resolución y su comprensión de los pasos necesarios para realizar la transformación de los valores de emisión en los otros sectores.

3. Convierte los valores de CO₂ de los sectores en potencias de 10⁹

Ahora, nos interesa conocer el total de emisiones y el total de absorciones de GEI.

Para esto, primero convertimos los datos a la misma potencia para poder sumarlos.

Realiza los cálculos en gigagramos (masa equivalente a mil millones de gramos ó 10⁹)

Por ejemplo, con el sector de energía se tiene:

$$8,18231 * 10^{13} = 8,18231 * 10^4 * 10^9 = 81823,1 * 10^9$$

Paso 1: Convertir datos de sector "Procesos industriales"

$$\begin{aligned} 4,3602 * 10^{12} &= 4,3602 * 10^{[3]} * 10^9 \\ &= [\mathbf{4360,2}] * 10^9 \end{aligned}$$

Paso 2: Convertir datos de sector "Usos de la tierra"

$$\begin{aligned} -3,99152 * 10^{13} &= -3,9915 * 10^{[4]} * 10^9 \\ &= [\mathbf{-39915}] * 10^9 \end{aligned}$$

Paso 3: Convertir datos de sector "Residuos"

$$\begin{aligned} 3 * 10^8 &= 3 * 10^{[-1]} * 10^9 \\ &= [\mathbf{0,3}] * 10^9 \end{aligned}$$

Figura 7: Diseño de pregunta 3 del Ejercicio 1

En relación a las preguntas cuatro y cinco, se ha definido un único paso para cada una de ellas, denominados "Calcular total de emisiones" y "Calcular balance" respectivamente. En estos pasos, el estudiante debe utilizar los datos previamente calculados para determinar el valor solicitado y responder cada una de las preguntas.

Con el objetivo de facilitar el proceso de respuesta y manejar la diversidad de posibles respuestas, se ha optado por proporcionar el texto de la respuesta de manera explícita pero incompleta, y se solicitará al estudiante que complete la respuesta con el valor calculado en un campo de texto. Este enfoque se ilustra en las siguientes dos figuras (Figura 8 y Figura 9), donde se muestra cómo se plantea la resolución de cada una de las preguntas.

4. ¿Cuál es el total de emisiones?

Recuerda que los valores de los sectores son:

- Sector de Energía: $81823,1 * 10^9$
- Sector de Procesos industriales: $4360,2 * 10^9$
- Sector de Residuos: $0,3 * 10^9$
- Sector de Usos de la tierra: $-39915 * 10^9$

Paso 1: Calcular total de emisiones

El total de emisiones es de $[86183,6] * 10^9$ gramos de CO₂

Figura 8: Diseño de pregunta 4 del Ejercicio 1

5. ¿Cuál es el balance entre las emisiones y absorciones de CO₂?

Paso 1: Calcular balance

El balance total es de $[46268,6] * 10^9$ gramos de CO₂

Figura 9: Diseño de pregunta 5 del Ejercicio 1

En la pregunta seis, se ha identificado un único paso clave para resolverla, denominado "Calcular proporción". En este paso, el estudiante deberá realizar los cálculos necesarios para determinar la proporción del sector de energía con respecto a las emisiones totales.

Además, para ayudar al estudiante a comprender el proceso, se proporcionará un ejemplo donde se calculará la proporción de otro sector. Este ejemplo servirá como guía para que el estudiante pueda aplicar el mismo método de cálculo con el sector de energía. La Figura 10, representa el diseño de esta pregunta.

6. ¿Cuál es el porcentaje que aporta el sector de energía en las emisiones totales?

Si queremos calcular que porcentaje aporta cierto sector debemos dividir las emisiones del sector entre las emisiones totales y multiplicar el resultado por 100.

Por ejemplo, para el sector de Procesos industriales se tiene:

$$\frac{\text{Emisiones sector Procesos Industriales}}{\text{Total emisiones}} = \frac{4360,2 * 10^9}{86183,6 * 10^9} \approx 0,05$$

Por lo tanto, el sector de Procesos industriales aporta en 5%, aproximadamente, de las emisiones totales.

Paso 1: Calcular proporción (aproxima a dos decimales)

$$\frac{\text{Emisiones sector Energía}}{\text{Total emisiones}} = \frac{[81823,1] \times 10^9}{[86183,6] \times 10^9} \approx [0.95]$$

¡Muy bien! Por lo tanto, el sector de energía aporta en un 95%, aproximadamente, de las emisiones totales.

Figura 10: Diseño de pregunta 7 del Ejercicio 1

4.3.3 Diseño de resolución paso a paso del Ejercicio 2

Para la primera pregunta del Ejercicio 2, se ha identificado un único paso denominado "Calcular resistencia en serie". La Figura 11 presenta el desarrollo de la pregunta junto con el paso identificado. Además, antes de realizar el paso, se proporcionará al estudiante un breve texto explicativo sobre cómo realizar el cálculo y se le entregará la fórmula correspondiente para calcular el valor de la resistencia en serie solicitada. Una vez que el estudiante complete el paso correctamente, se le proporcionará un mensaje de felicitaciones corroborando la respuesta completa con el valor correctamente calculado.

1. ¿Cuál es la resistencia total del circuito de sensores en serie?

En un circuito en serie la resistencia total del circuito se calcula sumando las resistencias individuales

$$R_{serie} = R1+R2+R3$$

Paso 1: Calcular resistencia en serie

$$R_{serie} = [100] + [100] + [100]$$

$$R_{serie} = [300] \text{ ohm}$$

Muy bien! por lo tanto, la resistencia total en el circuito en serie es de 300 ohm

Figura 11: Diseño de pregunta 1 del Ejercicio 2.

Para abordar la pregunta dos, la cual fue designada como una pregunta de selección, se ha establecido un paso específico denominado "Selecciona la alternativa correcta". Este paso el estudiante deberá elegir la opción adecuada entre las opciones proporcionadas. La Figura 12 ilustra la definición de la pregunta y muestra el paso correspondiente.

2. Si uno de los sensores falla, ¿Cómo afectaría esto al circuito?

Paso 1: Selecciona la alternativa correcta

1. El circuito continúa funcionando sin la lectura del sensor estropeado.

2. El circuito se interrumpe por completo y dejaría de funcionar

Figura 12: Diseño de pregunta 2 del Ejercicio 2.

En relación a la tercera pregunta, se ha decidido proporcionar un texto informativo antes del primer paso, brindando información relevante relacionada con la pregunta. Los pasos identificados para abordar esta tercera pregunta se pueden observar en la Figura 13.

3. Si se cambia el circuito a un circuito en paralelo. ¿Cuál sería la resistencia equivalente?

Para evitar problemas de medición causados por la falla de un sensor, se puede cambiar el circuito de un circuito en serie a un circuito en paralelo. En un circuito en paralelo, cada sensor está conectado en una rama independiente del circuito y la interrupción de un sensor no afecta a los demás. Sin embargo, la resistencia cambia al pasar a un circuito en paralelo, teniendo que aplicar una tensión mayor.

En un circuito en paralelo, la resistencia equivalente se calcula sumando las inversas de las resistencias individuales y luego tomando la inversa de la suma.

$$R_{paralelo} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$$

Paso 1: Reemplaza valores de resistencias

$$R_{eq} = \frac{1}{[\frac{1}{100}] + [\frac{1}{100}] + [\frac{1}{100}]}$$

Paso 2: Suma las fracciones del denominador

$$= \frac{1}{[\frac{3}{100}]}$$

Paso 3: Reduce la expresión a una fracción simple

$$= \frac{[100]}{[3]}$$

Muy bien! la resistencia equivalente es aproximadamente 33,3 ohm

Figura 13: Diseño de pregunta 3 del Ejercicio 2.

Para abordar la cuarta pregunta, se han identificado cuatro pasos clave: "Modelar ecuación", "Sumar las fracciones del denominador", "Reducir la expresión a una fracción simple" y "Despejar x de la expresión anterior".

En el primer paso, denominado "Modelar ecuación", se le solicita al estudiante que forme la ecuación correspondiente para obtener el valor de "x" de la resistencia equivalente. En este paso, el estudiante debe reemplazar el valor correspondiente en la expresión para calcular la resistencia en un circuito en paralelo.

En el segundo paso, denominado "Sumar las fracciones del denominador", se le pide al estudiante que realice la suma de las fracciones presentes en el denominador de la

ecuación. Este paso tiene como objetivo simplificar la expresión y llevarla a una forma más manejable.

En el tercer paso, denominado "Reducir la expresión a una fracción simple", se solicita al estudiante que simplifique la expresión obtenida en el paso anterior y la lleve a una fracción simple. Esto implica realizar operaciones aritméticas adicionales para simplificar y reducir la expresión lo más posible.

Por último, en el cuarto paso, denominado "Despejar x de la expresión anterior", el estudiante debe despejar la variable "x" de la expresión obtenida en el paso anterior y calcular su valor correspondiente.

La Figura 14 representa el diseño para esta pregunta, mostrando los pasos mencionados y proporcionando la estructura necesaria para que el estudiante complete cada paso.

4. Determinar el valor 'x' (en ohm) de tres nuevas resistencias para mantener una resistencia equivalente en paralelo igual a la resistencia en serie de 300 ohm

Paso 1: Modelar ecuación

$$\frac{1}{[\frac{1}{x}] + [\frac{1}{x}] + [\frac{1}{x}]} = 300$$

Paso 1: Suma las fracciones del denominador

$$\Rightarrow \frac{1}{[\frac{3}{x}]} = 300$$

Paso 3: Reduce la expresión a una fracción simple

$$\Rightarrow [\frac{x}{3}] = 300$$

Paso 4: Despeja 'x' de la expresión anterior

$$\Rightarrow x = [900]$$

Muy bien! el valor de cada una de las tres nuevas resistencias debe ser de 900 ohm

Figura 14: Diseño de pregunta 4 del Ejercicio 2.

4.3.4 Diseño de resolución paso a paso para el Ejercicio 3

Para abordar la primera pregunta del ejercicio tres, la cual fue identificada como una pregunta de comprensión del enunciado, se optó por utilizar el componente de selección. De esta manera, el diseño para esta pregunta se puede observar en la Figura 15.

1. Según los datos de la tabla, ha disminuido o ha aumentado la superficie del glaciar Grey a lo largo del tiempo?

Paso 1: Selecciona la alternativa verdadera

The diagram illustrates a two-step selection process. It consists of a large rounded rectangle containing two smaller rectangular boxes. The top box contains the text "1. La superficie del glaciar ha aumentado en el tiempo". The bottom box contains the text "2. La superficie del glaciar ha disminuido en el tiempo", with this text being significantly larger and bolder than the text in the first box.

Figura 15: Diseño de pregunta 1 del Ejercicio 3.

En relación a la segunda pregunta, el profesor co-patrocinante mencionó la importancia de ser minucioso para lograr una correcta comprensión del ejercicio para su resolución. Por lo tanto, se han diseñado dos pasos iniciales de selección para verificar la comprensión del estudiante en la identificación de las variables dependiente e independientes.

Además, con el fin de mejorar la comprensión del proceso, se ha decidido agregar un breve texto antes de cada paso para establecer una mejor conexión entre ellos. La Figura 16 representa esta dinámica, mostrando los dos primeros pasos.

2. Identificar variables dependiente e independiente y representa los datos en un gráfico cartesiano.

En este ejercicio...

Paso 1: Selecciona la alternativa verdadera

1. La superficie del glaciar varia con respecto al tiempo.

2. El tiempo varía con respecto a la superficie del glaciar.

Entonces, ¿cuál es la variable dependiente y cual la independiente?

Paso 2: Selecciona la alternativa correcta

1. La variable dependiente es la superficie del glaciar y la independiente el tiempo

2. La variable dependiente es el tiempo y la independiente la superficie del glaciar.

Figura 16: Diseño de paso 1 y 2 de la pregunta 2 del Ejercicio 3.

Adicionalmente, el profesor co-patrocinante ha resaltado la importancia de que el alumno pueda identificar elementos de un gráfico en un plano cartesiano. Por esta razón, se ha definido un tercer paso, denominado "Identificar punto en el gráfico", en el cual se solicita al estudiante que identifique un punto específico dentro de los datos proporcionados en el enunciado.

Para facilitar este proceso, se le proporcionará al estudiante un gráfico que contendrá tres puntos, y su tarea será identificar y seleccionar el punto correspondiente de acuerdo con la información dada. La Figura 17 representa un ejemplo visual de este paso.

Paso 3: Identificar punto en el gráfico

Generalmente se utiliza la variable 'y' (eje vertical) para la variable dependiente y la variable 't' (eje horizontal) para la variable independiente

Identifica el punto en la gráfica que denota que en el año 1998 el área del glaciar era de $243,7 \text{ km}^2$

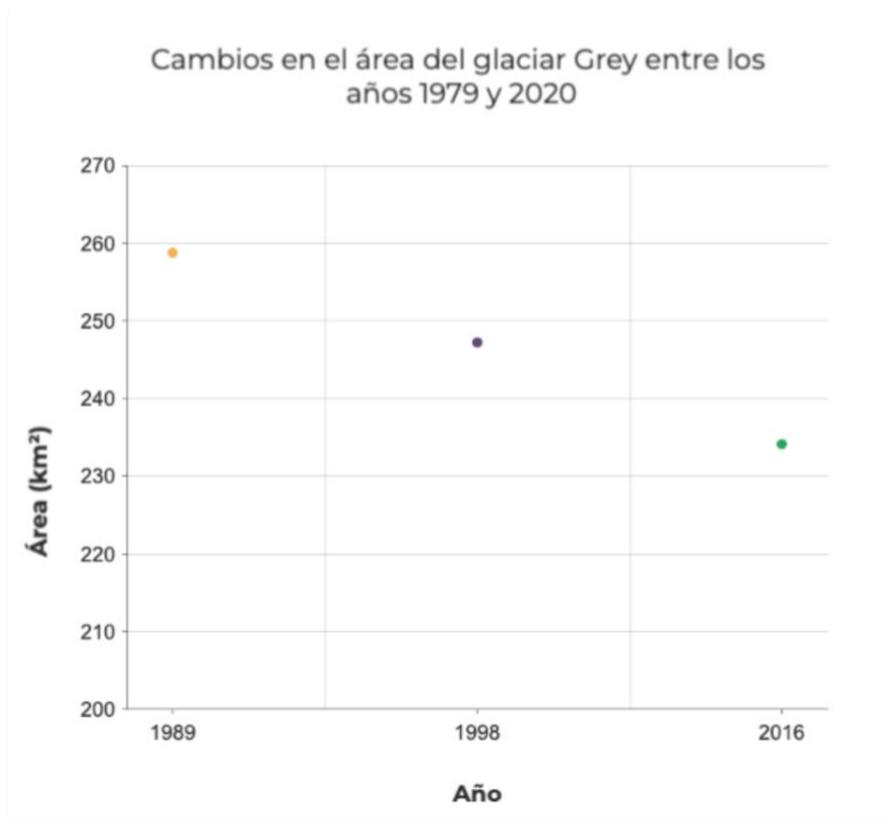


Figura 17: Diseño del paso 3 de la pregunta 2 del Ejercicio 3.

Este enfoque tiene como objetivo desarrollar la habilidad del estudiante para interpretar y trabajar con gráficos en un plano cartesiano, lo cual es relevante para la resolución de la pregunta.

Para predecir la superficie (aproximadamente) del glaciar en un año específico, se requiere ajustar los datos proporcionados. Por ello, se ha establecido el paso 4, denominado "Realiza el ajuste lineal".

En este paso, se presentará al estudiante un gráfico que mostrará los datos proporcionados junto a una recta. Además, se proporcionará la ecuación correspondiente a dicha recta, en la que el estudiante deberá ajustar la pendiente "m" usando un control deslizante (*slider*). De esta manera, el estudiante podrá utilizar la ecuación ajustada para realizar la predicción de la superficie del glaciar en un año determinado.

La Figura 18 representa el diseño correspondiente a este paso, mostrando el gráfico con los puntos y la recta, así como el control deslizante para ajustar el valor de la pendiente.

Los puntos en la gráfica nos muestran una tendencia, la cual podemos modelar mediante el ajuste de una recta. Esto nos permitirá predecir, aproximadamente, la superficie del glaciar en un año determinado.

Para realizar esto necesitamos realizar un ajuste lineal lo más aproximado a los puntos en la gráfica.

Paso 4: Realiza el ajuste lineal

Ajuste la pendiente 'm' de manera que la recta se ajuste lo mejor posible a los datos recolectados (puntos).

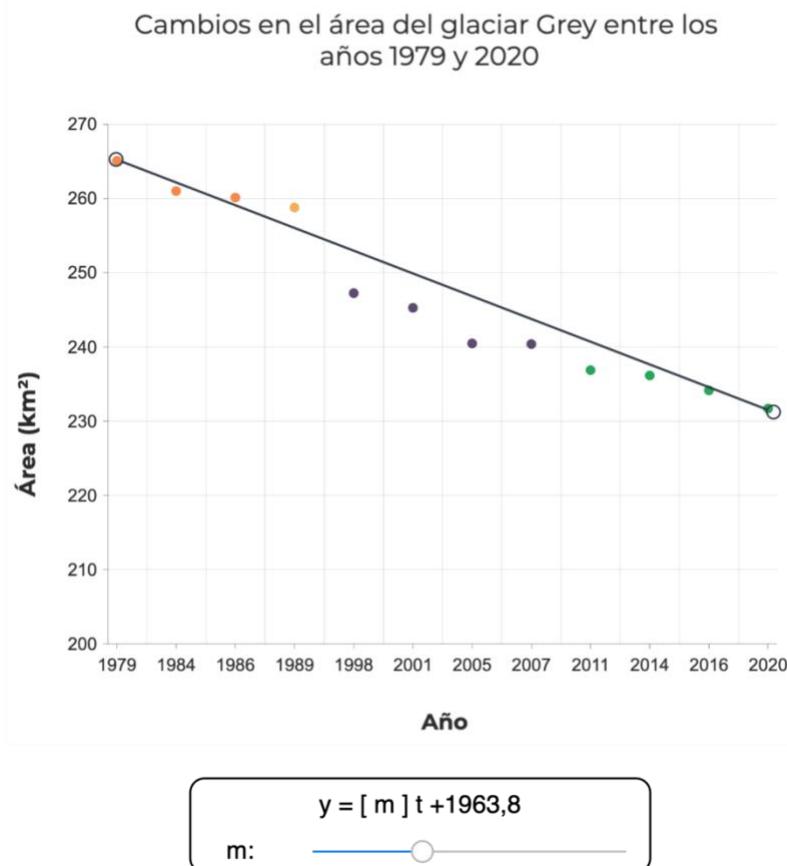


Figura 18: Diseño del paso 4 de la pregunta 2 del Ejercicio 3.

Adicionalmente al paso anterior, se ha decidido incorporar una pregunta de comprensión relacionada con la ecuación obtenida. Para esto, se ha definido el paso 5, como se muestra en la Figura 19.

Antes de este paso, se realizará al estudiante una pregunta específica y se le solicitará que seleccione la alternativa correcta como respuesta. Este enfoque tiene como objetivo evaluar la comprensión del estudiante sobre la ecuación obtenida y su capacidad para aplicarla correctamente en un contexto determinado.

Al proporcionar una pregunta de comprensión adicional, se fomenta una mayor reflexión y análisis por parte del estudiante, lo que contribuye a una comprensión más profunda del concepto y su aplicación.

¿Que significa que la pendiente de la recta sea negativa?

Paso 5: Selecciona la alternativa correcta

1. Que el área del glaciar decrece en el tiempo.

2. Que el área del glaciar crece en el tiempo.

Figura 19: Diseño del paso 5 de la pregunta 2 del Ejercicio 3.

Una vez ajustada la recta y obtenida la ecuación correspondiente, se puede proceder a responder la pregunta 3. En esta pregunta, se solicita al estudiante calcular la superficie de hielo estimada para el año 2023. Para abordar esta pregunta, se han definido dos pasos específicos.

El primer paso, denominado "Reemplaza el valor de la variable t", consiste en que el estudiante sustituya el valor del año solicitado en la ecuación del ajuste lineal. Para ello, el estudiante debe reemplazar la variable "t" por el valor correspondiente al año 2023 en la ecuación obtenida.

El segundo paso, denominado "Resuelve para y", implica que el estudiante realice las operaciones aritméticas necesarias para calcular el valor de "y". Esto implica resolver la ecuación con el valor sustituido y obtener el resultado correspondiente, que representa la superficie de hielo estimada para el año 2023.

La Figura 20 representa el diseño correspondiente a estos pasos.

3. Calcular la superficie de hielo estimada para el año 2023.

Paso 1: Reemplaza el valor de la variable 't'

$$y = -0,8583[2023] + 1963,8$$

Paso 2: Resuelve para 'y'

$$\begin{aligned}y &= [-1736,3409] + 1963,8 \\y &= [227,4591]\end{aligned}$$

Muy bien! Por lo tanto, en el año 2023 la superficie del glaciar sería de 227,4591 km²

Figura 20: Diseño de los pasos 1 y 2 de la pregunta 3.

Para abordar la pregunta 4, que consiste en estimar el año en el cual la superficie del glaciar alcanzaría el 50% del tamaño medido en el año 1979 (año inicial de los datos), se han definido tres pasos fundamentales.

El primer paso, denominado "Plantear la ecuación", requiere que el estudiante utilice la ecuación obtenida en la pregunta dos. En particular, el estudiante deberá reemplazar el valor de la pendiente "m" obtenido, en forma aproximada anteriormente, en la misma pregunta y reemplazar el valor de "y" correspondiente a la mitad de la superficie medida en el año 1979.

El segundo paso, denominado "Restar valor constante", consiste en realizar una operación previa para despejar la variable "t" correspondiente al valor solicitado en la pregunta.

Por último, el tercer paso, denominado "Despejar variable del tiempo", se enfoca en despejar la variable "t" y realizar las operaciones aritméticas necesarias para obtener su valor.

La Figura 21 representa el diseño de estos pasos, junto con la respuesta entregada en un mensaje de felicitaciones una vez que el estudiante completa correctamente los pasos.

4. Estimar en que año la superficie del glaciar llegaría al 50% del tamaño medido en 1979

Paso 1: Plantear la ecuación lineal

$$\left[\frac{265,05}{2} \right] = [-0,8583]t + 1963,8$$

Para calcular el año (representado por la variable t) en que el glaciar llegaría al 50% del tamaño medido en 1979 debemos despejar la variable 't'

Paso 2: Restar valor constante

$$\frac{265,05}{2} - [1963,8] = -0,8583t$$

Paso 3: Despejar variable del tiempo

$$t = \frac{(1963,8 - \frac{265,05}{2})}{[0,8583]}$$

$$t = [2133,60713]$$

Muy bien! Por lo tanto, en el año 2133 el glaciar llegaría al 50% de su tamaño medido en el año 1979.

Figura 21: Diseño de pregunta 4 del Ejercicio 3.

4.3.5 Diseño de resolución paso a paso para el Ejercicio 4

Para abordar la pregunta 1 del ejercicio 4, que consiste en calcular el rendimiento para las velocidades de 20, 60 y 100 km/h, se ha decidido realizar un paso específico para el cálculo de cada velocidad. De esta manera, se han definido el paso 1 para calcular el rendimiento a 20 km/h, el paso 2 para calcular el rendimiento a 60 km/h, y el paso 3 para calcular el rendimiento a 100 km/h.

Dado que estos pasos son similares en su procedimiento, se ha optado por ofrecer ayuda de manera decreciente en los pasos sucesivos. De esta manera, en el paso 1 se proporcionará al estudiante más recuadros de texto para que realice los pasos necesarios y obtenga el resultado final. En el paso 2, se le pedirá al estudiante que reemplace únicamente el valor de la velocidad correspondiente y el resultado final. Por último, en el paso 3, simplemente se le solicitará al estudiante que ingrese el resultado final, de manera que realice los cálculos de manera cada vez más autónoma.

La Figura 22 representa esta dinámica, mostrando el diseño correspondiente a los pasos mencionados y cómo se ofrecerá la ayuda de manera decreciente.

1. Calcula el rendimiento para las velocidades de 20,60 y 100 en km/h

Paso 1: Calcular rendimiento para la velocidad de 20 km/h

$$\begin{aligned}y &= -\frac{1}{40}[20]^2 + \frac{7}{2}[20] \\&= [-10] + [70] = [60]\end{aligned}$$

Paso 2: Calcular rendimiento para la velocidad de 60 km/h

$$y = -\frac{1}{40}[60]^2 + \frac{7}{2}[60] = [120]$$

Paso 3: Calcular rendimiento para la velocidad de 100 km/h

$$y = [100]$$

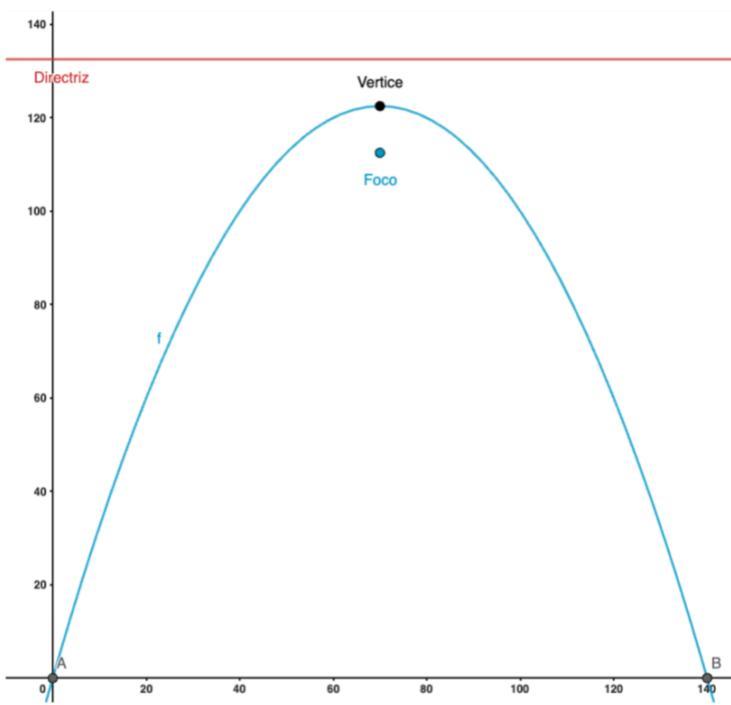
Figura 22: Diseño de pregunta 1 del Ejercicio 4.

En cuanto a la pregunta dos, se han definido tres pasos fundamentales. El primero de ellos es un paso de comprensión, en el cual se le pregunta al estudiante sobre el punto donde se obtiene el máximo rendimiento. Para facilitar su comprensión, se le entrega un breve texto explicativo y se proporciona una imagen que grafica la parábola (que representa la expresión presentada en el enunciado del ejercicio) junto con sus elementos relevantes.

En este paso, el estudiante deberá leer el texto y analizar la imagen para comprender el concepto de máximo rendimiento e identificar el punto correspondiente en la gráfica. Posteriormente, se le solicitará que responda seleccionando la alternativa correcta que corresponda a ese punto. La Figura 23 representa el diseño para este paso.

2. ¿A qué velocidad se obtiene el máximo rendimiento?

Se pretende obtener la velocidad con que se obtendría el máximo rendimiento. Sabemos que la ecuación representa una parábola invertida como se ve en la siguiente imagen



Paso 1: ¿En qué punto de la parábola podríamos encontrar su máximo rendimiento? Selecciona la alternativa correcta:

- 1. Foco
- 2. Vértice**
- 3. Directriz

Figura 23: Diseño de paso 1 de la pregunta 2 del Ejercicio 4.

El segundo paso, denominado "Identificar coeficientes", tiene como objetivo que el estudiante identifique los coeficientes necesarios para calcular la coordenada "x" del vértice, que representa el valor de la velocidad que corresponde al máximo rendimiento.

Antes de este paso, se proporcionará al estudiante un texto informativo como recordatorio, que incluirá la expresión para calcular la coordenada "x" del vértice y la ecuación

correspondiente para que pueda identificar correctamente los coeficientes necesarios (ver Figura 24).

El tercer paso, denominado "Calcular coordenada x del vértice", consiste en el reemplazo de los coeficientes identificados en el paso anterior en la expresión correspondiente y realizar los cálculos aritméticos necesarios para obtener el valor de la velocidad.

La Figura 25 representa el diseño y la resolución de ambos pasos, brindando una referencia visual para ilustrar el proceso de cálculo y mostrar cómo se obtiene el valor de la velocidad, para el máximo rendimiento, correspondiente al vértice de la parábola.

Para obtener la coordenada x del vértice, es decir el valor de la velocidad para el máximo rendimiento, podemos utilizar la siguiente expresión:

$$x = \frac{-b}{2a}$$

Donde 'b' y 'a' son coeficientes de la expresión cuadrática ($y = ax^2 + bx + c$). En nuestro caso recordemos que la ecuación es:

$$y = -\frac{1}{40}x^2 + \frac{7}{2}x$$

Paso 2: Identificar coeficientes

$$a = \left[-\frac{1}{40} \right]. \quad ; \quad b = \left[\frac{7}{2} \right]$$

Paso 3: Calcular coordenada x del vértice

$$x = \frac{-[\frac{7}{2}]}{2 * [-\frac{1}{40}]}$$

$$-\frac{7}{2} * \frac{[-40]}{2} = [70]$$

Muy bien! Por lo tanto, la velocidad que permite un mayor rendimiento de combustible es 70 km/h

Figura 24: Diseño de pasos 2 y 3 de la pregunta 2 del Ejercicio 4.

Para abordar la tercera pregunta, se ha definido un único paso denominado "Calcular valor del máximo rendimiento" (ver Figura 25). En este paso, se le entrega al estudiante la ecuación de manera parcial, por lo que el estudiante deberá completarla con el valor de velocidad calculado previamente. Además, se le solicita que realice los cálculos aritméticos necesarios para obtener el valor del máximo rendimiento.

3. ¿Cuál es el máximo rendimiento?

El máximo rendimiento de combustible está dado por 'y', al reemplazar x por el valor de la velocidad donde se obtiene el máximo rendimiento

Paso 1: Calcular valor del máximo rendimiento

$$y = -\frac{1}{40}[70]^2 + \frac{7}{2}[70] = [122,5]$$

Muy bien! El máximo rendimiento corresponde a 122,5

Figura 25: Diseño de pregunta 3 del Ejercicio 4.

4.3.5 Diseño de resolución paso a paso para el Ejercicio 5

La pregunta uno del Ejercicio 5 corresponde a una pregunta de comprensión del enunciado. Para abordarla, se ha establecido un único paso de selección. El diseño de esta pregunta, junto con su respectivo paso, se puede apreciar en la Figura 26.

1. Sin considerar los precios, ¿qué tipo de árbol usarías si quisieras tener más árboles en esta plantación?

Paso 1: Selecciona la alternativa correcta

1. Eucalipto

2. Pino radiata

Figura 26: Diseño de pregunta 1 del Ejercicio 5.

Para abordar la pregunta dos del ejercicio 5, que consiste en plantear un sistema de ecuaciones de 2x2, se ha decidido proporcionar un ejemplo relacionado con el modelado de la primera ecuación, que corresponde a la ecuación de áreas.

En este enfoque, se le entregará al estudiante un texto informativo que definirá las variables a utilizar en el problema. Además, se ha definido el paso 1, denominado "Identificar áreas", donde se le pedirá al estudiante que identifique la expresión correspondiente a las áreas necesarias para cada tipo de árbol. Una vez que el estudiante complete este primer paso, se le proporcionará la ecuación uno, que representa la ecuación de áreas, y se le solicitará que defina la segunda ecuación.

En el paso 2, el estudiante deberá completar la ecuación correspondiente a los costos asociados a la plantación de cada tipo de árbol. Este proceso permitirá al estudiante relacionar las variables y los costos correspondientes a cada tipo de árbol en la ecuación.

La Figura 27 representa el diseño de estos dos primeros pasos, donde se muestran los textos informativos, los recuadros para identificar las áreas y los costos, y las ecuaciones correspondientes.

Considerando que cada pino cuesta más que cada eucalipto, tal vez no podríamos plantar solo pinos si tenemos una restricción de presupuesto.

2. Si contamos con un presupuesto de \$640.000, ¿cuantos árboles de cada tipo se deben plantar para maximizar el espacio?

Para responder esta pregunta primero definimos las variables a usar:

x: cantidad de eucaliptos a plantar
y: cantidad de pinos a plantar

Paso 1: Identificar áreas

El área total de eucaliptos es $9x$ y el área total de pinos es $[7y]$

Muy bien! Eso quiere decir que la ecuación de áreas es:

$$\text{área}_{\text{eucalipto}} + \text{área}_{\text{pinos}} = 1 \text{ hectarea}$$

Por lo tanto, la ecuación de áreas queda:

$$\text{ec1: } 9x + 7y = 10000$$

Paso 2: Definir segunda ecuación de costos

$$\text{ec2: } [528x] + [544y] = 640000$$

Figura 27: Diseño de paso 1 y 2 de la pregunta 2 del Ejercicio 5.

Una vez definidas las ecuaciones, es necesario despejar una de las variables para poder resolver el sistema. Es por esto que se definieron los pasos 3 y 4.

En el paso 3, denominado "Despejar x de ec1", el estudiante deberá realizar las operaciones necesarias para despejar la variable "x" de la primera ecuación. Esto implica realizar cálculos y simplificaciones algebraicas para obtener la expresión de x en función de las otras variables.

En el paso 4, denominado "Sustituir exp1 en ec2", el estudiante deberá tomar la expresión obtenida previamente y reemplazarla en la segunda ecuación. Esto permitirá reducir el sistema a una única ecuación en función de la variable "y".

A continuación, en los pasos 5 y 6, se le pide al estudiante realizar operaciones adicionales para reducir y simplificar la expresión resultante, lo que permitirá obtener una solución más sencilla y clara.

La Figura 28 representa el diseño de estos pasos, mostrando los pasos numerados y proporcionando recuadros de texto para que el estudiante complete las operaciones y simplificaciones correspondientes.

Ya tenemos un sistema de dos ecuaciones, ahora puedes despejar una de las dos variables en función de la otra.

Paso 3: Despejar 'x' de ec1

$$\text{exp1: } x = \frac{[10000-7y]}{9}$$

Ya que tenemos una de las variables la podemos sustituir en la ec2

Paso 4: Sustituir exp1 en ec2

$$\text{exp2: } 528\left(\frac{10000-7y}{9}\right) + 544y = 640000$$

Ahora, para poder factorizar y despejar 'y' debemos sumar ambas expresiones

Paso 5: Sumar fracciones

$$\text{exp3 : } \frac{[528(10000-7y)+9*544y]}{9} = 640000$$

Paso 6: simplificar exp3

$$\text{exp4 : } \frac{5280000 + [1200]y}{9} = 640000$$

Figura 28: Diseño de los pasos 3 al 6 de la pregunta 2 del Ejercicio 5.

Una vez obtenida la expresión reducida, es necesario despejar la variable "y" para poder calcular su valor. Para ello, se ha definido el paso 7, denominado "Despejar y en exp4", en el cual el estudiante deberá realizar las operaciones necesarias para despejar la variable "y" en la expresión obtenida previamente.

Luego, en el paso 8, denominado "Calcular y", el estudiante deberá realizar las operaciones aritméticas necesarias para obtener el valor final de la variable "y".

Como recordatorio, se proporciona la expresión 1, que corresponde a la expresión que define a la variable "x", la cual es utilizada en el paso 9, denominado "Calcular x". En este paso, el estudiante deberá utilizar la expresión 1 junto con el valor calculado previamente de la variable "y" para obtener el valor de "x".

La Figura 29 representa esta dinámica, mostrando los pasos definidos.

Paso 7: Despejar 'y' en exp4

$$\text{exp5 : } y = \frac{1920000 - 1760000}{400}$$

Paso 8: Calcular 'y'

$$\text{exp6 : } y = [400]$$

Muy bien! Los pinos son 400. Ahora ¿Cuántos eucaliptos son?

Recuerda que la $\text{exp1: } x = \frac{[10000-7y]}{9}$

Paso 9: Calcular 'x'

$$\text{exp7 : } x = [800]$$

Muy bien! Por lo tanto, para maximizar el espacio de plantaciones, se deberá plantar 800 eucaliptos y 400 pinos.

Figura 29: Definición de pasos del 7 al 9 de la pregunta 2 del Ejercicio 5.

4.4 Diseño de interfaz de los ejercicios

En esta subsección, se desarrollarán *mockups* para comprender el diseño general de los ejercicios y cómo se desplegarán los distintos componentes en el tutor cognitivo. Los mockups permitirán visualizar y evaluar la apariencia y la interacción de los elementos en el proceso de resolución de los ejercicios.

La Figura 30 muestra un mockup del *layout* general de los ejercicios, en donde destacan los siguientes elementos:

- *Exercise title*: Texto que muestra el título del ejercicio
- *Exercise statement*: Texto que contiene el enunciado del ejercicio, el cual puede estar presente o ausente dependiendo el ejercicio.
- *Math expression*: Representa una expresión matemática relevante para el ejercicio, y puede estar presente o ausente.
- *Table / Image*: Es un componente que muestra una imagen o una tabla asociada al enunciado del ejercicio. Su presencia puede variar según el ejercicio.
- *Explanation*: Es un texto explicativo que se muestra antes de cada paso, y puede o no estar presente.
- *Component*: Componente que se utiliza para que el estudiante responda cada paso, este puede ser componente matemático, componente de selección o componente gráfico.
- *Alert*: Es un mensaje de retroalimentación que proporciona el tutor cognitivo cuando el estudiante ingresa una respuesta.
- Botón Aceptar: Permite al estudiante confirmar la respuesta ingresada para que sea evaluada.
- Botón Pista: Ofrece una pista o ayuda adicional al estudiante para ayudarlo a resolver el paso en caso de que lo necesite.

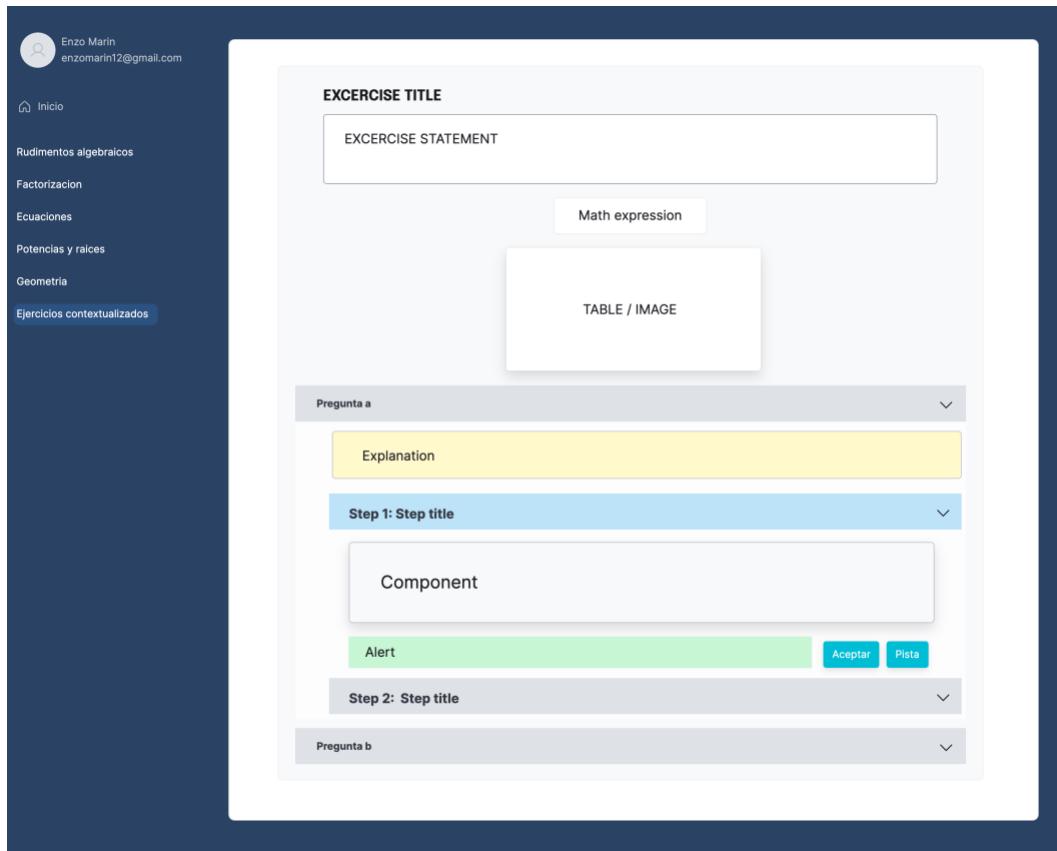


Figura 30: *Mockup* general del diseño de los ejercicios

La Figura 31 corresponde al diseño del primer paso de la primera pregunta del ejercicio 1. Este paso utiliza el componente de selección, lo que implica que se despliegan las alternativas definidas en su archivo JSON. En este caso, no se utiliza un botón "Aceptar" adicional, ya que cada una de las alternativas cuenta como un botón independiente.

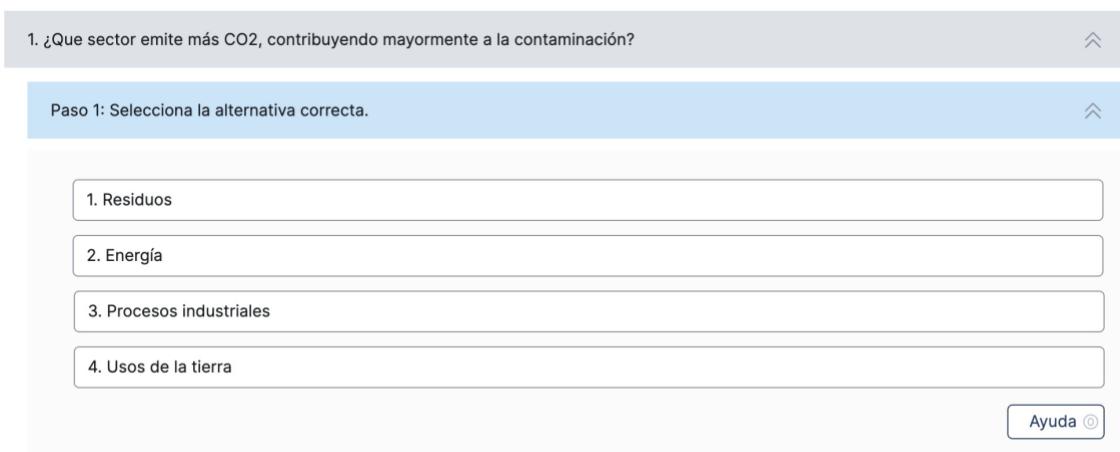


Figura 31: *Mockup* de la selección de alternativa.

La Figura 32 corresponde a la retroalimentación que el tutor proporcionará cuando el estudiante seleccione una alternativa incorrecta. En este caso, se marca en color rojo la alternativa seleccionada y se mostrará una alerta indicando que ha cometido un error. Esto permite al estudiante identificar de manera clara y visual que su respuesta no es la correcta. Por otro lado, si el estudiante selecciona la alternativa correcta, esta se marcará en verde y se desplegará una alerta indicando que ha seleccionado la opción correcta (ver Figura 33).

The mockup shows a user interface for a question about CO2 emissions. The question is: "1. ¿Qué sector emite más CO2, contribuyendo mayormente a la contaminación?". Below it, a blue bar says "Paso 1: Selecciona la alternativa correcta.". A list of four options follows:

- 1. Residuos
- 2. Energía
- 3. Procesos industriales** (highlighted in red)
- 4. Usos de la tierra

A red error message "⚠ Respuesta incorrecta!" is displayed below the list. In the bottom right corner, there is a "Ayuda" button.

Figura 32: *Mockup* de la selección de alternativa con respuesta errónea.

The mockup shows a user interface for a question about CO2 emissions. The question is: "1. ¿Qué sector emite más CO2, contribuyendo mayormente a la contaminación?". Below it, a green bar says "Paso 1: Selecciona la alternativa correcta.". A list of four options follows:

- 1. Residuos
- 2. Energía** (highlighted in green)
- 3. Procesos industriales
- 4. Usos de la tierra

A green success message "🕒 Respuesta correcta: El sector de energía es el sector que más contamina" is displayed below the list. In the bottom right corner, there is a "Ayuda" button.

Figura 33: *Mockup* de la selección de alternativa con respuesta correcta.

La Figura 34 muestra el diseño del primer paso de la tercera pregunta del ejercicio 1, utilizando el componente matemático. En este paso, se proporcionará ayuda al estudiante

entregando la respuesta de manera parcial, presentando recuadros vacíos que el estudiante debe completar para responder el paso.

Además, la Figura 34 también muestra la alerta correspondiente que se mostrará cuando el estudiante presione el botón "Aceptar" sin completar todos los recuadros requeridos. Esta alerta tiene el propósito de recordar al estudiante que debe llenar todos los campos antes de enviar la respuesta.

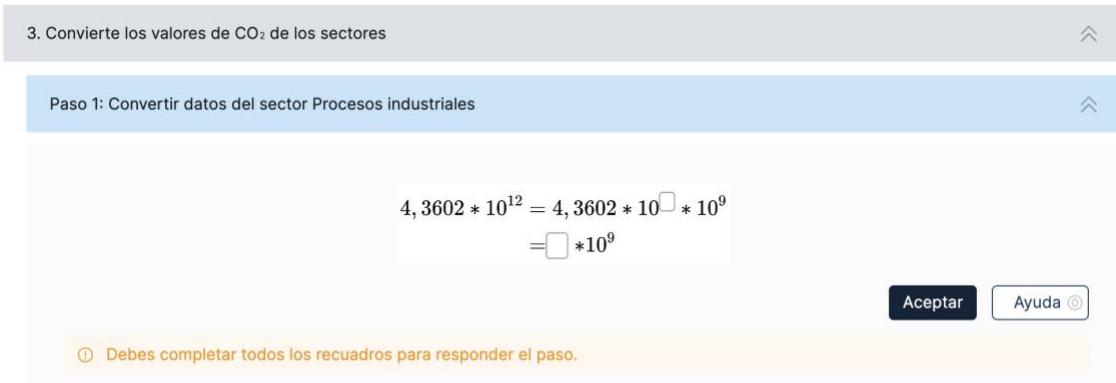


Figura 34: *Mockup* componente matemático con respuesta vacía.

La Figura 35 muestra un ejemplo del despliegue de los *hints* (ayudas/pistas) que se le proporcionará al estudiante cuando este lo solicite. Este *hint* incluye un título, un texto explicativo, un marcador que indica el *hint* actual en relación al total y dos botones que permiten al estudiante desplazarse entre los *hints* disponibles.

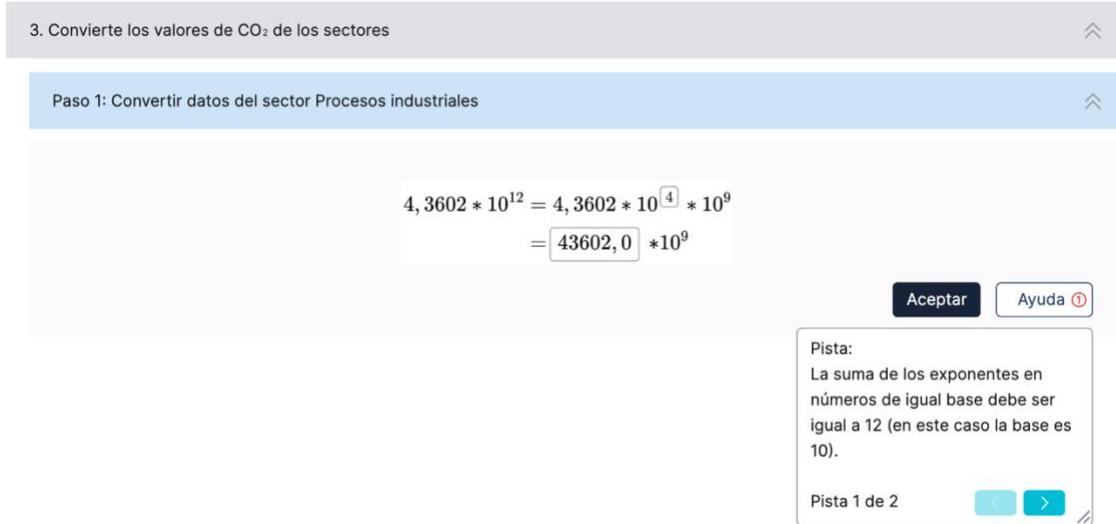


Figura 35: Mockup del despliegue de un *hint*.

Por último, la Figura 36 muestra el *mockup* del cuarto paso de la segunda pregunta del ejercicio 3, utilizando el componente gráfico. En este paso, se presentará al estudiante un gráfico con los datos y la recta correspondientes para realizar el ajuste lineal. Además, se le proporcionará la ecuación de la recta junto con un control deslizante (*slider*) que permitirá al estudiante variar el valor de la pendiente “m” para realizar el ajuste adecuado.

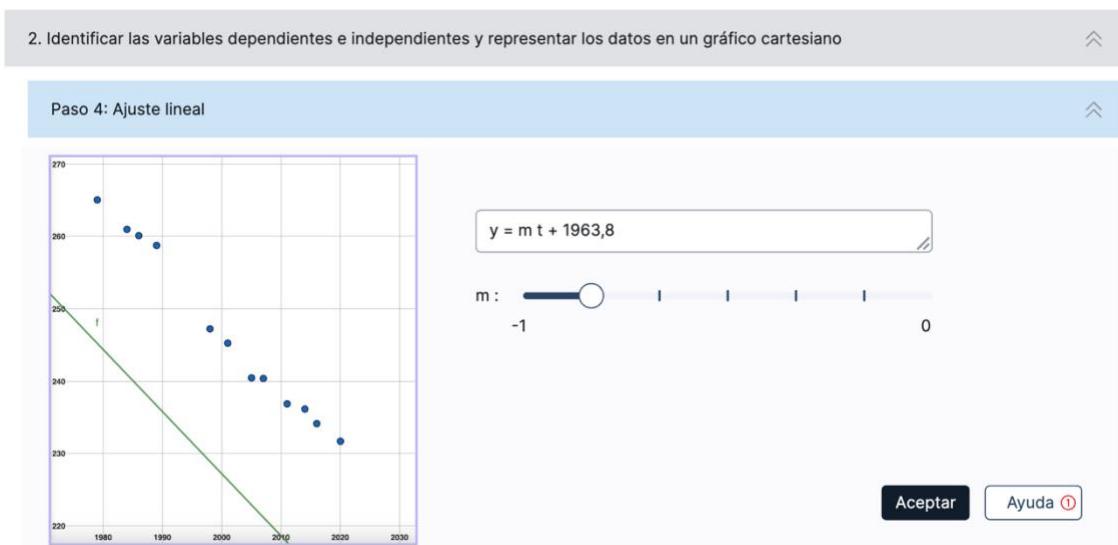


Figura 36: *Mockup* del componente gráfico.

4.5 Modelamiento de ejercicios

4.5.1 Definición de habilidades de los ejercicios

En esta subsección, se llevará a cabo el etiquetado de los componentes de conocimiento (KC, su sigla en inglés) para cada paso de los ejercicios. Estos KCs representan conceptos o elementos de aprendizaje específicos y se utilizan en el tutor para realizar un seguimiento de los conocimientos del estudiante a lo largo del tiempo.

Cada paso de las preguntas de un ejercicio será etiquetado con cero o más KCs, lo que permitirá identificar los conocimientos específicos que el estudiante debe dominar en cada etapa del ejercicio. Algunos ejemplos de estos KCs, ya existentes en la plataforma, son "*ReducExp*: Realizar operaciones algebraicas para simplificar la ecuación" y "*DespVarEcu*: Despejar variable en una ecuación", entre otros.

El proceso de etiquetado se realizó en colaboración con los profesores patrocinante y co-patrocinante. Durante las reuniones conjuntas, se analizaron detalladamente los pasos de los ejercicios y se identificaron las habilidades y conocimientos necesarios para resolver cada paso. Posteriormente, se asignaron los KC correspondientes a cada paso, asegurando una alineación adecuada entre los pasos y los objetivos de aprendizaje establecidos.

Gracias al etiquetado de los KC, el tutor cognitivo puede identificar las fortalezas y debilidades del estudiante en relación a los conceptos y habilidades abordados en los ejercicios. Esto permite ofrecer una guía más precisa y enfocada en el proceso de aprendizaje, brindando recomendaciones específicas para fortalecer las áreas en las que el estudiante necesita mejorar. Esta información será de gran utilidad para el análisis y la evaluación de los avances del estudiante, los cuales se abordan en trabajos de título paralelos a éste.

Es relevante mencionar que, durante el proceso de etiquetado, algunos pasos no se etiquetaron con KCs. La razón detrás de esta decisión es que se determinó que la habilidad practicada en esos pasos, aunque es relevante para avanzar en el ejercicio, no es una habilidad que se deseé modelar dentro del espacio de habilidades manejado por el tutor cognitivo. Además, agregar estas habilidades como KCs implicaría la necesidad de crear varios ejercicios adicionales para practicar esas habilidades.

En las Tablas 8, 9, 10, 12 y 13 se encuentran los resultados del etiquetado de los KC para cada ejercicio.

Tabla 8: KCs del Ejercicio 1.

Pregunta	Paso	Título del paso	Descripción de KC
1	1	Selecciona la alternativa correcta	<ul style="list-style-type: none"> Ordenar números expresados en notación científica (OrdNC).
2	1	Selecciona la alternativa correcta	<ul style="list-style-type: none"> Ordenar números expresados en notación científica (OrdNC).
3	1	Convertir datos del sector Procesos industriales	<ul style="list-style-type: none"> Sumar y restar exponentes en misma base (SumResExp). Multiplicación de números (Mult). Multiplicación decimal base 10 (MultDecBase10).
	2	Convertir datos del sector Usos de la tierra	<ul style="list-style-type: none"> Sumar y restar exponentes en misma base (SumResExp). Multiplicación de números (Mult). Multiplicación decimal base 10 (MultDecBase10).
	3	Convertir datos del sector Residuos	<ul style="list-style-type: none"> Sumar y restar exponentes en misma base (SumResExp). Multiplicación de números (Mult). Multiplicación decimal base 10 (MultDecBase10).
4	1	Calcular total de emisiones	<ul style="list-style-type: none"> Suma de números decimales (SumDec).
5	1	Calcular total de absorciones	<ul style="list-style-type: none"> Sin etiquetar.
6	1	Calcular balance	<ul style="list-style-type: none"> Suma de números decimales (SumDec).
7	1	Calcular proporción	<ul style="list-style-type: none"> Cancelación términos semejantes (CancTermSem).
	2	Responder	<ul style="list-style-type: none"> Notación porcentaje (NotPorc).

Tabla 9: KCs del Ejercicio 2.

Pregunta	Paso	Título del paso	Descripción de KCs
1	1	Calcular resistencia en serie	<ul style="list-style-type: none"> • Sin etiquetar.
2	1	Selecciona la alternativa correcta	<ul style="list-style-type: none"> • Sin etiquetar.
3	1	Reemplaza valores de resistencias	<ul style="list-style-type: none"> • Sustituir en formula (SusFor).
	2	Suma las fracciones del denominador	<ul style="list-style-type: none"> • Sumar y restar fracciones (SRFrac).
	3	Reduce la expresión a una fracción simple	<ul style="list-style-type: none"> • Expresar división de fracciones como multiplicación (DivFrac).
4	1	Modelar ecuación	<ul style="list-style-type: none"> • Expresar algebraicamente (ExpA).
	2	Suma las fracciones del denominador	<ul style="list-style-type: none"> • Sumar y restar fracciones (SRFrac).
	3	Reduce la expresión a una fracción simple	<ul style="list-style-type: none"> • Expresar división de fracciones como multiplicación (DivFrac).
	4	Despeja x	<ul style="list-style-type: none"> • Despejar variable en una ecuación (DespVarEcu).

Tabla 10: KCs del Ejercicio 3.

Pregunta	Paso	Título del paso	Descripción de KCs
1	1	Selecciona la alternativa verdadera.	<ul style="list-style-type: none"> • Sin etiquetar.
2	1	Selecciona la alternativa verdadera.	<ul style="list-style-type: none"> • Identificar variable dependiente e independiente (IdVar).
	2	Selecciona la alternativa correcta.	<ul style="list-style-type: none"> • Identificar variable dependiente e independiente (IdVar).
	3	Identificar punto en el grafico	<ul style="list-style-type: none"> • Identificar punto en plano cartesiano (IdPtGraf).
	4	Realiza el ajuste lineal	<ul style="list-style-type: none"> • Identificar punto en plano cartesiano (IdPtGraf). • Comprensión pendiente de la recta (CompPendRec). • Expresar algebraicamente (ExpA).
	5	Selecciona la alternativa correcta	<ul style="list-style-type: none"> • Sin etiquetar.

Tabla 11: (Continuación)

Pregunta	Paso	Título del paso	Descripción de KCs
3	1	Reemplaza el valor de la variable t	<ul style="list-style-type: none"> Sustituir en formula (SusFor). Identificar variable dependiente e independiente (IdVar).
	2	Resuelve para 'y'	<ul style="list-style-type: none"> Aritmética de números (AN).
4	1	Plantear la ecuación lineal	<ul style="list-style-type: none"> Expresar algebraicamente (ExpA).
	2	Restar valor constante	<ul style="list-style-type: none"> Realizar operaciones algebraicas para simplificar la ecuación (ReducExp).
	3	Despejar variable del tiempo	<ul style="list-style-type: none"> Despejar variable en una ecuación (DespVarEcu).

Tabla 12: KCs del Ejercicio 4.

Pregunta	Paso	Título del paso	Descripción de KCs
1	1	Calcular rendimiento para la velocidad de 20 km/h	<ul style="list-style-type: none"> Aritmética de números (AN). Sustituir en formula (SusFor).
	2	Calcular rendimiento para la velocidad de 60 km/h	<ul style="list-style-type: none"> Aritmética de números (AN). Sustituir en formula (SusFor).
	3	Calcular rendimiento para la velocidad de 100 km/h	<ul style="list-style-type: none"> Aritmética de números (AN). Sustituir en formula (SusFor).
2	1	¿En qué punto de la parábola podríamos encontrar su máximo rendimiento?	<ul style="list-style-type: none"> Identificar punto en plano cartesiano (IdPtGraf).
	2	Identificar coeficientes	<ul style="list-style-type: none"> Identificar coeficientes (IdeCoef).
	3	Calcular coordenada x del vértice	<ul style="list-style-type: none"> Sustituir en formula (SusFor). Aritmética de números (AN). Expresar división de fracciones como multiplicación (DivFrac).
3	1	Calcular valor del máximo rendimiento	<ul style="list-style-type: none"> Sustituir en formula (SusFor). Aritmética de números (AN).

Tabla 13: KCs del Ejercicio 5.

Pregunta	Paso	Título del paso	Descripción de KCs
1	1	Selecciona la alternativa correcta.	<ul style="list-style-type: none"> • Sin etiquetar.
2	1	Identificar áreas	<ul style="list-style-type: none"> • Sin etiquetar.
	2	Definir segunda ecuación de costos	<ul style="list-style-type: none"> • Plantear ecuación desde un contexto (PlanEcuContex). • Expresar algebraicamente (ExpA).
	3	Despejar ‘x’ de ec1	<ul style="list-style-type: none"> • Despejar variable en una ecuación (DespVarEcu).
	4	Sustituir exp1 en ec2	<ul style="list-style-type: none"> • Método de sustitución (ReempVarEcu).
	5	Sumar fracciones	<ul style="list-style-type: none"> • Aritmética de números (AN).
	6	Simplificar exp3	<ul style="list-style-type: none"> • Realizar operaciones algebraicas para simplificar la ecuación (ReducExp).
	7	Despejar ‘y’ en exp4	<ul style="list-style-type: none"> • Despejar variable en una ecuación (DespVarEcu).
	8	Calcular ‘y’	<ul style="list-style-type: none"> • Aritmética de números (AN). • Despejar variable en una ecuación (DespVarEcu).
	9	Calcular ‘x’	<ul style="list-style-type: none"> • Método de sustitución (ReempVarEcu). • Aritmética de números (AN).

4.5.2 Componentes utilizados

A continuación se listan los componentes utilizados para la resolución de los pasos de cada ejercicio. Los detalles de sus atributos se describen en la siguiente subsección y su implementación se detallan en el capítulo 5.

- *mathComponent*: Este componente permite la visualización de un campo matemático en el que el estudiante puede ingresar o editar expresiones matemáticas y comprobar su respuesta. Adicionalmente, proporciona un teclado virtual para facilitar el ingreso de símbolos matemáticos. (Github – enzomarin/TutorCognitivo, s.f)
- *selectionComponent*: Este componente permite al estudiante responder una pregunta mediante la selección de una alternativa entre varias proporcionadas. (Github – enzomarin/TutorCognitivo, s.f)
- *graphComponent*: Componente que se utiliza para la integración de los componentes gráficos implementados. (Github- enzomarin/TutorCognitivo, s.f)
- *selectPoint*: Este componente permite la selección de un punto en un plano cartesiano entre varios proporcionados. (Github – enzomarin/TutorCognitivo, s.f)

- *linearFit*: Este componente permite al estudiante realizar un ajuste lineal, interactuando con la gráfica y realizando el ajuste mediante la variación de los parámetros de la función lineal mediante controles deslizantes (*sliders*) (Github – enzomarin/TutorCognitivo, s.f)

4.5.3 Definición de la estructura de los ejercicios

En esta etapa del modelado de los ejercicios, se llevó a cabo la identificación de similitudes entre ellos, con el propósito de establecer una estructura general que permitiera su modelado de manera consistente. Para lograr este objetivo, se definió una estructura JSON que representa el modelo general de datos para los ejercicios. A partir de esta estructura se detallaron los atributos más relevantes para la definición de un ejercicio.

La Tabla 14 presenta los atributos que conforman la estructura general, y que sirven como base para el modelado y desarrollo de cada ejercicio.

Tabla 14: Definición de atributos en la estructura general de los ejercicios

Campo	Descripción	Tipo
<i>code</i>	Identificador único del ejercicio.	<i>String</i>
<i>type</i>	Texto que indica el tipo del ejercicio.	<i>String</i>
<i>presentation</i>	Objeto que contiene los atributos necesarios para la presentación del ejercicio.	<i>Object</i>
<i>learningObjectives</i>	Objeto que contiene los atributos necesarios para la presentación de los objetivos de aprendizajes del ejercicio.	<i>Object</i>
<i>statement</i>	Texto del enunciado del ejercicio.	<i>String</i>
<i>mathExpression</i>	Expresión matemática asociada al enunciado del ejercicio (opcional).	<i>String / undefined</i>
<i>img</i>	url de la dirección a la imagen asociada al enunciado del problema (opcional).	<i>String / undefined</i>
<i>table</i>	Objeto con los atributos necesarios para definir la tabla asociada al enunciado del problema (opcional).	<i>object / undefined</i>
<i>text</i>	Texto previo a las preguntas.(opcional)	<i>String / undefined</i>
<i>questions</i>	Arreglo de objetos, donde cada objeto es una pregunta y contiene los atributos relacionados a dicha pregunta.	<i>Object[]</i>

En las Tablas 15 y 16 se definen los atributos para los campos “*presentation*” y “*learningObjectives*” respectivamente.

Tabla 15: Definición de los atributos para el campo *presentation*

Campo	Descripción	Tipo
<i>title</i>	Título del ejercicio	<i>String</i>
<i>urlImg</i>	url con la dirección a la imagen de presentación del ejercicio	<i>String</i>

Tabla 16: Definición de los atributos para el campo *learningObjectives*

Campo	Descripción	Tipo
<i>title</i>	Título de los objetivos de aprendizajes	<i>String</i>
<i>text</i>	Texto donde se explica brevemente lo que se aprenderá al resolver el ejercicio	<i>String</i>
<i>listObj</i>	Arreglo de strings que contiene la lista de objetivos de aprendizajes	<i>String[]</i>

La Figura 37 muestra la estructura JSON general para los ejercicios. Esta permite una representación coherente y consistente de los ejercicios, facilitando su implementación en el tutor cognitivo. Además, brinda flexibilidad para incorporar diferentes tipos de ejercicios y adaptarlos a las necesidades específicas de los estudiantes. Esto mediante la definición del tipo de algunos atributos como “*undefined*”, lo que quiere decir que no son obligatorios al crear un nuevo ejercicio.

```
{
  "code": "codigo del ejercicio",
  "type": "wordProblem",
  "presentation": {
    "title": "Título del ejercicio",
    "urlImg": "url de la imagen de presentación"
  },
  "learningObjectives": {
    "title": "¿Qué aprendizajes son pretendidos al desarrollar este ejercicio?",
    "text": "Explicación sobre lo que se va a aprender al resolver este ejercicio",
    "listObj": ["obj1", "obj2", "obj3"]
  },
  "meta": {},
  "statement": "Enunciado del ejercicio",
  "mathExpression": "Expresión matemática asociada al ejercicio",
  "img": "url de la imagen del ejercicio",
  "table": {
    "header": [ // ARREGLO DE ENCABEZADOS ],
    "rows": [ // ARREGLO DE FILAS ],
    "tableCaption": "Título de la tabla "
  },
  "text": "texto explicativo previo a las preguntas",
  "questions": [
    { // PREGUNTA 1 },
    { // PREGUNTA 2 },
    { // PREGUNTA N }
  ]
}
```

Figura 37: JSON de la estructura general de los ejercicios.

En la Tabla 17 se encuentran definidos los atributos necesarios para establecer la estructura de una tabla. Estos atributos proporcionan información clave sobre cómo se organizarán y presentarán los datos dentro de la tabla.

Tabla 17: Definición de los atributos para el campo "*table*"

Campo	Descripción	Tipo
<i>header</i>	Arreglo de objetos, donde cada objeto es una columna y tiene los atributos “align” y “value”, los cuales corresponden a la alineación del texto y el valor (nombre de la columna) respectivamente	<i>Object[]</i>
<i>rows</i>	Arreglo de objetos, donde cada objeto es una fila y tiene un atributo “data” el cual contiene un arreglo con el valor de cada celda.	<i>Object[]</i>
<i>alignRows</i>	Propiedad que permite alinear las filas en una tabla, esta puede tomar los valores: “left”, “right”, “center”, “justify”, “end” o “start”	<i>String</i>
<i>tableCaption</i>	Título de la tabla	<i>String</i>

Estos atributos se pueden observar en la estructura JSON presentada en la Figura 38, la cual representa la definición de la tabla asociada al enunciado del Ejercicio 1.



```

TutorIntegrado - exercise1.json

1 "table": {
2   "header": [
3     {
4       "align": "center",
5       "value": "Sector"
6     },
7     {
8       "align": "center",
9       "value": "Co2 Neto"
10    }],
11   "rows": [
12     {
13       "data": ["Energía", "$8,18231*10^{13}$"]
14     },
15     {
16       "data": ["Procesos Industriales", "$4,3602*10^{12}$"]
17     },
18     {
19       "data": ["Usos de la tierra (Agricultura, Silvicultura y otros )", "$-3.9915*10^{13}$"]
20     },
21     {
22       "data": ["Residuos", "$3*10^8$"]
23     }
24   ],
25   "alignRows": "left",
26   "tableCaption": "Tabla: Niveles de CO2 por sector en Chile durante el año 2013 "
27 }

```

Figura 38: Estructura JSON de la tabla del Ejercicio 1.

En la Tabla 18 se encuentran definidos los atributos necesarios para la estructura de las preguntas asociadas a un ejercicio. Estos atributos permiten especificar los detalles de cada pregunta y su presentación en el tutor cognitivo.

Tabla 18: Definición de atributos para la estructura de las preguntas.

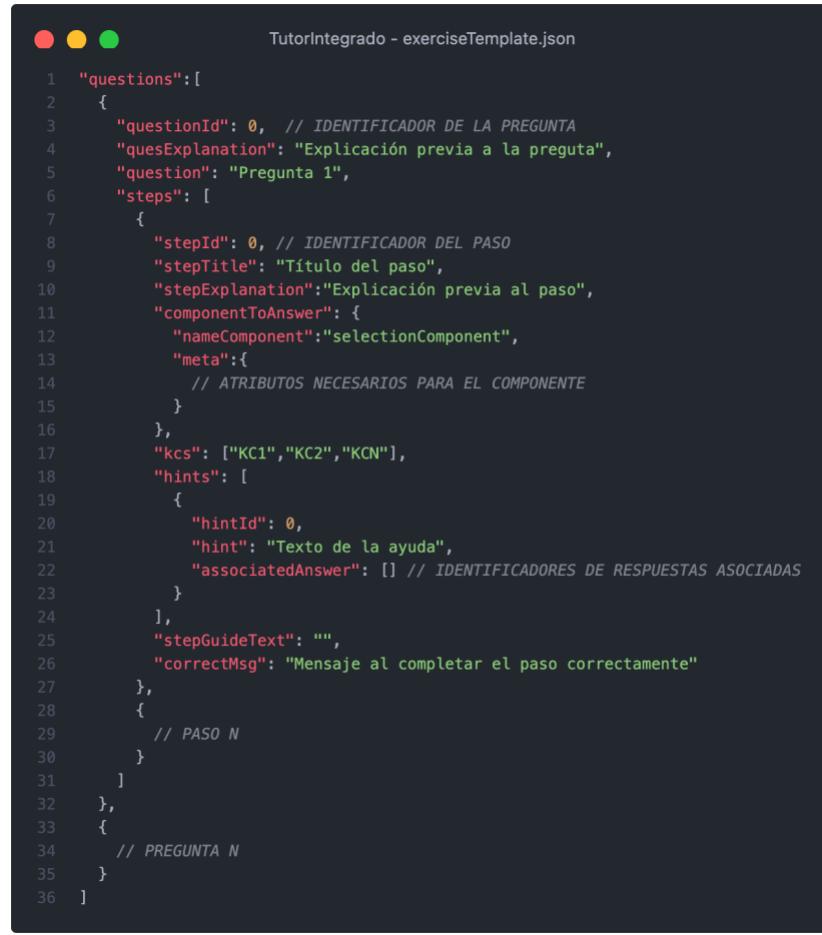
Campo	Descripción	Tipo
<i>questionId</i>	Identificador único de cada pregunta.	<i>Number</i>
<i>question</i>	Texto de la pregunta.	<i>String</i>
<i>quesExplanation</i>	Texto explicativo previo a una pregunta.	<i>String / undefined</i>
<i>steps</i>	Arreglo de objetos, donde cada objeto es un paso y contiene los atributos relacionados a los pasos del ejercicio.	<i>Object[]</i>

En la Tabla 19 se definen los atributos que están asociados a cada paso de una pregunta. Estos atributos permiten especificar las instrucciones y componentes necesarios para la resolución paso a paso de cada pregunta.

Tabla 19: Definición de los atributos de la estructura para los pasos de una pregunta.

Campo	Descripción	Tipo
<i>stepId</i>	Identificador único de cada paso.	<i>Number</i>
<i>stepTitle</i>	Título del paso.	<i>String</i>
<i>stepExplanation</i>	Texto explicativo previo a cada paso.	<i>String / undefined</i>
<i>componentToAnswer</i>	Objeto donde se define el componente que se utiliza para responder cada paso.	<i>Object</i>
<i>kcs</i>	Arreglo de kc asociados a la resolución del paso.	<i>String[]</i>
<i>hints</i>	Arreglo de objetos, donde cada objeto es un <i>hint</i> y contienen el identificador del <i>hint</i> , el texto a mostrar como ayuda y un arreglo de números que asocia el <i>hint</i> a respuestas específicas.	<i>Object[]</i>
<i>stepGuideText</i>	Texto posterior al título de cada paso.	<i>String / undefined</i>
<i>correctMsg</i>	Mensaje que se muestra en la alerta cuando el estudiante ha respondido el paso correctamente.	<i>String / undefined</i>

En la Figura 39 se presenta la estructura JSON que representa las preguntas asociadas a un ejercicio. Esta estructura refleja los atributos definidos en la Tabla 18 y en la Tabla 19, los cuales son necesarios para proporcionar la información y las instrucciones necesarias para cada pregunta y sus respectivos pasos.



```

1 "questions": [
2   {
3     "questionId": 0, // IDENTIFICADOR DE LA PREGUNTA
4     "questExplanation": "Explicación previa a la pregunta",
5     "question": "Pregunta 1",
6     "steps": [
7       {
8         "stepId": 0, // IDENTIFICADOR DEL PASO
9         "stepTitle": "Título del paso",
10        "stepExplanation": "Explicación previa al paso",
11        "componentToAnswer": {
12          "nameComponent": "selectionComponent",
13          "meta": {
14            // ATRIBUTOS NECESARIOS PARA EL COMPONENTE
15          }
16        },
17        "kcs": ["KC1", "KC2", "KCN"],
18        "hints": [
19          {
20            "hintId": 0,
21            "hint": "Texto de la ayuda",
22            "associatedAnswer": [] // IDENTIFICADORES DE RESPUESTAS ASOCIADAS
23          }
24        ],
25        "stepGuideText": "",
26        "correctMsg": "Mensaje al completar el paso correctamente"
27      },
28      {
29        // PASO N
30      }
31    ]
32  },
33  {
34    // PREGUNTA N
35  }
36 ]

```

Figura 39: Estructura JSON del arreglo de preguntas de un ejercicio

En relación a la definición de los atributos para los componentes utilizados en los pasos, se enfrentó a la dificultad de que cada componente requiere diferentes atributos. Para abordar esta situación, se decidió implementar un atributo llamado “*meta*” (ver Figura 39) que varía en función del componente utilizado. Este atributo se define dentro del atributo “*componentToAnswer*”, el cual es un objeto que contiene los atributos “*meta*” y “*nameComponent*”.

El atributo “*nameComponent*” se utiliza para identificar el tipo de componente a utilizar y puede tomar los valores “*mathComponent*”, “*selectionComponent*” o “*graphComponent*”, dependiendo de las necesidades del paso.

El atributo “*meta*” proporciona flexibilidad al permitir la definición de atributos específicos para cada tipo de componente. Esto facilita la adaptación de los ejercicios a diferentes contextos y necesidades, ya que cada tipo de componente puede tener atributos personalizados asociados a él.

Al utilizar esta estrategia, se logra una mayor modularidad, escalabilidad y flexibilidad en la definición de los componentes de los ejercicios. Además, se crea la posibilidad de

agregar nuevos componentes en el futuro según las necesidades que surjan, sin afectar la estructura general de los ejercicios.

La Tabla 20 presenta la definición de los atributos necesarios para el componente "*mathComponent*". Este componente se utiliza en los pasos de las preguntas donde se requiere una respuesta matemática por parte del estudiante.

Tabla 20: Definición del campo *meta* para el componente *mathComponent*.

Campo	Descripción	Tipo
<i>readOnly</i>	Booleano que define si el componente es estático.	<i>Boolean</i>
<i>expression</i>	Texto en formato <i>Latex</i> que contiene la expresión a completar por el estudiante.	<i>String</i>
<i>answers</i>	Arreglo de objetos, donde cada objeto es una respuesta y contiene los atributos “ <i>id</i> ”, “ <i>placeholderId</i> ” y “ <i>value</i> ”	<i>Object[]</i>
<i>idCorrectAnswers</i>	Arreglo de números para identificar las respuestas correctas.	<i>Number[]</i>

En la definición de los atributos para el componente "*mathComponent*", es importante destacar el atributo "*readOnly*" que determina si el componente es de solo lectura o editable. Si el valor de este atributo es *false* (falso), todo lo que se defina en el atributo "*expression*" será editable, lo que permite dejar el atributo "*expression*" en blanco para que el estudiante pueda ingresar la respuesta completa.

Por otro lado, si el atributo "*readOnly*" es *true* (verdadero), todo lo que se defina en el atributo "*expression*" será estático, lo que implica que no se podrá editar. Sin embargo, es posible definir campos editables dentro del texto estático utilizando la sintaxis "*\$placeholder [a] {}\$*", donde "a" corresponde al identificador del *placeholder*. Esta sintaxis permite proporcionar respuestas parciales con ciertos elementos que el estudiante puede completar.

Esta funcionalidad brinda flexibilidad al permitir la combinación de elementos estáticos y editables en el componente "*mathComponent*", lo que facilita la presentación de respuestas parciales y guías para el estudiante dentro del contexto del ejercicio.

La Figura 40 muestra un ejemplo de la estructura JSON de este componente.


TutorIntegrado - exerciseTemplate.json

```

1 "componentToAnswer": {
2   "nameComponent": "mathComponent",
3   "meta": {
4     "expression": "\placeholder[a]{}",
5     "answers": {
6       "id": 1,
7       "placeholderId": "a",
8       "value": "x"
9     },
10    "idCorrectAnswers": [1]
11  }
12 },

```

Figura 40: Estructura JSON del componente "*mathComponent*".

La Tabla 21 presenta los atributos necesarios para definir el componente "*selectionComponent*". Estos atributos se puede visualizar en la estructura JSON de la Figura 41.

Tabla 21: Definición del campo *meta* para el componente *selectionComponent*.

Campo	Descripción	Tipo
<i>answers</i>	Arreglo de objetos, donde cada objeto es una alternativa y contiene los atributos “ <i>id</i> ” y “ <i>value</i> ”	<i>Object[]</i>
<i>idCorrectAnswers</i>	Identificador de la alternativa correcta.	<i>Number</i>


TutorIntegrado - exerciseTemplate.json

```

1 "componentToAnswer": {
2   "nameComponent": "selectionComponent",
3   "meta": {
4     "answers": [
5       {
6         "id": 0,
7         "value": "Alternativa 1"
8       },
9       {
10         "id": 1,
11         "value": "Alternativa 2"
12       }
13     ],
14     "idCorrectAnswers": 0
15   }
16 },

```

Figura 41: Estructura JSON del componente "*selectionComponent*"

La Tabla 22 representa los atributos necesarios para el despliegue del componente *graphComponent*

Tabla 22: Definición del campo *meta* para el componente *graphComponent*

Campo	Descripción	Tipo
<i>component</i>	<i>Enum</i> que define el componente grafico a utilizar, este puede ser “ <i>selectPoint</i> ” o “ <i>linearFit</i> ”.	<i>Enum</i>
<i>metaComponent</i>	Objeto que contiene los atributos necesarios para el despliegue de cada componente.	<i>Object</i>

En el componente “*graphComponent*”, el campo “*metaComponent*” es un campo dinámico que puede tener distintos atributos dependiendo del tipo de componente gráfico utilizado. La estructura de los atributos para “*metaComponent*” se especifica en las Tablas 23 y 24 para los componentes de “*selectPoint*” y “*linearFit*” respectivamente.

Tabla 23: Definición del campo *metaComponent* para el componente *selectPoint*.

Campo	Descripción	Tipo
<i>data</i>	Arreglo que contiene información sobre los puntos que se deben desplegar en el plano cartesiano.	<i>Point[]</i>
<i>correctPoint</i>	Arreglo que contiene las coordenadas (x, y) del punto correcto que el estudiante debe identificar.	<i>Number[]</i>
<i>graphSettings</i>	Objeto que contiene las configuraciones específicas para el gráfico cartesiano interactivo.	<i>Object</i>

Tabla 24: Definición del campo *metaComponent* para el componente *linearFit*.

Campo	Descripción	Tipo
<i>data</i>	Arreglo que contiene información sobre los puntos que se deben desplegar en el gráfico cartesiano.	<i>Point[]</i>
<i>linearFunction</i>	Objeto que contiene los parámetros “m” y “b” de la función lineal que se ajustará en el gráfico. Cada parámetro puede ser un “ <i>value</i> ” o un “ <i>slider</i> ”	<i>Object</i>
<i>correctAnswer</i>	Objeto que contiene los rangos de valores correctos para los parámetros “m” y “b” del ajuste lineal, en caso de que estos sean <i>sliders</i> .	<i>Object</i>
<i>graphSettings</i>	Objeto que contiene las configuraciones específicas para el gráfico cartesiano interactivo.	<i>Object</i>

La Figura 42 muestra la estructura JSON utilizada para representar el paso 3 de la segunda pregunta del ejercicio 3, el cual hace uso del componente gráfico "selectPoint". En este ejemplo, se puede visualizar la definición de los atributos mencionados en la Tabla 23. Uno de los atributos relevantes es "data", que representa un arreglo de puntos. Cada punto puede ser definido solo con su coordenada, o se pueden personalizar mediante los siguientes atributos adicionales:

- *coord*: Es un arreglo que contiene las coordenadas (x, y) del punto en el plano cartesiano.
- *name*: Es un atributo opcional que puede contener un nombre o etiqueta para el punto, lo que ayuda a identificarlo visualmente.
- *color*: Es un atributo opcional que especifica el color del punto en el gráfico.
- *isStatic*: Es un atributo opcional que indica si el punto es estático o interactivo. Un punto estático no se puede mover en el gráfico, mientras que un punto interactivo permite que los estudiantes puedan moverlo.
- *face*: Es un atributo opcional que define la forma visual del punto en el gráfico. Esta forma puede ser una cruz, un círculo, un cuadrado o un signo de suma, y esto ayuda a diferenciar visualmente los puntos.



```

TutorIntegrado - exercise3.json
1  "componentToAnswer": {
2    "nameComponent": "graphComponent",
3    "meta": {
4      "component": "selectPoint",
5      "metaComponent": {
6        "data": [
7          {
8            "coord": [1979, 265.05],
9            "name": "Punto1",
10           "color": "green",
11           "isStatic": false,
12           "face": "cross"
13         },
14         {"coord": [1998, 247.23]},
15         {"coord": [2016, 234.14]}
16       ],
17       "correctPoint": [1998, 247.23],
18       "graphSettings": {
19         // CONFIGURACIÓN DEL GRÁFICO
20       }
21     }
22   }
23 }

```

Figura 42: Ejemplo de estructura JSON del componente *selectPoint*.

Continuando con la descripción, la Figura 43 presenta la estructura JSON utilizada para representar el paso 4 de la segunda pregunta del Ejercicio 3, el cual emplea el componente gráfico "linearFit". Al igual que en el componente "selectPoint", el componente

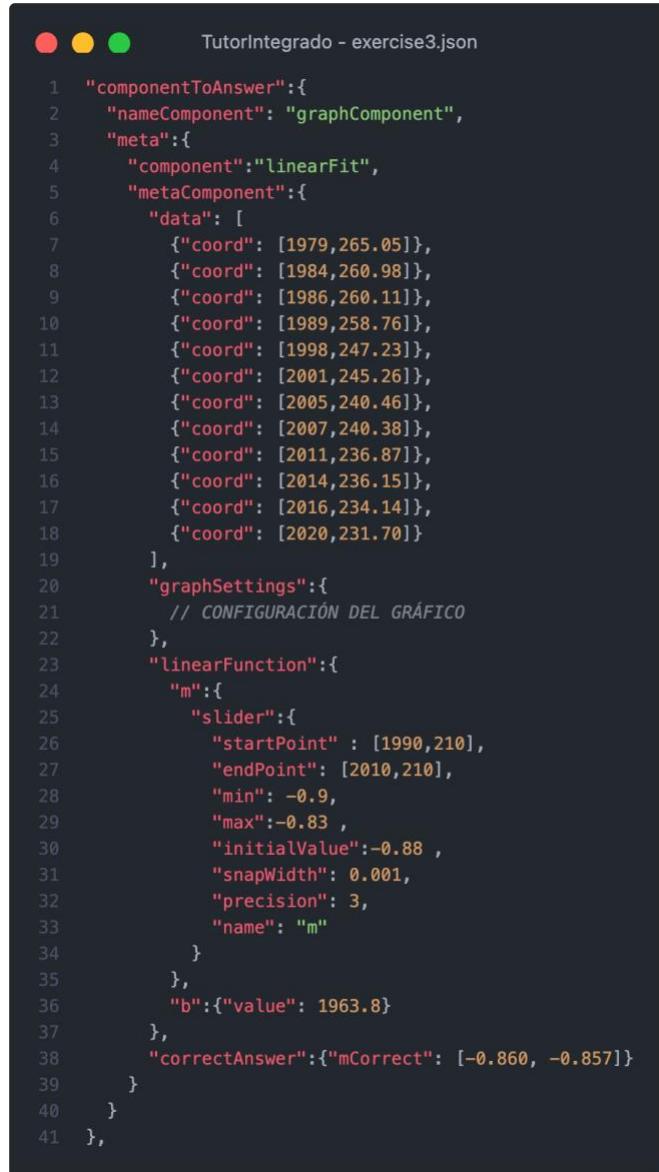
"linearFit" también utiliza el atributo "data" para representar los puntos en el gráfico, permitiendo definirlos solo por sus coordenadas o personalizarlos con atributos adicionales.

Además del atributo "data", el componente "linearFit" incluye el atributo "linearFunction", el cual se utiliza para definir la recta (una función lineal) que se ajustará a los puntos en el gráfico. Este atributo cuenta con los siguientes sub-atributos para definir la función lineal:

- m : Representa la pendiente "m" de la recta " $y = mx + b$ ". Dependiendo de la definición específica del ejercicio, este valor puede ser un número fijo o un objeto "slider". Cuando es un objeto "slider", el estudiante puede ajustar el valor de "m" deslizando el *slider* hacia la izquierda o la derecha.
- b : Representa el término independiente "b" de la recta " $y = mx + b$ ". Al igual que el coeficiente "m", este valor también puede ser un número fijo o un objeto "slider", dependiendo del ejercicio.

En caso que "m" y/o "b" sean *sliders*, los atributos para su configuración se describen a continuación:

- *startPoint*: Representa el punto inicial de coordenadas (x, y) donde se desplegará el *slider* en el gráfico.
- *endPoint*: Representa el punto final de coordenadas (x, y) donde se ubicará el final del *slider* en el gráfico. El slider se extenderá desde el punto de inicio hasta el punto final en el plano cartesiano, permitiendo generar sliders horizontales o verticales, dependiendo de la ubicaciones de los puntos "startPoint" y "endPoint".
- *min*: Define el valor mínimo que puede tomar el *slider*.
- *max*: Define el valor máximo que puede tomar el *slider*.
- *initialValue*: Define el valor inicial del *slider* cuando se carga el gráfico.
- *snapWidth*: Define el tamaño del paso o incremento que se utiliza al mover el *slider*.
- *precision*: Define la precisión o cantidad de decimales permitidos en el valor del *slider*.
- *name*: Define el nombre o etiqueta que se muestra junto al *slider* en el gráfico.



```

1 "componentToAnswer": {
2   "nameComponent": "graphComponent",
3   "meta": {
4     "component": "linearFit",
5     "metaComponent": {
6       "data": [
7         {"coord": [1979, 265.05]},
8         {"coord": [1984, 260.98]},
9         {"coord": [1986, 260.11]},
10        {"coord": [1989, 258.76]},
11        {"coord": [1998, 247.23]},
12        {"coord": [2001, 245.26]},
13        {"coord": [2005, 240.46]},
14        {"coord": [2007, 240.38]},
15        {"coord": [2011, 236.87]},
16        {"coord": [2014, 236.15]},
17        {"coord": [2016, 234.14]},
18        {"coord": [2020, 231.70]}
19      ],
20      "graphSettings": {
21        // CONFIGURACIÓN DEL GRÁFICO
22      },
23      "linearFunction": {
24        "m": {
25          "slider": {
26            "startPoint": [1990, 210],
27            "endPoint": [2010, 210],
28            "min": -0.9,
29            "max": -0.83,
30            "initialValue": -0.88,
31            "snapWidth": 0.001,
32            "precision": 3,
33            "name": "m"
34          }
35        },
36        "b": {
37          "value": 1963.8
38        },
39        "correctAnswer": {
39          "mCorrect": [-0.860, -0.857]
40        }
41      }
42    }
43  }

```

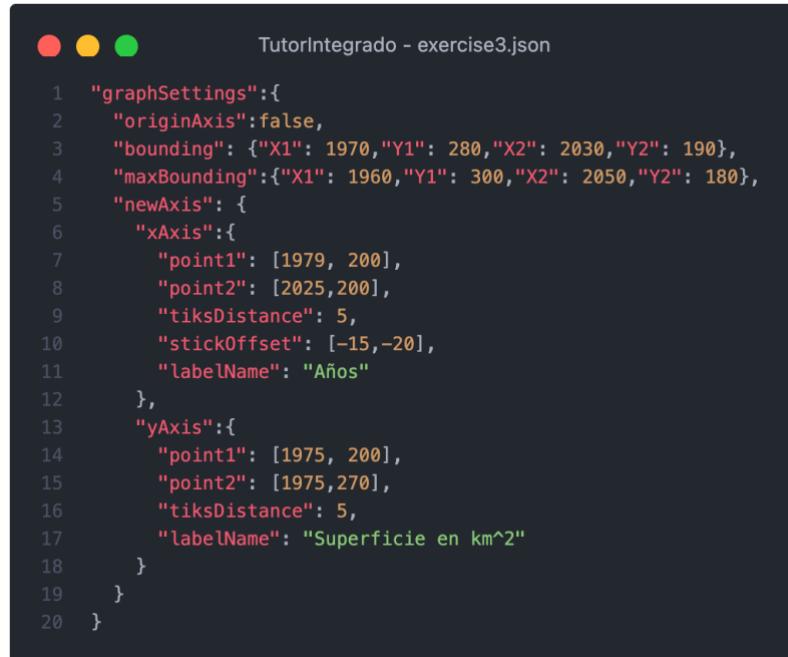
Figura 43: Ejemplo de estructura JSON del componente *linearFit*.

Por último, ambos componentes utilizan el atributo “*graphSettings*” para la configuración inicial del grafico cartesiano, la estructura JSON de este atributo se puede visualizar en la Figura 44, y los atributos se describen a continuación:

- *originAxis*: Es un booleano que define si se utilizarán los ejes originales centrados en el origen. Si se establece en “*false*”, no se mostrarán los ejes originales, y será necesario definir manualmente la posición de los nuevos ejes mediante los atributos “*newAxis*”.
- *bounding*: Es un objeto que define el rango de visualización inicial del gráfico cartesiano. Se representa mediante dos puntos de coordenadas (x1, y1) y (x2, y2),

donde el rectángulo formado por estos dos puntos corresponde al área visible del gráfico. Todo lo que se encuentre fuera de este rango no será visible inicialmente.

- *maxBounding*: Objeto que define el rango máximo de visualización del grafico cartesiano. Limita el rango de movimiento y zoom en el grafico cartesiano. Los puntos que definen “*maxBounding*” deben contener los puntos de “*bounding*”.
- *newAxis*: Es un objeto que permite definir nuevos ejes personalizados y contiene los atributos “*xAxis*” e “*yAxis*”, cada uno de estos atributos contienen los siguientes sub-atributos:
 - *point1*: Es un arreglo que contiene las coordenadas (x, y) del punto inicial del nuevo eje.
 - *point2*: Es un arreglo que contiene las coordenadas (x, y) del punto final del nuevo eje. El nuevo eje se creará como una línea que atraviesa los puntos “*point1*” y “*point2*”.
 - *tiksDistance*: Es un número que define la distancia entre las marcas o divisiones en el eje.
 - *stickOffset*: Es un arreglo que indica el desplazamiento de las marcas del nuevo eje. Permite personalizar la posición de las marcas en el eje.
 - *labelName*: Es un string que contiene el nombre o etiqueta del nuevo eje.



The screenshot shows a code editor window with three tabs at the top: a red one, a yellow one, and a green one. The title bar says "TutorIntegrado - exercise3.json". The main area contains the following JSON code:

```
1  "graphSettings":{  
2    "originAxis":false,  
3    "bounding": {"X1": 1970,"Y1": 280,"X2": 2030,"Y2": 190},  
4    "maxBounding": {"X1": 1960,"Y1": 300,"X2": 2050,"Y2": 180},  
5    "newAxis": {  
6      "xAxis":{  
7        "point1": [1979, 200],  
8        "point2": [2025,200],  
9        "tiksDistance": 5,  
10       "stickOffset": [-15,-20],  
11       "labelName": "Años"  
12     },  
13     "yAxis":{  
14       "point1": [1975, 200],  
15       "point2": [1975,270],  
16       "tiksDistance": 5,  
17       "labelName": "Superficie en km^2"  
18     }  
19   }  
20 }
```

Figura 44: Estructura JSON del atributo *graphSettings*.

5. IMPLEMENTACIÓN

En este capítulo se abordará la implementación de los ejercicios, que corresponde al cuarto objetivo específico del proyecto. El desarrollo de esta etapa se llevará a cabo mediante *sprints*, donde en cada uno de estos se entregará un incremento del producto final.

En cada *sprint* se llevarán a cabo reuniones semanales para mostrar los avances y recibir retroalimentación por parte de los profesores. Al finalizar cada *sprint*, se realizará una revisión y evaluación del incremento realizado, asegurando su correcta funcionalidad y su conformidad con los requisitos establecidos. Luego, se planificará y comenzará el siguiente *sprint*, continuando así con el proceso iterativo de implementación hasta lograr la finalización exitosa del proyecto.

La implementación de los ejercicios se realizará siguiendo el diseño y la estructura definida en los capítulos anteriores. Además, se utilizarán las tecnologías especificadas en el capítulo de Metodología de Desarrollo, y las herramientas seleccionadas para este proceso serán: *Visual Studio Code* como editor de código, *Github* para el control de versiones y el correo institucional como canal de comunicación.

5.1 Sprint 1

En la implementación de este proyecto, se trabajó con la última versión del proyecto general de tutores cognitivos. La estructura de archivos del proyecto general se puede observar en la Figura 45. Cabe destacar que las distintas implementaciones de los tutores cognitivos se encuentran en el directorio "src/components". En dicho directorio, cada carpeta corresponde a la implementación de los diferentes tópicos disponibles actualmente en la plataforma. Además, dentro de cada carpeta, se encuentran los archivos que definen los diversos componentes y la lógica necesaria para cada implementación de los tutores.

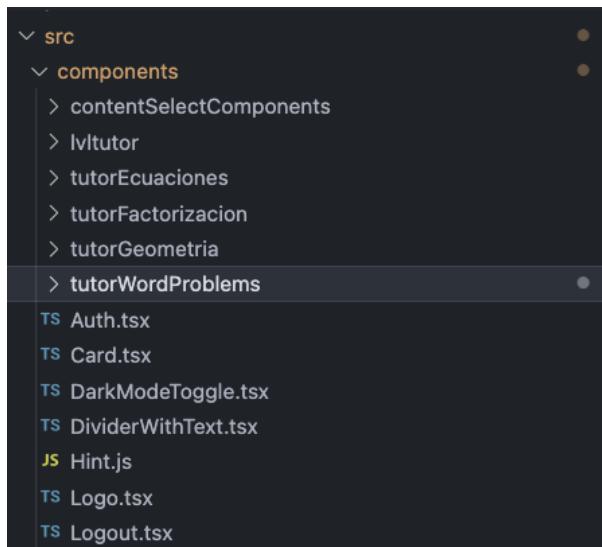


Figura 45: Estructura de archivos del proyecto de tutores cognitivos

5.1.1 Sprint backlog

Para este primer sprint las tareas a realizar son:

- Crear estructura de archivos para el proyecto.
- Desplegar elementos del Ejercicio 1 (enunciado, tabla/imagen, preguntas).
- Desplegar pasos para cada pregunta.
- Implementación del componente “*selectionComponent*”.

Adicionalmente a estas tareas generales, se debe cumplir los siguientes criterios para cada paso:

- Desplegar título del paso (*stepTitle*).
- Desplegar campo “*explanation*” en caso de estar definido.
- Identificar que componente se utilizará para responder el paso.
- Desplegar una alerta con el mensaje correspondiente en caso de que la respuesta sea correcta o incorrecta.

5.1.2 Product increment

Para el desarrollo de este proyecto en particular, se creó la carpeta “*tutorWordProblems*” en la ruta “*src/components*” del proyecto general. La estructura de archivos de esta carpeta se puede observar en la Figura 46.

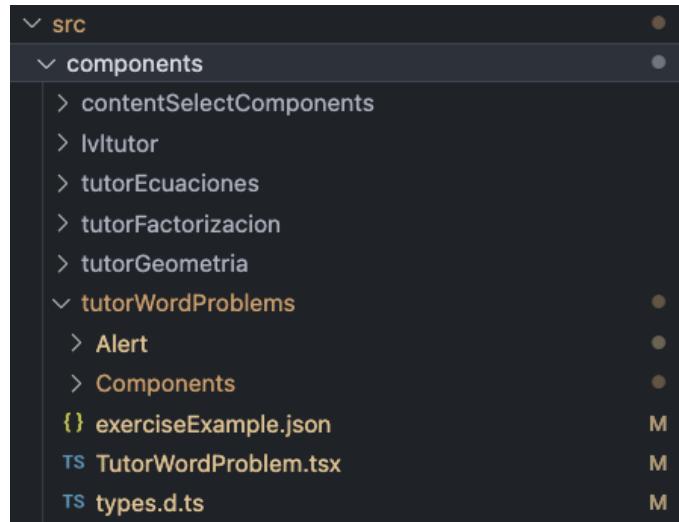


Figura 46: Estructura de archivos del proyecto

Dentro de la carpeta “*tutorWordProblems*” se encuentran tres archivos importantes para el desarrollo del proyecto:

- “*exerciseExample.json*”: Este archivo contiene la estructura JSON del ejercicio 1 y se utiliza como referencia para facilitar el desarrollo. Además, proporciona una

estructura predefinida que sirve como punto de partida para la creación de ejercicios similares.

- "*TutorWordProblem.tsx*": En este archivo se importan los componentes necesarios y se define la lógica para el despliegue de los ejercicios en la plataforma de tutores cognitivos. Aquí se lleva a cabo la integración de los diferentes componentes y se define cómo se mostrará y se interactuará con el ejercicio en la plataforma.
- "*types.d.ts*": Este archivo es de tipado de *TypeScript* y se utiliza para definir los tipos de datos que se utilizarán en el proyecto. Esto proporciona una mayor claridad y ayuda a prevenir errores al momento de utilizar variables y objetos en el código.

Dentro de la carpeta “*tutorWordProblems*” también se crearán carpetas adicionales para los componentes comunes que será utilizados por los ejercicios. Por ejemplo, una de esta carpeta es “*Alert*” (ver Figura 46), la cual contendrá los archivos necesarios para el despliegue de la alerta que notifica al estudiante si se ha equivocado o ha respondido correctamente un paso.

La estructura para los componentes se puede observar en la Figura 47. Esta contiene los archivos correspondientes a cada componente y una carpeta denominada “*tools*” que contendrá los componentes necesarios para el correcto funcionamiento de cada componente.

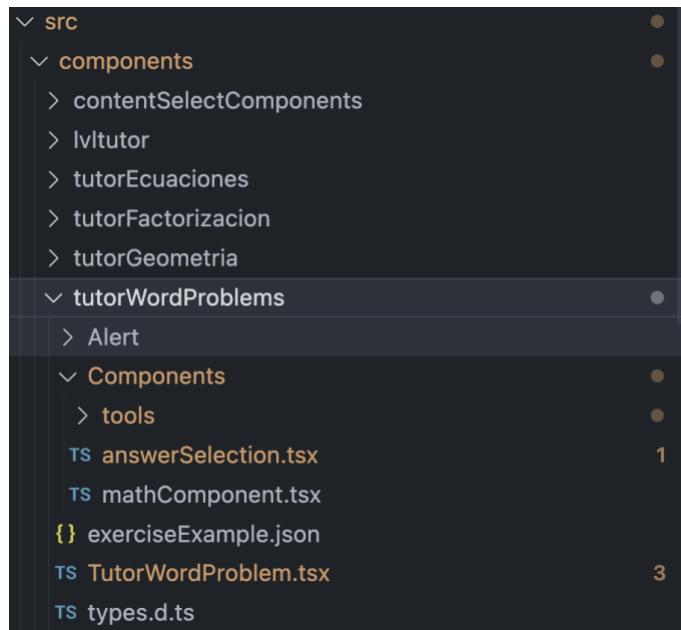


Figura 47: Estructura de archivos de los componentes

La Figura 48 muestra la pieza de software del Ejercicio 1, en la cual se despliegan el enunciado, la tabla y las preguntas correspondientes. Esta pieza de software es parte de la

primera entrega del primer *sprint* y representa la interfaz gráfica donde los estudiantes interactúan con el ejercicio.

The screenshot shows a software application window. On the left, there's a sidebar with a user profile (Enzo marin, enzomarin12@gmail.com), navigation links (Inicio, Selección por Código, Search by Code), and a Tópicos section listing Factorización, Fracción Algebraica, Ecuaciones, Potencias y raíces, Geometría, and Ejercicios en contexto. A small icon of a person is at the bottom of the sidebar. The main content area contains a text block about Chile's CO2 emissions tracking, followed by a table titled 'SECTOR' and 'CO2 NETO'. The table data is as follows:

SECTOR	CO2 NETO
Energía	$8,18231 * 10^{13}$
Procesos Industriales	$4,3602 * 10^{12}$
Agricultura, Silvicultura y otros usos de la tierra	$-3.9915 * 10^{13}$
Residuos	$3 * 10^8$

Below the table is a caption: 'Tabla: Niveles de CO2 por sector en Chile durante el año 2013'. A note states: 'Los numeros positivos representan emisiones de GEI, mientras que los negativos representan absorciones de GEI'. It also says: 'A partir de la tabla anterior nos gustaría identificar:' followed by four numbered questions:

1. ¿Qué sector emite más CO2, contribuyendo mayormente a la contaminación?
2. ¿Qué sector absorbe CO2, disminuyendo la contaminación?
3. Convierte los valores de CO2 de los sectores en potencias de 10^9
4. ¿Cuál es el total de emisiones?

Figura 48: Pieza de software correspondiente al Ejercicio 1.

Las preguntas del ejercicio son desplegables, lo que significa que, al hacer clic en cada pregunta, se despliegan los pasos correspondientes para resolverla. Esta funcionalidad permite a los estudiantes seguir una secuencia clara y detallada de pasos para abordar cada pregunta de manera adecuada.

En la Figura 49 se puede observar el despliegue de los pasos para las preguntas 1, 2 y 3 del ejercicio. Es importante destacar que, aunque en la figura se presenten los pasos de todas las preguntas, la dinámica real de resolución requiere que los estudiantes respondan las preguntas y los pasos de manera secuencial, es decir, el estudiante debe responder correctamente un paso para desbloquear el siguiente, de la misma manera para las preguntas. Esta función se implementará posteriormente en el proyecto.

Figura 49: Despliegue de los pasos de las preguntas

En lo que respecta al componente de selección, presentado en la Figura 50, brinda al estudiante la capacidad de elegir una opción entre varias alternativas proporcionadas, de las cuales solo una es la correcta.

1. ¿Qué sector emite más CO2, contribuyendo mayormente a la contaminación?

Selecciona la alternativa correcta

0. Residuos

1. Energia

2. Procesos industriales

3. Usos de la tierra

Figura 50: Componente de selección.

Cuando el estudiante selecciona una opción incorrecta, se resalta en color rojo y se muestra una alerta indicando que la respuesta es incorrecta, como se muestra en la Figura 51. Por otro lado, cuando el estudiante selecciona la respuesta correcta, la opción se resalta en color verde. En caso de que el campo "*correctMsg*" esté definido, se muestra el mensaje correspondiente en la alerta. Por el contrario, si el campo "*correctMsg*" no está definido, se muestra un mensaje por defecto, en este caso es "Respuesta correcta", como se puede

observar en la Figura 52. Además, después de seleccionar la respuesta correcta, se desactiva la posibilidad de seleccionar otras respuestas.

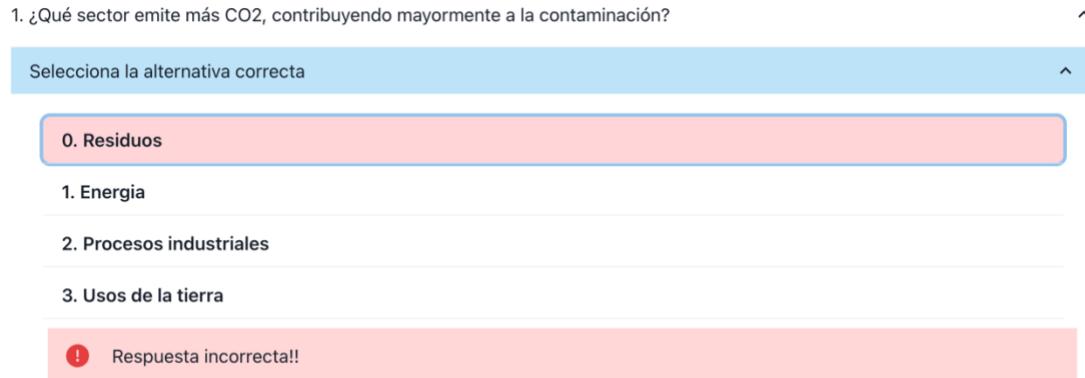


Figura 51: Despliegue de alerta al ingresar respuesta incorrecta.

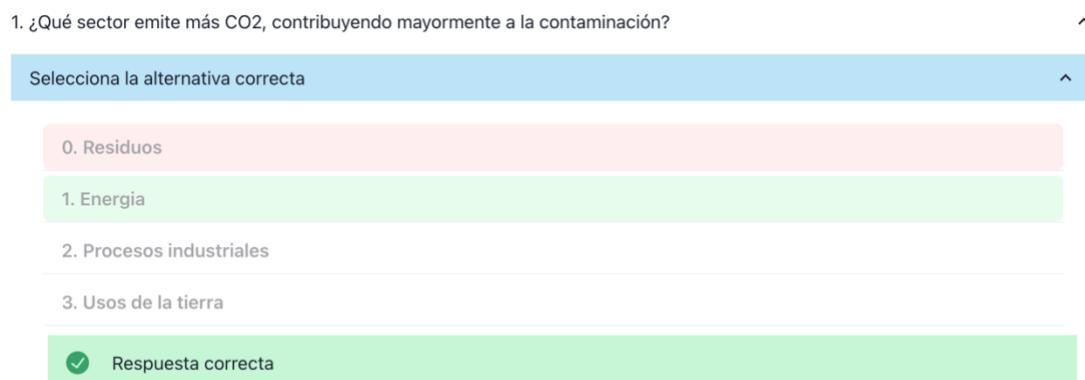
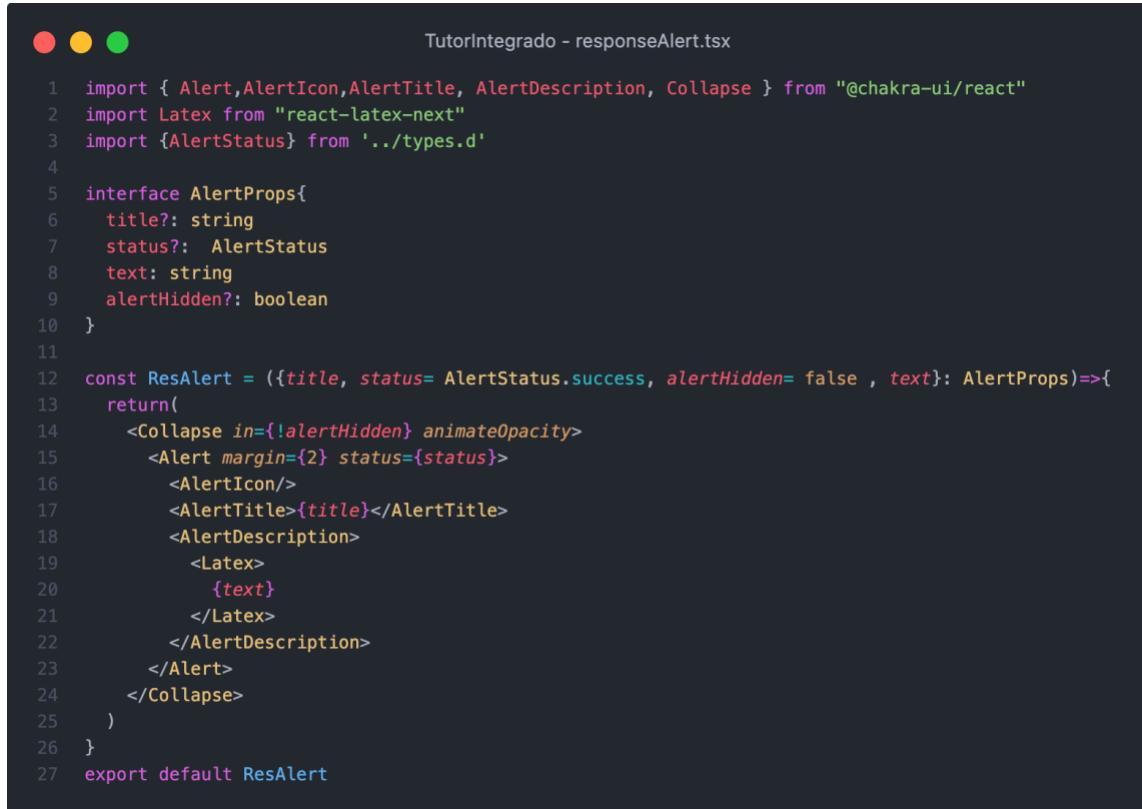


Figura 52: Despliegue de alerta al ingresar respuesta correcta.

Para el despliegue de las alertas, se ha separado la lógica de la visualización en dos componentes. En primer lugar, se creó un archivo llamado "*responseAlert.tsx*" en la ruta "*tutorWordProblems/Alert*". En este archivo, se define una *interface* llamada "*AlertProps*" que especifica los parámetros que el componente debe recibir. Estos parámetros incluyen el "*title*", que corresponde al título de la alerta, el "*status*", que indica el estado de la alerta, el cual puede ser "*success*", "*error*", "*warning*" o "*info*", el "*text*", que corresponde al contenido de la alerta, y "*alertHidden*", que es un booleano que determina si la alerta está oculta o no.

Dentro del archivo "*responseAlert.tsx*", se retorna el componente "*ResAlert*", el cual se puede utilizar como un componente de *React*. Este componente recibe los parámetros previamente definidos y se encarga de renderizar la alerta con los parámetros proporcionados.

El código correspondiente al archivo "responseAlert.tsx" se muestra en la Figura 53. Este código define la estructura y el comportamiento del componente "ResAlert", permitiendo que se visualice la alerta de acuerdo con los parámetros recibidos. Esto facilita la implementación y reutilización de las alertas en diferentes partes de la aplicación.



```
1 import { Alert, AlertIcon, AlertTitle, AlertDescription, Collapse } from "@chakra-ui/react"
2 import Latex from "react-latex-next"
3 import {AlertStatus} from '../types.d'
4
5 interface AlertProps{
6   title?: string
7   status?: AlertStatus
8   text: string
9   alertHidden?: boolean
10 }
11
12 const ResAlert = ({title, status= AlertStatus.success, alertHidden= false , text}: AlertProps)=>{
13   return(
14     <Collapse in={!alertHidden} animateOpacity>
15       <Alert margin={2} status={status}>
16         <AlertIcon/>
17         <AlertTitle>{title}</AlertTitle>
18         <AlertDescription>
19           <Latex>
20             {text}
21           </Latex>
22         </AlertDescription>
23       </Alert>
24     </Collapse>
25   )
26 }
27 export default ResAlert
```

Figura 53: Código componente *ResAlert*

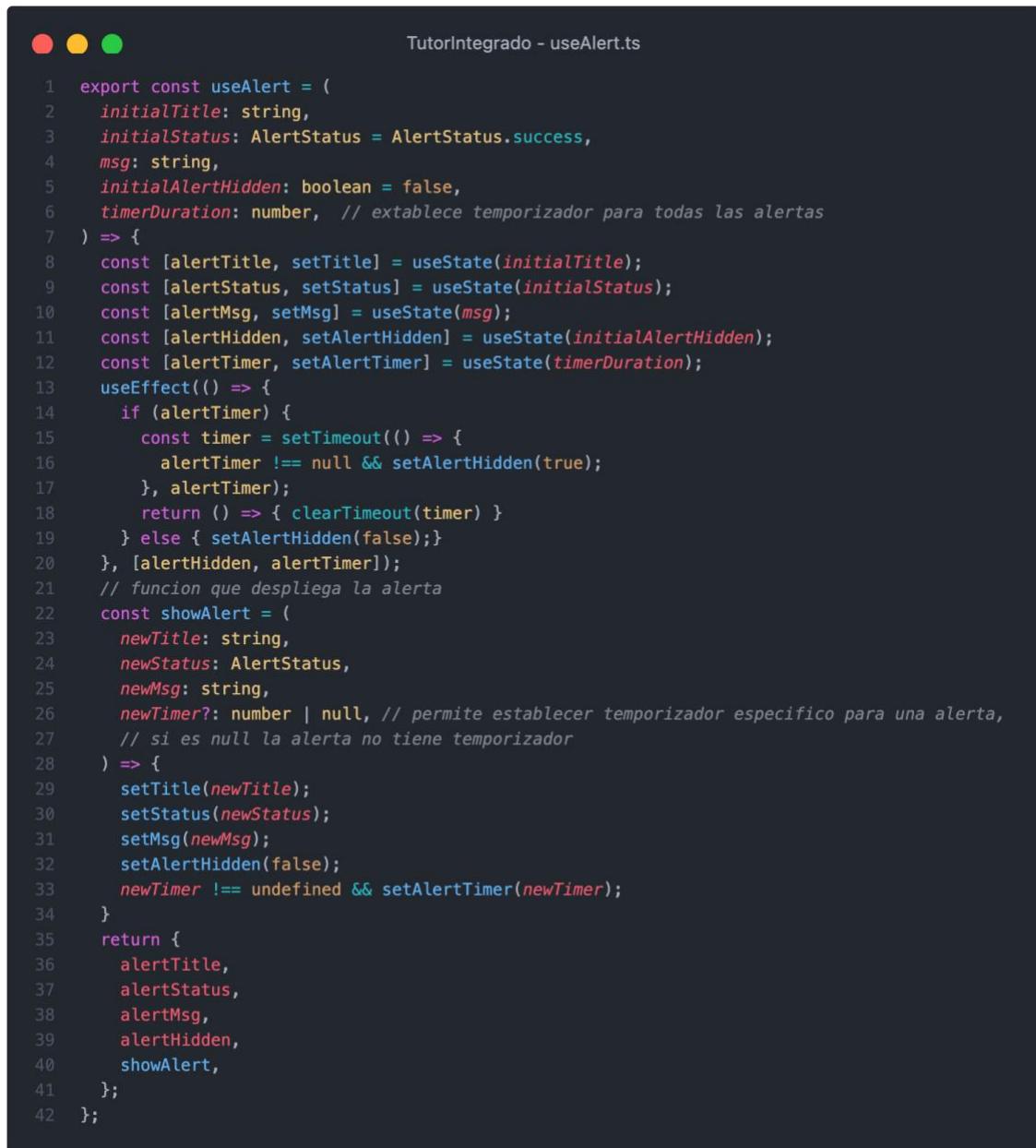
Para la lógica relacionada con las alertas, se ha creado un *custom hook* llamado "useAlert.ts". Los *custom hooks* son funciones especiales en React que permiten reutilizar lógica entre componentes funcionales (Meta open Source, 2023). En este caso, el *custom hook* se encarga de manejar el estado de las alertas y proveer las funciones necesarias para mostrar u ocultar las alertas según sea necesario.

En el archivo "useAlert.ts" (ver Figura 54), se define la lógica para crear y controlar las alertas en la aplicación. El *custom hook* utiliza los *hooks* proporcionados por React, como *useState* y *useEffect*, para gestionar el estado de las alertas. En el archivo se definen estados para los elementos de la alerta como el título, su estado, el texto y un booleano para ocultarla. Además, se incluye un estado para controlar el tiempo durante el cual se mostrará la alerta.

En el *custom hook* también se define la función "showAlert", que recibe como parámetros un título, el estado, el mensaje y un temporizador opcional. El temporizador puede ser un

número para especificar un tiempo determinado durante el cual se mostrará la alerta, o puede ser nulo si se desea que la alerta permanezca visible indefinidamente.

La función del *custom hook* es proporcionar una forma sencilla y reutilizable de manejar las alertas en la aplicación. Al utilizar este *custom hook* en los componentes necesarios, se puede acceder al estado de las alertas y la función para mostrar las alertas de manera fácil y eficiente.



```
1 export const useAlert = (
2   initialTitle: string,
3   initialStatus: AlertStatus = AlertStatus.success,
4   msg: string,
5   initialAlertHidden: boolean = false,
6   timerDuration: number, // establece temporizador para todas las alertas
7 ) => {
8   const [alertTitle, setTitle] = useState(initialTitle);
9   const [alertStatus, setStatus] = useState(initialStatus);
10  const [alertMsg, setMsg] = useState(msg);
11  const [alertHidden, setAlertHidden] = useState(initialAlertHidden);
12  const [alertTimer, setAlertTimer] = useState(timerDuration);
13  useEffect(() => {
14    if (alertTimer) {
15      const timer = setTimeout(() => {
16        alertTimer !== null && setAlertHidden(true);
17      }, alertTimer);
18      return () => { clearTimeout(timer) }
19    } else { setAlertHidden(false); }
20  }, [alertHidden, alertTimer]);
21  // función que despliega la alerta
22  const showAlert = (
23    newTitle: string,
24    newStatus: AlertStatus,
25    newMsg: string,
26    newTimer?: number | null, // permite establecer temporizador específico para una alerta,
27    // si es null la alerta no tiene temporizador
28  ) => {
29    setTitle(newTitle);
30    setStatus(newStatus);
31    setMsg(newMsg);
32    setAlertHidden(false);
33    newTimer !== undefined && setAlertTimer(newTimer);
34  }
35  return {
36    alertTitle,
37    alertStatus,
38    alertMsg,
39    alertHidden,
40    showAlert,
41  };
42};
```

Figura 54: Código del *custom hook* *useAlert*

La Figura 55 representa el código para la implementación del componente “*SelectionComponent*”. En este, se define tanto la estructura como la lógica del componente. Dicho componente recibe los siguientes parámetros:

- *meta*: Objeto que contiene el campo “*answers*”, correspondiente al arreglo de alternativas a desplegar, y “*idCorrectAnswers*”, el cual corresponde al índice del arreglo de alternativas que identifica la respuesta correcta. La interfaz (*SelectionMeta*) del campo “*meta*” se detalla en la Figura 56.
- *Hints*: Corresponde a un arreglo que contiene los mensajes de ayuda (pista) para el paso. Estas ayudas proporcionan información adicional o pistas que pueden ser mostradas al estudiante para guiarlo durante la resolución del ejercicio. La implementación y utilización de estas pistas se abordarán en secciones posteriores del documento.
- *correctMsg*: Corresponde a el mensaje a presentar en la alerta cuando el estudiante responde correctamente el paso. Este mensaje se puede utilizar para felicitar al estudiante por una respuesta correcta o proporcionar una retroalimentación positiva.

Es importante destacar que, aunque estos parámetros son comunes en diversos componentes, el parámetro “*meta*” puede variar según el componente en cuestión, adaptándose a las necesidades particulares de cada uno. Los elementos que constituyen el parámetro “*meta*” se definen de manera específica para cada tipo de componente en sus respectivas interfaces.

Todas las interfaces utilizadas a partir de este punto se definen en el archivo “*types.d.ts*” del proyecto. Estas interfaces establecen la estructura esperada de los datos en los componentes y definen la estructura de objetos u otras partes del código.

En el código de la Figura 55 se pueden identificar dos estados importantes, “*userSelectAnswer*” e “*isCorrectUserAnswer*”. El primero almacena el índice de la respuesta seleccionada por el estudiante, mientras que el segundo contiene una variable booleana que determina si el paso ha sido respondido correctamente.

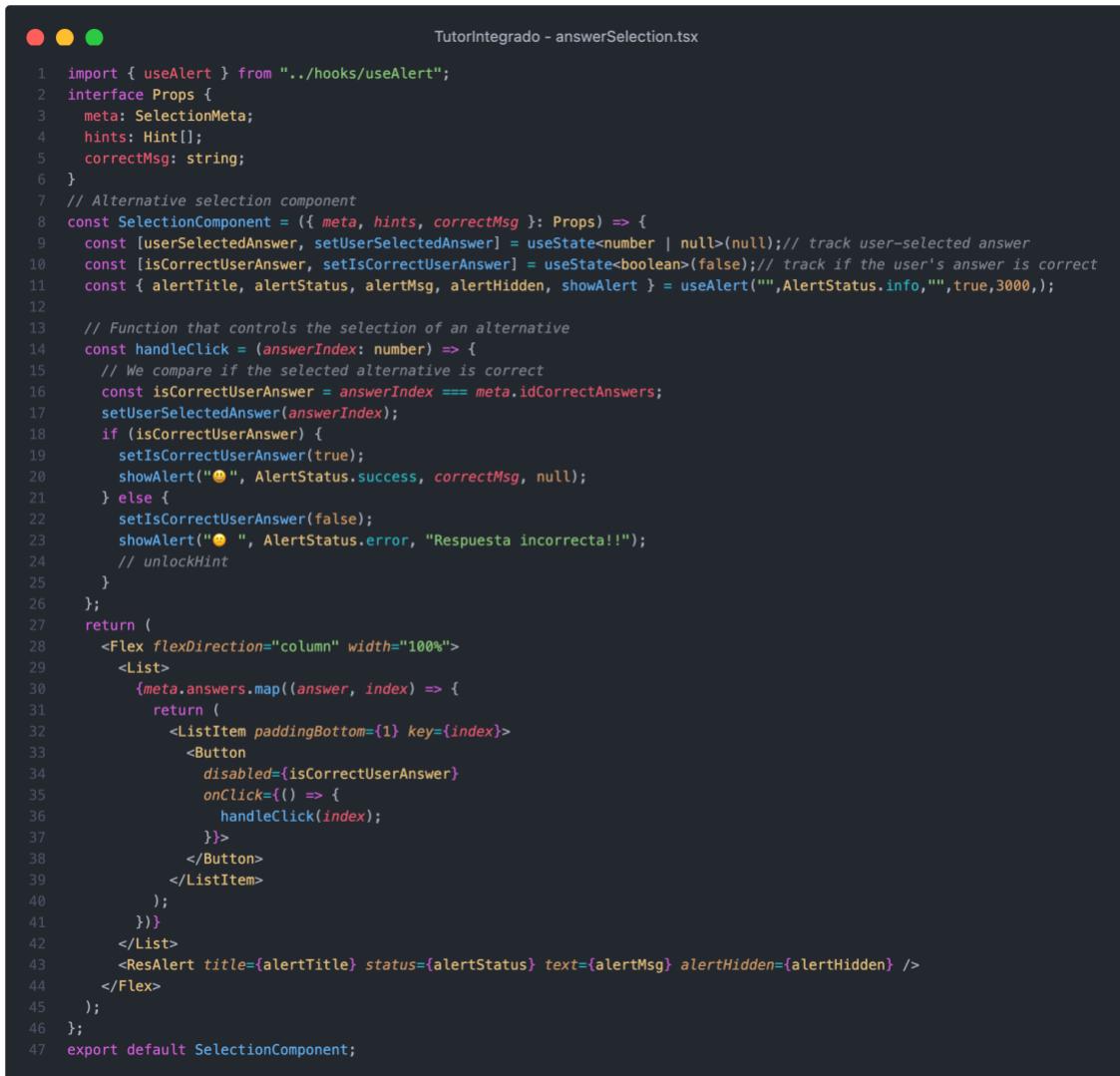
El componente utiliza el *custom hook* “*useAlert*” definido previamente para manejar el estado de las alertas. El *custom hook* se inicializa con valores predeterminados, como un título y mensaje de alerta vacíos, un estado “*info*” y un temporizador inicial. Como resultado, el *custom hook* devuelve la función “*showAlert*” y los elementos necesarios para la visualización de la alerta, como “*alertTitle*”, “*alertStatus*”, “*alertMsg*” y “*alertHidden*”. Estos elementos son los parámetros que el componente “*ResAlert*” necesita para el despliegue de la alerta.

Adicionalmente, en el componente se define la función “*handleClick*”. Esta función compara la alternativa seleccionada por el estudiante con la respuesta correcta almacenada en “*meta.idCorrectAnswers*”. Dependiendo de si la respuesta es correcta o incorrecta se

llama a la función “*showAlert*” y se le pasan los nuevos valores del título, estado, mensaje y temporizador para el despliegue de la alerta.

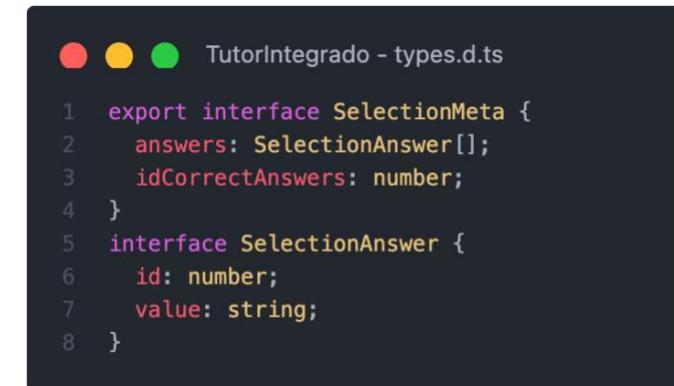
Por ejemplo, si la respuesta es correcta, se invocaría la función “*showAlert*” con el título de la alerta correspondiente a un emoticón sonriente, el estado “*success*”, el mensaje obtenido del parámetro “*correctMsg*” y el temporizador definido como *null*. Esto desplegaría una alerta indicando que la respuesta es correcta y se mantendría visible de forma indefinida.

Finalmente, el componente “*SelectionComponent*” renderiza cada alternativa mediante un mapeo del campo “*meta.answers*”. Para cada alternativa, se crea un elemento “*Button*” y se asigna la función “*handleClick*” al atributo “*onClick*” del botón. De esta manera, la función “*handleClick*” se ejecuta cada vez que el estudiante selecciona una alternativa.



```
 1 import { useAlert } from "../hooks/useAlert";
 2 interface Props {
 3   meta: SelectionMeta;
 4   hints: Hint[];
 5   correctMsg: string;
 6 }
 7 // Alternative selection component
 8 const SelectionComponent = ({ meta, hints, correctMsg }: Props) => {
 9   const [userSelectedAnswer, setUserSelectedAnswer] = useState<number | null>(null); // track user-selected answer
10   const [isCorrectUserAnswer, setIsCorrectUserAnswer] = useState<boolean>(false); // track if the user's answer is correct
11   const { alertTitle, alertStatus, alertMsg, alertHidden, showAlert } = useAlert("", AlertStatus.info, "", true, 3000);
12
13   // Function that controls the selection of an alternative
14   const handleClick = (answerIndex: number) => {
15     // We compare if the selected alternative is correct
16     const isCorrectUserAnswer = answerIndex === meta.idCorrectAnswers;
17     setUserSelectedAnswer(answerIndex);
18     if (isCorrectUserAnswer) {
19       setIsCorrectUserAnswer(true);
20       showAlert("😊", AlertStatus.success, correctMsg, null);
21     } else {
22       setIsCorrectUserAnswer(false);
23       showAlert("😢", AlertStatus.error, "Respuesta incorrecta!!!");
24       // unlockHint
25     }
26   };
27   return (
28     <Flex flexDirection="column" width="100%">
29       <List>
30         {meta.answers.map((answer, index) => {
31           return (
32             <ListItem padding="1" key={index}>
33               <Button
34                 disabled={isCorrectUserAnswer}
35                 onClick={() => {
36                   handleClick(index);
37                 }}
38               </Button>
39             </ListItem>
40           );
41         })
42       </List>
43       <ResAlert title={alertTitle} status={alertStatus} text={alertMsg} alertHidden={alertHidden} />
44     </Flex>
45   );
46 };
47 export default SelectionComponent;
```

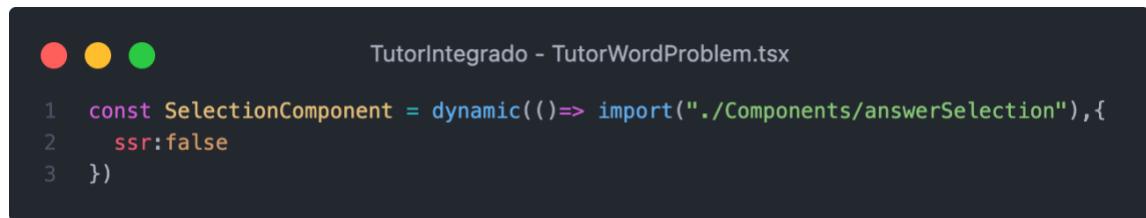
Figura 55: Código del componente de selección.



```
● ● ● TutorIntegrado - types.d.ts
1 export interface SelectionMeta {
2   answers: SelectionAnswer[];
3   idCorrectAnswers: number;
4 }
5 interface SelectionAnswer {
6   id: number;
7   value: string;
8 }
```

Figura 56: Definición de *interface SelectionMeta*

Este componente y los siguientes que se crearán durante el proyecto serán importados dinámicamente desde el archivo “*TutorWordProblem.tsx*” y serán almacenados en una constante. La Figura 57 muestra un ejemplo de cómo se importa el componente de selección y se almacena en la constante “*SelectionComponent*”. Una vez almacenado, este componente puede ser utilizado como un componente de *React*.



```
● ● ● TutorIntegrado - TutorWordProblem.tsx
1 const SelectionComponent = dynamic(()=> import("./Components/answerSelection"),{
2   ssr:false
3 })
```

Figura 57: Código para importar un componente

La Figura 58 corresponde a el código para desplegar los pasos de una pregunta y donde se utiliza el componente *SelectionComponent* previamente importado y almacenado. En este código se realiza un mapeo de los *steps* (pasos) de cada pregunta y para cada *step* (paso) se retorna un componente “*Accordion*” el cual es un componente de ChakraUI que permite plegar y desplegar un elemento.

```

1 <AccordionPanel bg={bgContentColor}>
2   {ques.steps.map((step, index) => {
3     return(
4       <Accordion allowMultiple key={index} >
5         {step.stepExplanation &&
6           <CardInfo text={step.stepExplanation} bgColor= {explanationBgColor} hideCard={false}></CardInfo>
7         }
8         <AccordionItem bgColor={itemBgColor}>
9           <h2>
10          <AccordionButton>
11            <Box as="span" flex="1" textAlign="left">
12              {step.stepTitle}
13            </Box>
14            <AccordionIcon/>
15          </AccordionButton>
16        </h2>
17        <AccordionPanel bg={bgContentColor}>
18          <Box padding={2}>
19            {
20              step.componentToAnswer.nameComponent === components.SLC
21              ? <SelectionComponent
22                correctMsg={step.correctMsg ?? "Muy bien!"}
23                hints={step.hints}
24                meta={step.componentToAnswer.meta as SelectionMeta } >
25              />
26              : (step.componentToAnswer.nameComponent === components.MLC)
27              ? <MathComponent
28                correctMsg={step.correctMsg ?? "Muy bien!"}
29                hints={step.hints}
30                meta={step.componentToAnswer.meta as MathComponentMeta}
31              />
32              : <p>otro componente</p>
33            }
34          </Box>
35        </AccordionPanel>
36      </AccordionItem>
37    </Accordion>
38  )
39 })
40 </AccordionPanel>

```

Figura 58: Código para el despliegue de pasos

Además, en el código se comprueba si existe el atributo *stepExplanation* en cada *step* y se despliega en un componente personalizado *CardInfo*. Luego, se muestra el título desde el atributo *stepTitle* de cada *step*.

En el archivo “*types.d.ts*” se definió un enum llamado “*components*” que se utiliza para definir un conjunto de constantes con los nombres de los componentes (ver Figura 59).



```
● ● ● TutorIntegrado - types.d.ts
1 export enum components{
2   MLC = "mathComponent",
3   SLC = 'selectionComponent',
4   GHPC = "graphComponent"
5 }
6
```

Figura 59: Definición del *enum* "components"

Este *enum* es utilizado en el código de la Figura 58 para verificar que componente se debe renderizar en cada paso. Para ello, se compara el atributo *nameComponent* del campo *componentToAnswer* de cada *step* con las constantes definidas en el *enum* *components*.

En el caso particular del componente de selección, el atributo *nameComponent*, definido previamente en el JSON del ejercicio, contiene el string “selectionComponent”, el cual es comparado con el valor que contiene el *enum components* en su atributo “SLC” (ver Figura 59). Si estos valores coinciden se despliega el componente de selección importado en la Figura 57.

Esta dinámica proporciona flexibilidad y escalabilidad al permitir la adición de futuros componentes. Para agregar un nuevo componente, se debe definir su nombre y valor correspondiente en el *enum components*, importar el componente en el archivo *TutorWordProblem.tsx* y realizar la verificación correspondiente para su despliegue.

5.1.3 Revisión de sprint

Una vez completadas las tareas definidas en el *sprint backlog*, se llevó a cabo la revisión del trabajo realizado. En esta etapa, los resultados fueron presentados a los profesores con el propósito de validar la entrega y obtener retroalimentación.

Durante dicha revisión, se recibió principalmente retroalimentación sobre aspectos visuales, en donde se destacó la importancia de no resaltar visualmente con color la alternativa seleccionada por el estudiante en caso de que esta sea incorrecta. También se sugirió reemplazar la numeración de las alternativas con casillas de verificación (*checkbox*) para cada alternativa.

Adicionalmente, se identificó un error en el despliegue de las alertas. Cuando un estudiante responde incorrectamente, se debe mostrar una alerta indicando el error, la cual se oculta después de un tiempo definido. Por el contrario, cuando la respuesta es correcta, se muestra una alerta con el mensaje correspondiente y la alerta permanece desplegada

indefinidamente. Sin embargo, si se seleccionaba una alternativa incorrecta e inmediatamente después se seleccionaba la alternativa correcta antes de que la alerta de respuesta incorrecta se ocultara, la alerta de respuesta correcta también desaparecía.

La retroalimentación recibida se consideró un aspecto relevante a tener en cuenta para mejorar la experiencia de uso del sistema. En consecuencia, se implementaron las sugerencias recibidas y se corrigió el problema identificado.

5.2 Sprint 2

La segunda entrega del software se enfocó en la implementación del componente matemático, el cual se emplea en los pasos que requieren su funcionalidad. Este componente está diseñado para presentar la información matemática especificada en los archivos JSON correspondientes a cada ejercicio.

Dichos archivos JSON contienen la expresión matemática correspondiente a cada paso, además de los recuadros en blanco que el estudiante debe completar para responder adecuadamente el paso. A continuación, se describen las tareas que se abordaron en esta segunda fase del desarrollo.

5.2.1 Sprint backlog

- Implementación del componente matemático.
- El componente debe desplegar la expresiones matemáticas definidas en el JSON de cada ejercicio, junto con los espacios en blanco (*placeholders*) destinados a ser completados por el estudiante.
- El componente debe validar las respuestas ingresadas por el estudiante en cada recuadro de respuesta (*placeholder*).

5.2.2 Product increment

La Figura 60 muestra la interfaz de la segunda entrega del software, donde se puede observar la pregunta 3 del Ejercicio 1. En esta se despliega el texto explicativo, el cual se define en el archivo JSON correspondiente, y el componente matemático utilizado para la resolución del paso.

Figura 60: Pieza de software correspondiente a la pregunta 3 del Ejercicio 1.

La definición de lo que debe desplegar el componente se define en el atributo “*expression*” del campo “*meta*” en la definición del ejercicio, como se muestra en la Figura 61. Este atributo admite el sistema de composición de texto *LaTex*, lo que permite generar las expresiones matemáticas de manera adecuada, precisa y personalizadas para cada paso.

En el caso específico de la pregunta 3 del Ejercicio 1 (ver Figura 61), se utiliza la sintaxis “*begin*” y “*end*” para generar múltiples líneas de expresión, y el símbolo “&” para alinear las líneas de la expresión. Además, se utiliza la sintaxis “placeholder [] {}”, la que especifica la parte de la expresión que el estudiante debe completar. Visualmente, esto se representa como un recuadro en el cual el estudiante puede ingresar su respuesta. En esta sintaxis se debe proporcionar un identificador entre los corchetes “[]” para cada recuadro (*placeholder*) definido, lo que permite identificar y validar cada respuesta individualmente.

Para lo anterior, en la definición del ejercicio, se debe especificar el atributo “*placeholderId*” junto con el atributo “*value*”. El atributo “*placeholderID*” especifica el identificador único para cada recuadro, y el atributo “*value*” define el valor que se espera que el estudiante ingrese en ese recuadro en particular. De esta manera, el tutor cognitivo puede validar las respuestas ingresadas por el estudiante y proporcionar retroalimentación específica para cada recuadro.

```

1 "componentToAnswer": {
2   "nameComponent": "mathComponent",
3   "meta": {
4     "readonly": true,
5     "expression": "\begin{align}4,3602*10^{12} &= 4,3602*10^{(\placeholder{a})}*10^9 \\\&= \placeholder{b}*10^9 \end{align}",
6     "answers": [
7       {
8         "id": 1,
9         "placeholderId": "a",
10        "value": "3"
11      },
12      {
13        "id": 2,
14        "placeholderId": "b",
15        "value": "4360,2"
16      }
17    ],
18    "correctAnswer": [1, 2]
19  },
20}

```

Figura 61: Campo "*componentToAnswer*" de la pregunta 3 del Ejercicio 1.

Para la implementación del componente matemático se utilizó *Mathlive*, el cual es un componente web que permite la presentación y manipulación de expresiones matemáticas en formato *LaTex* (*CortexJs*, s.f). Este componente brinda funcionalidades avanzadas para crear, editar y renderizar ecuaciones matemáticas de manera interactiva. Además, *Mathlive* proporciona un teclado virtual que integra una variedad de elementos matemáticos, lo que mejora la interacción del estudiante con la plataforma.

Para el uso de este componente web se implementó un componente denominado “*Mathfield*” (ver Figura 62). Este se encarga de facilitar la integración del campo de matemática de *Mathlive* en una aplicación de React. Este componente recibe los siguientes parámetros:

- *readOnly*: Valor booleano opcional que indica si el campo matemático debe ser solo de lectura, su valor por defecto es “*true*”.
- *value*: Cadena de texto que representa el valor actual del campo matemático.
- *mfe*: Instancia del componente “*MathfieldElement*” proporcionado por *MathLive*.
- *onChange*: Función de devolución de llamada que se ejecuta cuando el valor del campo matemático cambia.

Además, el componente utiliza *hooks* de *React* como *useRef*, *useMemo* y *useEffect* para gestionar el estado y efectos del componente, estos se describen a continuación:

- *containerRef*: Referencia a un elemento “*div*” que se utiliza como contenedor para el componente “*MathfieldElement*”.
- *mfe*: Instancia de *MathfieldElement* que se crea utilizando *useMemo*. Si no se proporciona una instancia específica, se crea una nueva instancia por defecto.
- *currentValue*: Referencia a una variable que almacena el valor actual del editor de matemáticas.

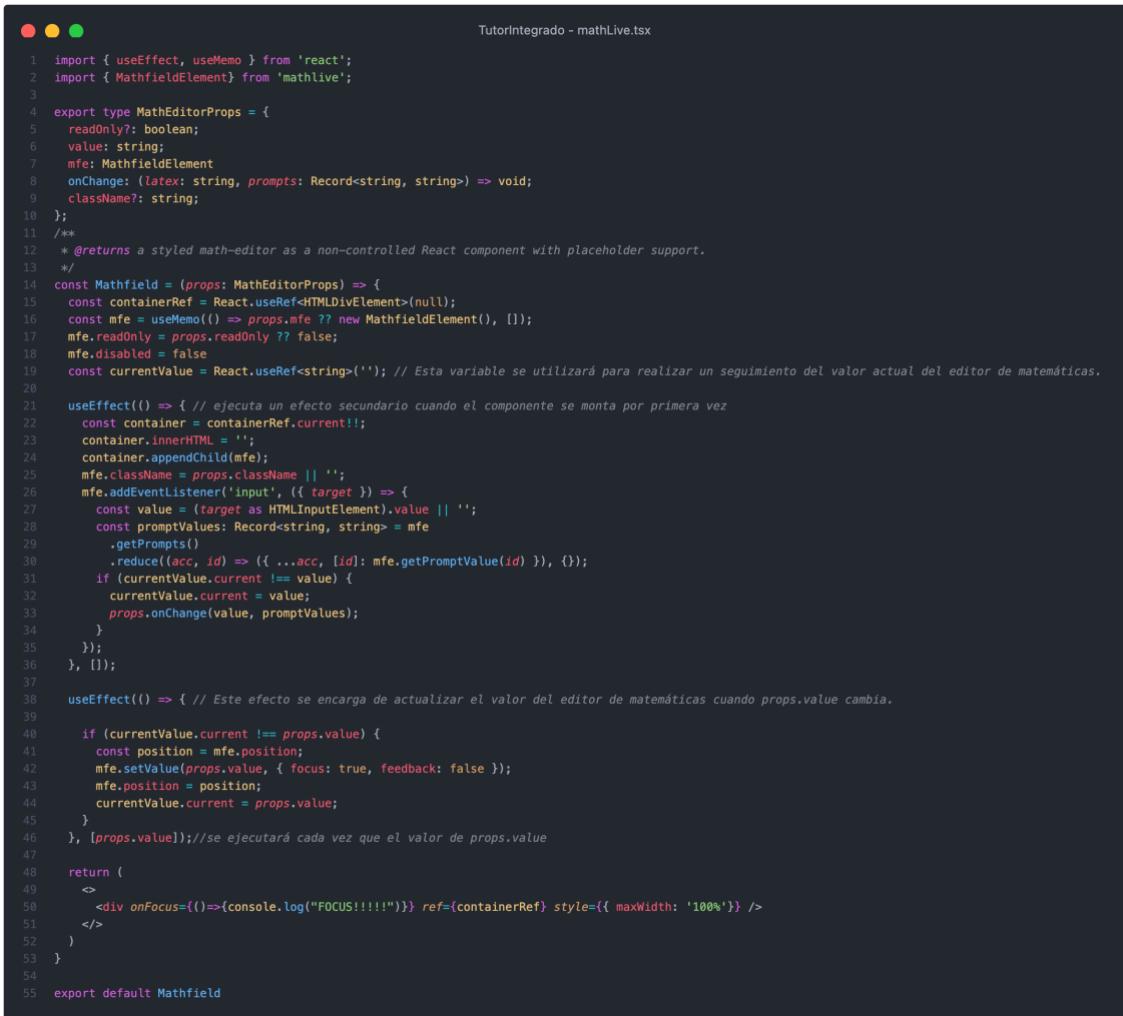
En el primer *useEffect*, se establece el elemento contenedor y se añade el evento de entrada para detectar los cambios en el valor del editor de matemáticas. Este efecto se ejecuta solo

una vez, cuando el componente se monta por primera vez (ver líneas 21 a 26 de la Figura 62).

En el segundo *useEffect*, se actualiza el valor del editor de matemáticas cuando la propiedad “*value*” cambia (ver líneas 38 a 53 de la Figura 62). Esto se hace mediante la llamada a “*mfe.setValue*”, al cual se le pasa el valor actual que contiene el editor. Además, se asegura de mantener la posición del cursor en el editor después de la actualización.

Por último, el componente *Mathfield* devuelve un elemento “*div*” que representa el contenedor del componente matemático, y finalmente, el componente *Mathfield* es exportado, lo que posibilita su utilización como un componente React en otros elementos donde se requiera su funcionalidad.

Esta práctica mejora significativamente la reutilización del componente y fomenta la modularidad del código, lo que facilita el mantenimiento y la escalabilidad del proyecto.



```

1 import { useEffect, useMemo } from 'react';
2 import { MathfieldElement } from 'mathlive';
3
4 export type MathEditorProps = {
5   readOnly?: boolean;
6   value: string;
7   mfe: MathfieldElement;
8   onChange: (latex: string, prompts: Record<string, string>) => void;
9   className?: string;
10 };
11 /**
12  * @returns a styled math-editor as a non-controlled React component with placeholder support.
13 */
14 const Mathfield = (props: MathEditorProps) => {
15   const containerRef = React.useRef<HTMLDivElement>(null);
16   const mfe = useMemo(() => props.mfe ?? new MathfieldElement(), []);
17   mfe.readOnly = props.readOnly ?? false;
18   mfe.disabled = false;
19   const currentValue = React.useRef<string>'(); // Esta variable se utilizará para realizar un seguimiento del valor actual del editor de matemáticas.
20
21   useEffect(() => { // ejecuta un efecto secundario cuando el componente se monta por primera vez
22     const container = containerRef.current!;
23     container.innerHTML = '';
24     container.appendChild(mfe);
25     mfe.className = props.className || '';
26     mfe.addEventListener('input', ({ target }) => {
27       const value = (target as HTMLInputElement).value || '';
28       const promptValues: Record<string, string> = mfe
29         .getPrompts()
30         .reduce((acc, id) => ({ ...acc, [id]: mfe.getPromptValue(id) }), {});
31       if (currentValue.current !== value) {
32         currentValue.current = value;
33         props.onChange(value, promptValues);
34       }
35     });
36   }, []);
37
38   useEffect(() => { // Este efecto se encarga de actualizar el valor del editor de matemáticas cuando props.value cambia.
39
40     if (currentValue.current !== props.value) {
41       const position = mfe.position;
42       mfe.setValue(props.value, { focus: true, feedback: false });
43       mfe.position = position;
44       currentValue.current = props.value;
45     }
46   }, [props.value]); // se ejecutará cada vez que el valor de props.value
47
48   return (
49     <>
50       <div onFocus={()=>{console.log("FOCUS!!!!")}} ref={containerRef} style={{ maxWidth: '100%' }} />
51     </>
52   )
53 }
54
55 export default Mathfield

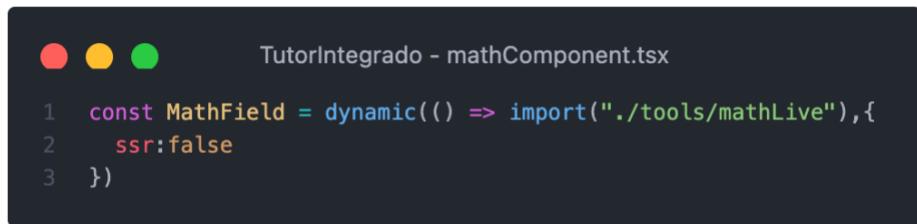
```

Figura 62: Código de la implementación del componente *Mathfield*

Además del componente *Mathfield*, también se implementó el componente “*MathComponent*”. Este último, tiene la responsabilidad de desplegar el campo matemático utilizando el componente *Mathfield* previamente definido, llevar a cabo las validaciones necesarias y mostrar las alertas correspondientes.

El componente *MathComponent* combina la funcionalidad del campo matemático con la lógica de manejo de alertas y validaciones, proporcionando una solución completa para la interacción con expresiones matemáticas en la aplicación.

La Figura 63 corresponde al código para importar el componente *Mathfield* en *MathComponent*.



```
const MathField = dynamic(() => import("./tools/mathLive"), { ssr:false })
```

Figura 63: Código para importar componente *Mathfield*.

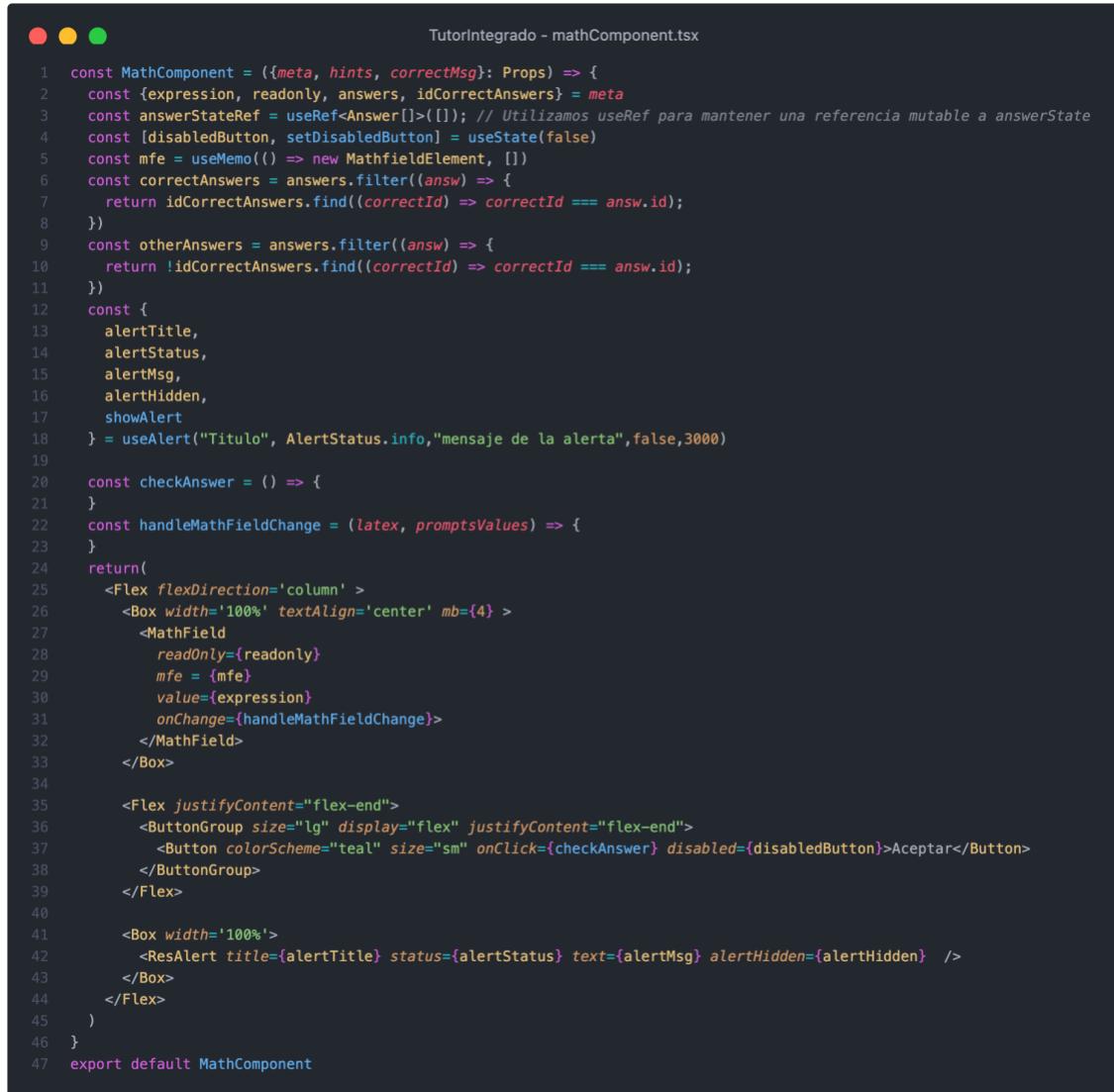
En la definición de *mathComponent*, se especifica que recibe los siguientes parámetros:

- *meta*: Contiene los atributos “*expression*”, “*readOnly*”, “*answers*” y “*correctAnswer*” definidos en el JSON del ejercicio. Estos atributos se utilizan para desplegar el campo matemático.
- *hints*: Contiene el arreglo de ayudas definidos en el ejercicio.
- *correctMsg*: Contiene un mensaje opcional a mostrar cuando el estudiante responde correctamente el paso.

La Figura 64 corresponde al código del componente *MathComponent*, en esta se puede observar los parámetros definidos previamente y ciertas constantes y *hooks* importantes para el correcto funcionamiento del componente, estas se describen a continuación:

- *answerStateRef*: Es una referencia mutable a un arreglo (*Answer[]*) que almacena las respuestas ingresadas por el estudiante. Se utiliza como alternativa a *useState* para mantener el estado de las respuestas sin provocar re-renderizaciones innecesarias cuando cambian los valores de los *placeholders*.
- *disabledButton*: Es un estado que puede tener valores booleanos y se utiliza para determinar si el botón "Aceptar" está activado o desactivado.
- *mfe* : Es una instancia de *MathfieldElement*, proporcionada por el componente web *Mathlive*. Se utiliza para interactuar con el campo matemático y realizar acciones como establecer el estado del recuadro de respuesta y recibir cambios en el contenido del campo (*placeholder*).
- *useAlert*: Custom hook definido previamente que se utiliza para mostrar las alertas correspondientes, como mensajes de advertencia, éxito o error.

Adicionalmente, en el componente se realiza una separación de las respuestas en dos arreglos. El primero, denominado “*correctAnswer*” contiene las respuestas que se consideran correctas, mientras que el segundo, denominado “*otherAnswers*” contiene respuestas que no son consideradas correctas, pero pueden servir para entregar pistas o ayudas específicas al estudiante.



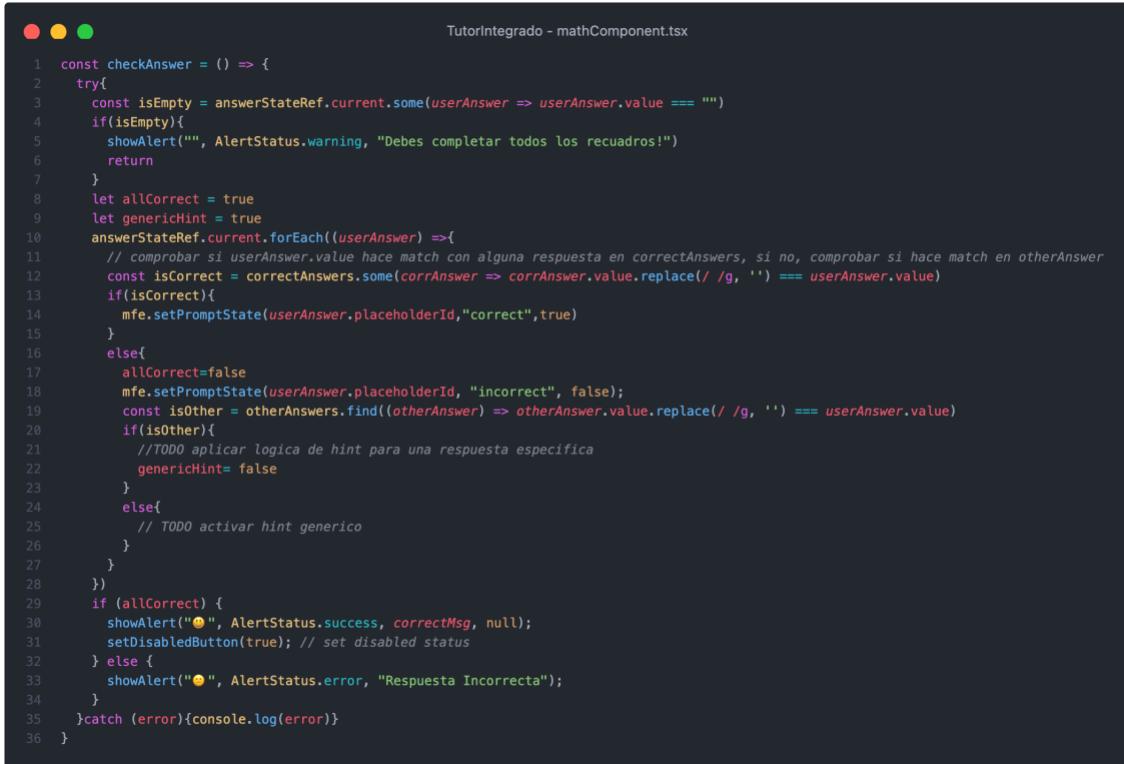
```

1  const MathComponent = ({meta, hints, correctMsg}: Props) => {
2    const {expression, readonly, answers, idCorrectAnswers} = meta
3    const answerStateRef = useRef<Answer[]>([]); // Utilizamos useRef para mantener una referencia mutable a answerState
4    const [disabledButton, setDisabledButton] = useState(false)
5    const mfe = useMemo(() => new MathfieldElement, [])
6    const correctAnswers = answers.filter((answ) => {
7      return idCorrectAnswers.find((correctId) => correctId === answ.id);
8    })
9    const otherAnswers = answers.filter((answ) => {
10      return !idCorrectAnswers.find((correctId) => correctId === answ.id);
11    })
12    const {
13      alertTitle,
14      alertStatus,
15      alertMsg,
16      alertHidden,
17      showAlert
18    } = useAlert("Titulo", AlertStatus.info,"mensaje de la alerta",false,3000)
19
20    const checkAnswer = () => {
21    }
22    const handleMathFieldChange = (latex, promptsValues) => {
23    }
24    return(
25      <Flex flexDirection='column' >
26        <Box width='100%' textAlign='center' mb={4} >
27          <MathField
28            readOnly={readonly}
29            mfe = {mfe}
30            value={expression}
31            onChange={handleMathFieldChange}>
32          </MathField>
33        </Box>
34
35        <Flex justifyContent="flex-end">
36          <ButtonGroup size="lg" display="flex" justifyContent="flex-end">
37            <Button colorScheme="teal" size="sm" onClick={checkAnswer} disabled={disabledButton}>Aceptar</Button>
38          </ButtonGroup>
39        </Flex>
40
41        <Box width='100%'>
42          <ResAlert title={alertTitle} status={alertStatus} text={alertMsg} alertHidden={alertHidden} />
43        </Box>
44      </Flex>
45    )
46  }
47  export default MathComponent

```

Figura 64: código del componente *MathComponent*

La Figura 65 corresponde al código de la función “*checkAnswer*” del componente *mathComponent*. Su principal objetivo es verificar las respuestas ingresadas por el estudiante y proporcionar la retroalimentación adecuada en función de la precisión de las respuestas.



```

1  const checkAnswer = () => {
2    try{
3      const isEmpty = answerStateRef.current.some(userAnswer => userAnswer.value === "")
4      if(isEmpty){
5        showAlert("", AlertStatus.warning, "Debes completar todos los recuadros!")
6        return
7      }
8      let allCorrect = true
9      let genericHint = true
10     answerStateRef.current.forEach((userAnswer) =>{
11       // comprobar si userAnswer.value hace match con alguna respuesta en correctAnswers, si no, comprobar si hace match en otherAnswer
12       const isCorrect = correctAnswers.some(corrAnswer => corrAnswer.value.replace(/ /g, '') === userAnswer.value)
13       if(isCorrect){
14         mfe.setPromptState(userAnswer.placeholderId,"correct",true)
15       }
16       else{
17         allCorrect=false
18         mfe.setPromptState(userAnswer.placeholderId, "incorrect", false);
19         const isOther = otherAnswers.find(otherAnswer) => otherAnswer.value.replace(/ /g, '') === userAnswer.value
20         if(isOther){
21           //TODO aplicar logica de hint para una respuesta especifica
22           genericHint= false
23         }
24         else{
25           // TODO activar hint generico
26         }
27       }
28     })
29     if (allCorrect) {
30       showAlert("!", AlertStatus.success, correctMsg, null);
31       setDisabledButton(true); // set disabled status
32     } else {
33       showAlert("!", AlertStatus.error, "Respuesta Incorrecta");
34     }
35   }catch (error){console.log(error)}
36 }

```

Figura 65: código de la función *checkAnswer*.

En su funcionamiento, la función verifica si el estudiante ha dejado algún recuadro de respuesta en blanco. Si encuentra alguno, muestra una alerta de advertencia (mediante el *custom hook useAlert*) indicando que es necesario completar todos los recuadros antes de continuar. Si hay respuestas vacías, la función termina en este punto y no procede a evaluar las respuestas ingresadas.

Luego, la función recorre las respuestas ingresadas por el estudiante, que se almacenan en la referencia mutable *answerStateRef*. Por cada respuesta, se verifica si la respuesta ingresada (*userAnswer.value*) coincide con alguna de las respuestas correctas (*correctAnswers*). Si hay una coincidencia, se marca el recuadro de respuesta correspondiente (*placeholder*) como "correct" usando el método "*mfe.setPromptState*" del componente *Mathfield*. Esto resalta visualmente que la respuesta es correcta (ver Figura 66).

Si la respuesta no coincide con ninguna respuesta correcta, se marca el recuadro de respuesta correspondiente como "incorrect", resaltando visualmente que la respuesta es incorrecta (ver Figura 66). Luego se busca si coincide con alguna de las respuestas que proporcionan ayuda. Si hay una coincidencia, se activa la lógica de pista específica para esa respuesta. Si no coincide con ninguna respuesta que proporcione ayuda, se activa la pista genérica para todas las respuestas incorrectas.

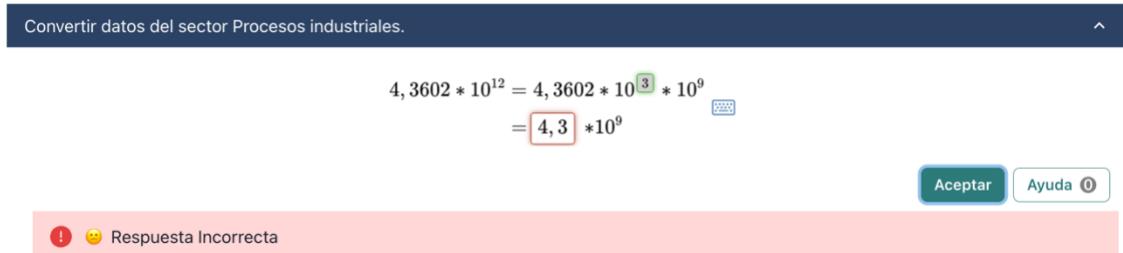


Figura 66: Retroalimentación visual por campo del componente *MathComponent*.

Durante este proceso, la función mantiene una variable (*allCorrect*) para rastrear si todas las respuestas son correctas. Si se encuentra una respuesta incorrecta, esta variable se establece en *false*, lo que indica que no todas las respuestas son correctas.

Al finalizar la evaluación de las respuestas, la función muestra una alerta de éxito si todas las respuestas son correctas (si *allCorrect* es *true*). Esta alerta puede contener un mensaje específico que se pasa como parámetro al componente (*correctMsg*). Luego, se desactiva el botón "Aceptar" para evitar que el estudiante envíe respuestas adicionales una vez que ha respondido correctamente. Por el contrario, si no todas las respuestas son correctas, se muestra una alerta de error para informar al estudiante que su respuesta es incorrecta.

La función "*handleMathFieldChange*", definida en la Figura 67, se encarga de manejar el cambio en el contenido del campo matemático. Esta recibe dos parámetros:

- *latex*: Representa la expresión matemática completa en formato *LaTeX*. Es el contenido actual del campo matemático.
- *promptsValues*: Es un objeto que contiene los valores de los recuadros de respuesta (*placeholders*) ingresados por el estudiante. Cada recuadro tiene un identificador único (*placeholderId*) y el valor ingresado por el estudiante (*value*).

Dentro de la función, se utiliza “*Object.entries(promptsValues)*” para convertir el objeto “*promptsValues*” en un arreglo de pares clave-valor. Esto se hace para poder manipular y transformar los datos más fácilmente.

A continuación, se utiliza el método *map* en el arreglo de pares clave-valor para crear un nuevo arreglo de respuestas. Cada respuesta es un objeto que contiene el “*placeholderId*” y el “*value*” del recuadro de respuesta correspondiente. Estos datos se obtienen mediante la desestructuración de cada par clave-valor en los parámetros de la función de mapeo.

Finalmente, el nuevo arreglo de respuestas se asigna a la referencia “*answerStateRef*”, que almacena las respuestas ingresadas por el estudiante.

```

TutorIntegrado - mathComponent.tsx

1 const handleMathFieldChange = (latex, promptsValues) => {
2   const entries = Object.entries(promptsValues) as [string, string][];
3   answerStateRef.current = entries.map(([placeholderId, value]) => ({
4     placeholderId,
5     value,
6   }));
7 }

```

Figura 67: Código de la función *handleMathFieldChange*

Una vez definido el componente *mathComponent*, este se importa y es utilizado cuando un paso específico lo requiere. La Figura 64 muestra el uso de este componente en el archivo "TutorWordProblems.tsx" cuando el atributo "nameComponent" del paso es igual a "component.MLC", que corresponde al string "*mathComponent*".

```

TutorIntegrado - TutorWordProblem.tsx

1 <AccordionPanel bg={bgContentColor}>
2   <Box padding={2}>
3   {
4     step.componentToAnswer.nameComponent === components.SLC
5     ? <SelectionComponent
6       correctMsg={step.correctMsg ?? "Muy bien!"}
7       hints={step.hints}
8       meta={step.componentToAnswer.meta as SelectionMeta} 
9     />
10    : (step.componentToAnswer.nameComponent === components.MLC)
11    ? <MathComponent
12      correctMsg={step.correctMsg ?? "Muy bien!"}
13      hints={step.hints}
14      meta={step.componentToAnswer.meta as MathComponentMeta}
15    />
16    : <p>otro componente</p>
17  }
18 </Box>
19 </AccordionPanel>

```

Figura 68: Ejemplo de uso del componente *mathComponent*.

5.2.3 Revisión de sprint

La revisión de esta entrega, se enfocó en verificar el correcto funcionamiento del componente matemático. Para ello, se llevó a cabo una reunión con los profesores patrocinante y co-patrocinante, donde se presentó la implementación y se procedió a resolver paso a paso el primer ejercicio. Durante esta revisión, se comprobó el funcionamiento adecuado del componente matemático y se recibieron comentarios y

retroalimentación, principalmente relacionados con el diseño de los ejercicios, como la redacción de los textos explicativos previos a cada paso.

Además, se identificaron algunos errores menores en las expresiones matemáticas desplegadas en el componente, así como un problema en el teclado virtual proporcionado por el componente *MathLive*. En el cual, al ingresar un valor desde el teclado virtual, este se cerraba automáticamente, lo que afectaba directamente la experiencia de uso.

Se tomó en cuenta esta retroalimentación para realizar ajustes y mejoras en el diseño de los ejercicios, los cuales fueron actualizados en el documento. También se hizo énfasis en la importancia de incluir un resumen al finalizar cada ejercicio, resaltando los conceptos matemáticos aprendidos a través de la resolución del ejercicio.

Los comentarios y sugerencias recibidas durante la revisión fueron valiosos para perfeccionar la funcionalidad del componente matemático y mejorar la experiencia global del usuario. Se implementaron las correcciones necesarias en las expresiones matemáticas y se trabajó en la solución del problema del teclado virtual para asegurar un entorno de trabajo más fluido y efectivo.

5.3 Sprint 3

El tercer *sprint* se centra en la implementación de componentes gráficos que amplían la interacción y la visualización de datos. En este sprint, se detallan dos componentes cruciales: "*SelectPoint*" y "*LinearFit*".

Similar a los componentes desarrollados previamente, estos nuevos componentes también reciben los parámetros "*meta*", "*hints*" y "*correctMsg*". En relación al parámetro "*meta*", su definición y los atributos contenidos en él se han explicado con anterioridad en detalle en el capítulo 4 de este documento.

5.3.1 Sprint backlog

- Implementar componente grafico para la identificación de puntos en un gráfico cartesiano.
- Implementar componente grafico para realizar un ajuste lineal.

5.3.2 Product increment

Para la implementación de los componentes gráficos se utilizó JSXGraph, la cual es una librería JavaScript para geometría interactiva, trazado de funciones, gráficos y visualización de datos en el navegador web (JSXGraph, s.f.).

La Figura 69 muestra la interfaz del componente de identificación de puntos en el plano cartesiano. En este caso, no se incluye un botón de "Aceptar" ya que el gráfico es interactivo. De esta manera, los estudiantes deben seleccionar el punto correcto que se les pide identificar directamente en el gráfico.

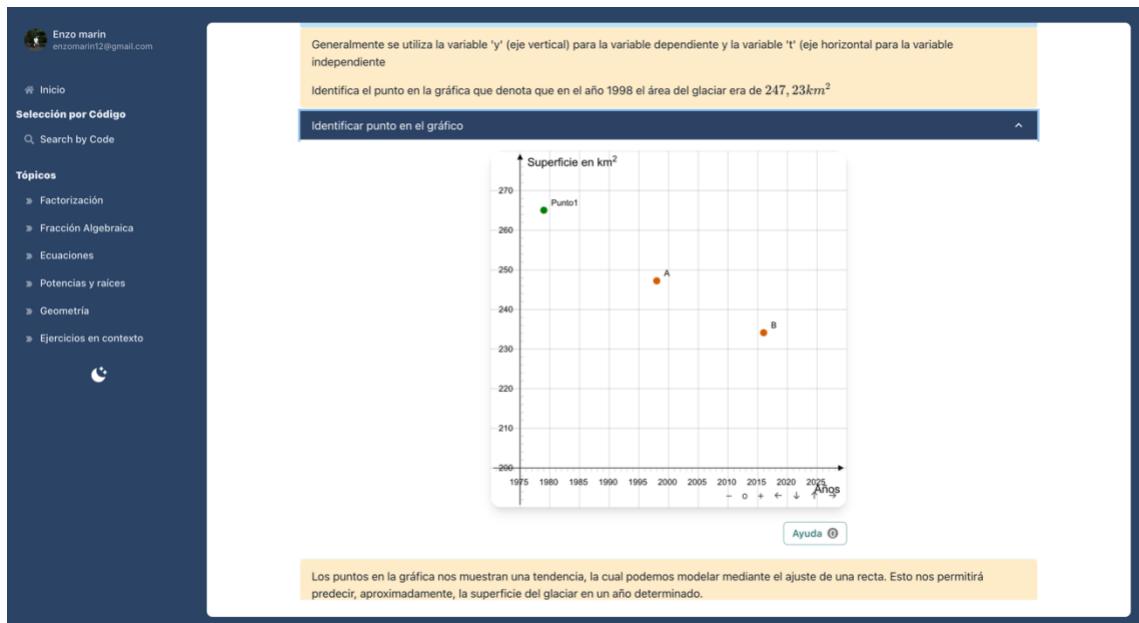


Figura 69: Interfaz del componente *selectPoint*.

Por otro lado, en el componente de ajuste lineal (Figura 70), se han implementado funcionalidades adicionales para ofrecer una experiencia interactiva y educativa más completa. En este componente, se permite al estudiante variar los parámetros de la función mediante el uso de *sliders*.

En el caso específico del Ejercicio 3, se ha definido el parámetro "m" como un *slider*, lo que significa que el estudiante puede ajustar el valor de "m" deslizando el slider hacia la izquierda o hacia la derecha. Por otro lado, el parámetro "b" ha sido establecido como un valor fijo, lo que implica que su valor no puede cambiarse y se mantiene constante durante el ejercicio.

Además, para mejorar la comprensión y la retroalimentación visual del ajuste lineal, se ha agregado un texto dinámico correspondiente a la función de la recta. Este texto se actualiza en tiempo real a medida que el estudiante mueve el/los *slider*s. De esta manera, el estudiante puede visualizar los cambios en la función lineal y comprender cómo afecta a la posición y la inclinación de la recta en el gráfico.

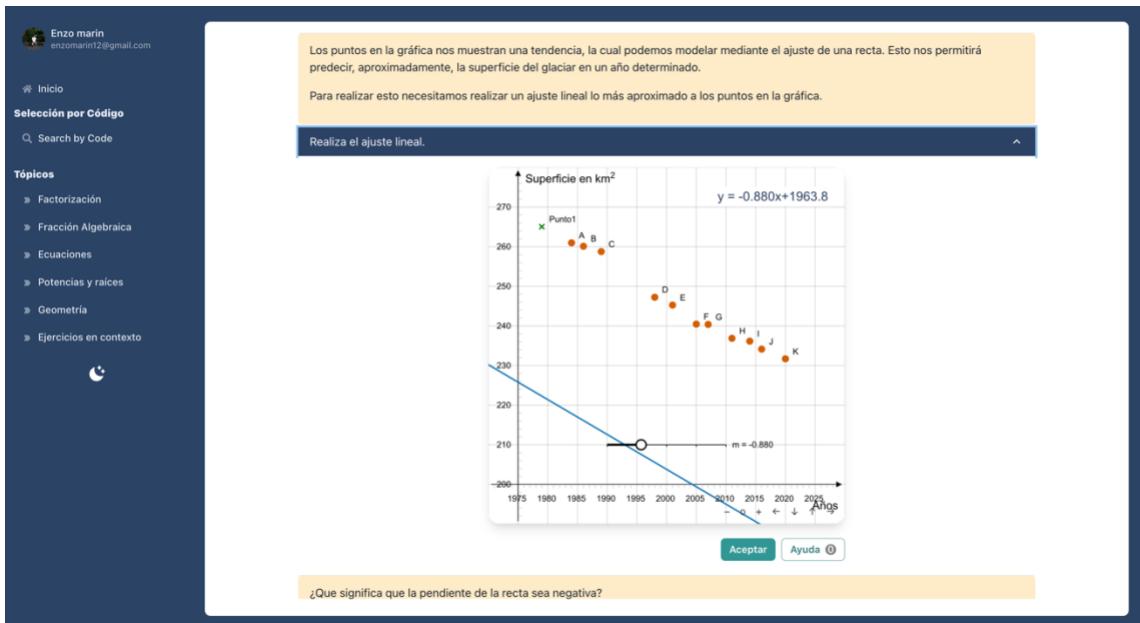
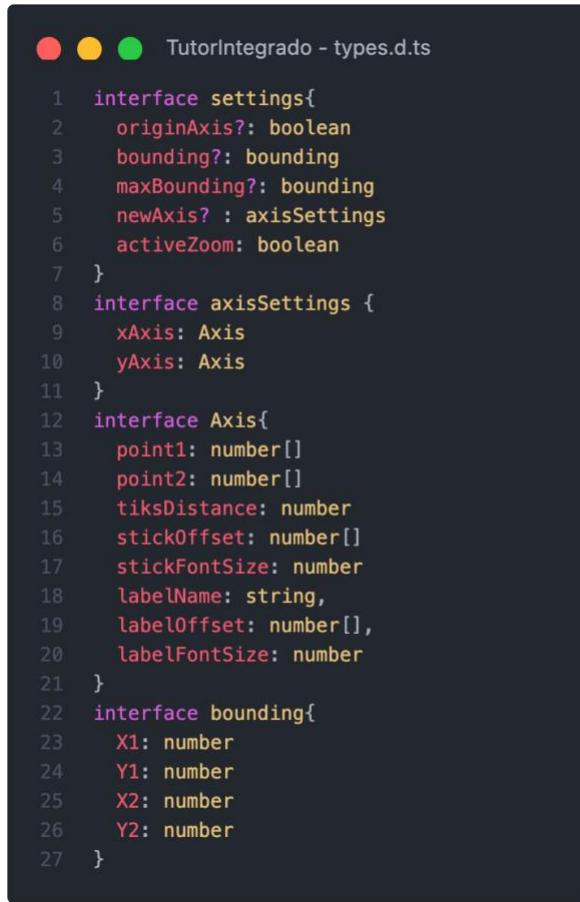


Figura 70: Interfaz del componente *linearFit*.

Para la implementación de estos componentes, se ha desarrollado un *custom hook* denominado “*useBoard*”. Este *custom hook* se encarga de realizar la inicialización y configuración del componente *JSXGraph* utilizado para la implementación de los gráficos interactivos en la plataforma. A continuación, se describen los parámetros que recibe este *custom hook*:

- *boardId*: Es una cadena de texto que representa el identificador único del gráfico *JSXGraph* que se va a crear. Este identificador se utiliza para referenciar el gráfico en el DOM⁴ y es necesario para renderizar correctamente el componente.
- *graphSettings*: Es un objeto que se define en la creación de los ejercicios y que contiene la configuración específica para el gráfico. Este objeto incluye varios atributos, los cuales se definen en la interface “*settings*” (ver Figura 71). Estos atributos permiten personalizar el comportamiento del gráfico, como el rango de visualización, la presencia o ausencia de ejes, el comportamiento del zoom, entre otros.

⁴ Las siglas DOM significan Document Object Model, o lo que es lo mismo, la estructura del documento HTML. Una página HTML está formada por múltiples etiquetas HTML, anidadas una dentro de otra, formando un árbol de etiquetas relacionadas entre sí, que se denomina árbol DOM (o simplemente DOM)



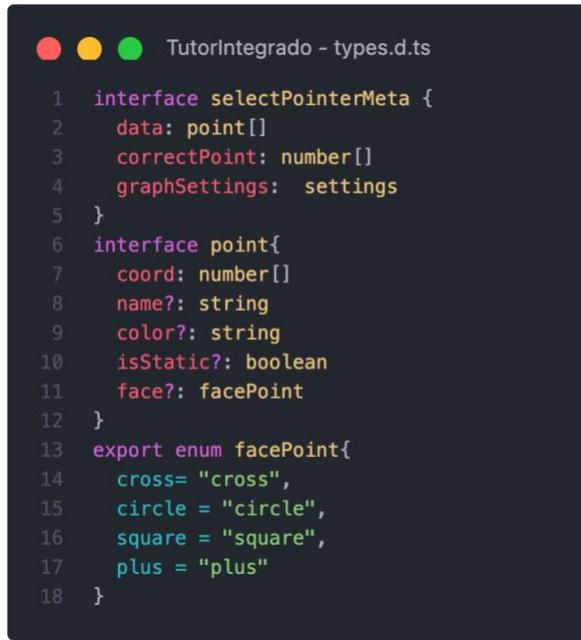
```
● ○ ● TutorIntegrado - types.d.ts
1 interface settings{
2   originAxis?: boolean
3   bounding?: bounding
4   maxBounding?: bounding
5   newAxis? : axisSettings
6   activeZoom: boolean
7 }
8 interface axisSettings {
9   xAxis: Axis
10  yAxis: Axis
11 }
12 interface Axis{
13   point1: number[]
14   point2: number[]
15   ticksDistance: number
16   stickOffset: number[]
17   stickFontSize: number
18   labelName: string,
19   labelOffset: number[],
20   labelFontSize: number
21 }
22 interface bounding{
23   X1: number
24   Y1: number
25   X2: number
26   Y2: number
27 }
```

Figura 71: *Interfaces* para la configuración del gráfico.

Una vez que el custom hook "*useBoard*" ha realizado la inicialización y configuración del gráfico JSXGraph con los parámetros proporcionados, este retorna un objeto que contiene ciertos valores relevantes para el despliegue del componente gráfico:

- *boardId*: El mismo valor que se recibió como parámetro, que representa el identificador único del gráfico.
- *boardRef*: Una referencia mutable que almacena la instancia del componente JSXGraph creado. Esta referencia permite interactuar con el gráfico y realizar acciones como la creación de puntos, líneas, funciones matemáticas, y otras características visuales y de interacción en el gráfico.
- *bgBoardColor*: Un valor que representa el color de fondo del gráfico. Este utiliza el *custom hook* “*useColorModeValue*” proporcionado por ChakraUI que puede recuperar un valor de los parámetros pasados en función del valor del modo de color activo (modo oscuro o modo claro).
- *disableBoard*: Una función que permite deshabilitar o suspender las actualizaciones del gráfico JSXGraph. Esto es útil en situaciones donde se desea detener la interacción con el gráfico o evitar actualizaciones innecesarias.

En cuanto al componente “*SelectPoint*”, la definición de su parámetro “*meta*” se define en la interfaz “*selectPointerMeta*” (ver Figura 72).



```
● ○ ● TutorIntegrado - types.d.ts
1 interface selectPointerMeta {
2   data: point[]
3   correctPoint: number[]
4   graphSettings: settings
5 }
6 interface point{
7   coord: number[]
8   name?: string
9   color?: string
10  isStatic?: boolean
11  face?: facePoint
12 }
13 export enum facePoint{
14   cross= "cross",
15   circle = "circle",
16   square = "square",
17   plus = "plus"
18 }
```

Figura 72: Interfaces para el componente *SelectPoint*.

La Figura 73 presenta el código del componente “*selectPoint*”. En este se utiliza el *hook* “*useState*” para almacenar las coordenadas del punto seleccionado por el estudiante en el gráfico y el *custom hook* “*useAlert*” para mostrar las alertas visuales al estudiante.

Para la configuración del componente *JSXGraph* y la creación del grafico interactivo, se utiliza el *custom hook* “*useBoard*” definido previamente. Este *hook* recibe el “*boardId*” proporcionado como primer argumento y las configuraciones del gráfico provenientes del campo “*graphSettings*” del parámetro “*meta*” definido al crear el ejercicio.

El primer *hook* “*useEffect*” se ejecuta una única vez debido al arreglo de dependencias vacío “[]”, lo que garantiza que la inicialización de los datos se realice solo una vez, evitando ciclos innecesarios de actualización.

Dentro de este *useEffect*, se recorre el arreglo de puntos “*data*” proporcionado en el parámetro “*meta*”. Cada punto es creado mediante el método *create* del objeto “*boardRef.current*”, que hace referencia al gráfico creado mediante el *custom hook* “*useBoard*”.

Para configurar cada punto en el gráfico, se extraen las propiedades relevantes del objeto “*point*” definido en el atributo “*data*” (cuya descripción detallada se proporcionó en el capítulo 4 del documento). Las propiedades extraídas incluyen la posición del punto en el plano cartesiano (coordenadas “*x*” e “*y*”). Además, se realiza una verificación de la

existencia de atributos opcionales como "name", "color", "face" e "isStatic". En los casos en que no se proveen valores para los atributos "face" e "isStatic", se asignan valores predeterminados de "circle" y "true", respectivamente. Estos atributos opcionales brindan la posibilidad de personalizar tanto la apariencia como el comportamiento de cada punto en el gráfico de manera flexible

Una vez que se crea un punto en el gráfico, se procede a añadir un evento "down" al punto. La activación de este evento se produce cuando el usuario hace clic en el punto. En el momento en que esto sucede, las coordenadas "x" e "y" de la posición del punto se almacenan en el estado "userAnswer". Esta acción posibilita el seguimiento de la respuesta proporcionada por el estudiante en el gráfico.

El segundo *useEffect* se encarga de comprobar si el punto seleccionado por el estudiante es correcto. Este *hook* se ejecuta cada vez que el estado "userAnswer" cambia, es decir, cada vez que el usuario selecciona un punto. En este *useEffect*, se realiza la comprobación correspondiente con el valor correcto y se despliegan las alertas correspondientes mediante el custom *hook* "useAlert".

Finalmente, el componente "SelectPoint" retorna un elemento "div" que corresponde al contenedor del componente gráfico. Esto se realiza estableciendo el "id" del elemento "div" con el valor de "boardId" obtenido del custom *hook* "useBoard". De esta manera, el gráfico interactivo se renderiza dentro del elemento "div" y se muestra al estudiante en la interfaz.

```

1 import { AlertStatus, facePoint, Hint, selectPointerMeta } from "../types.d"
2 import { useEffect, useState } from "react";
3 import { Flex } from "@chakra-ui/react";
4 import ResAlert from "../Alert/responseAlert";
5 import { useAlert } from "../hooks/useAlert";
6 import { useBoard } from "../hooks/useBoard";
7
8 interface Props{
9   meta: selectPointerMeta
10  hints: Hint[]
11 }
12 export const SelectPoint = ({meta, hints}: Props)>{
13   const {correctPoint, data} = meta
14   const [userAnswer, setUserAnswer] = useState([])
15   const {alertTitle, alertStatus, alertMsg, alertHidden, showAlert} = useAlert("", AlertStatus.info, "", false, 3000)
16   const {boardId, boardRef, bgBoardColor, disableBoard} = useBoard("jxgbox", meta.graphSettings)
17
18   // al utilizar useEffect S/D se ejecuta una unica vez despues del primer renderizado del componente
19   useEffect(()=>{    // inicialización de datos
20     data.forEach(point =>{
21       const isStatic = point?.isStatic ?? true
22       const face = point?.face ?? facePoint.circle
23       const p = boardRef.current.create("point", point.coord, {name: point?.name, color: point?.color, face: face, fixed: isStatic})
24       p.on("down", ()=> setUserAnswer(p.coords.usrCoords))
25     })
26   },[])
27
28   useEffect(()=>{
29     if(userAnswer.length !== 0){
30       const answer = userAnswer.slice(-2)
31       const isCorrect = answer.every((element, index)=> element === correctPoint[index])
32       if (isCorrect){
33         showAlert("!", AlertStatus.success, "Muy bien!", null)
34         if (boardRef.current){
35           disableBoard()
36           // JXG.JSXGraph.freeBoard(boardRef.current)
37         }
38       } else {
39         showAlert("!", AlertStatus.error, "Respuesta Incorrecta")
40         // TODO: DESPLEGAR HINTS
41       }
42     }
43   },[userAnswer])
44
45   return(
46     <Flex flexDirection="column" alignContent="center" flexWrap="wrap" width="100%" maxWidth="100%" >
47       <div id={boardId} style={{ width: "100%", height: "100%", position: "absolute", top: 0, left: 0 }}></div>
48       <ResAlert title={alertTitle} status={alertStatus} alertHidden={alertHidden} text={alertMsg}></ResAlert>
49     </Flex>
50   )
51 }

```

Figura 73: Código del componente *selectPoint*

El componente "*LinearFit*" se utiliza para crear un gráfico interactivo en el que se representa una función lineal y se permite al estudiante ajustar los valores de "m" y "b" de la ecuación de la función lineal mediante *sliders*. El objetivo del estudiante es ajustar los valores de "m" y/o "b" de manera que la función lineal (la recta) representada en el plano cartesiano se acerque lo más posible a una serie de puntos dados.

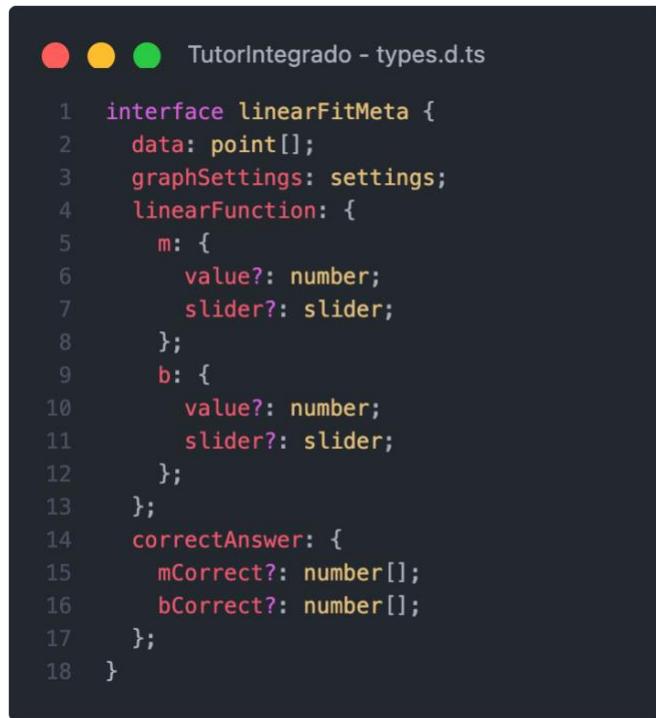
Al igual que el componente "*SelectPoint*", el componente "*LinearFit*" recibe dos parámetros: "*meta*" y "*hints*" (ver Figura 74). En este caso, el parámetro "*meta*" contiene la información necesaria para configurar el gráfico, incluyendo el conjunto de puntos a representar (*data*), información sobre los valores de "m" y "b" de la función lineal (*linearFunction*), los rangos de valores correctos para "m" y/o "b" (*correctAnswer*), y las

configuraciones del gráfico (*graphSettings*). La definición del parámetro “*meta*” se puede observar en la interfaz presentada en la Figura 75.



```
● ● ● TutorIntegrado - linearFit.tsx
1 interface Props{
2   meta : linearFitMeta
3   hints: Hint[]
4 }
5 export const LinearFit = ({meta, hints}: Props) =>{
6   // CODIGO DEL COMPONENTE
7 }
```

Figura 74: Definición de componente *LinearFit*.



```
● ● ● TutorIntegrado - types.d.ts
1 interface linearFitMeta {
2   data: point[];
3   graphSettings: settings;
4   linearFunction: {
5     m: {
6       value?: number;
7       slider?: slider;
8     };
9     b: {
10       value?: number;
11       slider?: slider;
12     };
13   };
14   correctAnswer: {
15     mCorrect?: number[];
16     bCorrect?: number[];
17   };
18 }
```

Figura 75: Interfaz *linearFitMeta*.

Dentro del componente “*LinearFit*”, se definen las variables “*mval*” y “*bval*” como funciones (ver Figura 76). Estas funciones se utilizan para obtener los valores actuales de “*m*” y “*b*” respectivamente.

Al igual que en el componente “*SelectPoint*”, en el componente “*LinearFit*” se utiliza el hook “*useEffect*” para realizar ciertas tareas una vez que el componente ha sido renderizado en el DOM, como la creación de los puntos proporcionados en el atributo “*data*” del parámetro “*meta*”.

Cada punto es creado mediante el método *create* del objeto “*boardRef.current*”, que hace referencia al gráfico creado mediante el *custom hook* “*useBoard*”. Además, se configuran las propiedades del punto como la posición en el gráfico (coordenadas “x” e “y”), el nombre del punto, el color, la forma del punto y si el punto es estático o no.

```

● ○ ● ● TutorIntegrado - linearFit.tsx
1 let mval: () => number
2 let bval: () => number
3
4 useEffect(()=>{
5   data.forEach(point =>{
6     const isStatic = point?.isStatic ?? true
7     const facePoint = point?.face ?? "circle"
8     boardRef.current.create("point", point.coord, {name: point?.name, color: point?.color, fixed: isStatic, face: facePoint })
9   })
10
11 // RESTO DEL CODIGO
12 },[])

```

Figura 76: Código para el despliegue de los datos del componente *LinearFit*.

Es importante destacar que a diferencia del componente “*SelectPoint*”, no se asigna un evento “*down*” a estos puntos. Esto se debe a que en este componente no se espera que el estudiante seleccione o interactúe directamente con los puntos del gráfico.

La Figura 77 presenta la lógica para la creación de los sliders en el componente “*LinearFit*”. En esta sección, se determina si los *sliders* para los valores de “m” y “b” deben ser creados y configurados según la información proporcionada en el atributo “*linearFunction*” del parámetro “*meta*”.

```

● ○ ● ● TutorIntegrado - linearFit.tsx
1 if(m.slider && b.slider){
2   const mSlider = createSlider(m.slider)
3   const bSlider = createSlider(b.slider)
4   mSliderRef.current = mSlider
5   bSliderRef.current = bSlider
6   mval = () => mSlider.Value()
7   bval = () => bSlider.Value()
8 }
9 else if (m.slider && !b.slider){
10  const mSlider = createSlider(m.slider)
11  mSliderRef.current = mSlider
12  mval = () => mSlider.Value()
13  bval = () => b?.value
14 }
15 else if (!m.slider && b.slider){
16  const bSlider = createSlider(b.slider)
17  bSliderRef.current = bSlider
18  mval = () => m?.value
19  bval = () => bSlider.Value()
20 }

```

Figura 77: Código para la creación de los *sliders* del componente *LinearFit*.

En el primer caso, si tanto "*m.slider*" como "*b.slider*" existen y son verdaderos, se crean dos *sliders* separados para "m" y "b". Luego, se almacenan las referencias de estos *sliders* en las variables "*mSliderRef*" y "*bSliderRef*". Finalmente, se definen las funciones "mval" y "bval" para que retornen los valores actuales de "m" y "b" obtenidos de los *sliders* respectivos mediante *mSlider.Value()* y *bSlider.Value()*.

En el segundo caso, si solo "*m.slider*" es verdadero y "*b.slider*" es falso, se crea solo el *slider* para "m". Se almacena la referencia del *slider* en "*mSliderRef*", y se define "mval" de la misma forma que en el caso anterior. Sin embargo, en este caso, "bval" se configura para que retorne el valor fijo de "b" proporcionado en el atributo "linearFunction" del parámetro "meta".

En el tercer caso, si "*m.slider*" es falso y "*b.slider*" es verdadero, se crea solo el *slider* para "b". Se almacena la referencia del *slider* en "*bSliderRef*", y se define "bval" de manera similar a los casos anteriores, retornando el valor actual del *slider* para "b". Sin embargo, "mval" se configura para que retorne el valor fijo de "m" proporcionado en el atributo "linearFunction" del parámetro "meta".

Estas condiciones aseguran que los *sliders* se creen y se configuren correctamente según los valores proporcionados en la *prop* "meta", permitiendo ajustar los valores de "m" y/o "b" de la función lineal según las necesidades del ejercicio.

Luego de la creación y configuración de los *sliders*, se procede a crear la función lineal representada por la ecuación de la recta " $y = mx + b$ " utilizando los valores de "m" y "b" obtenidos de los *sliders* o de los valores constantes proporcionados en la *prop* "meta". Esto se realiza utilizando una función anónima que toma como argumento "x" y retorna el valor de "y" de acuerdo con la ecuación de la recta. La Figura 78 representa cómo se crea la función lineal y se despliega utilizando el método "create" del objeto "*boardRef.current*":

```

1 const linF = function(x){return mval()*x + bval()}
2 var G = boardRef.current.create('functiongraph',[linF], {strokeWidth: 2});

```

Figura 78: Código para la definición y despliegue de la función lineal.

Finalmente, el componente devuelve un elemento "*div*" que contiene el gráfico interactivo creado con *JSXGraph*, junto con un botón "Aceptar" que permite al estudiante comprobar su respuesta. Esto se logra asignando la función "checkAnswer" al atributo "onClick" del botón. Esta función verifica si el estudiante ha ajustado correctamente los valores de "m" y/o "b". Si la respuesta es correcta, se muestra una alerta de éxito y se desactiva el botón "Aceptar", evitando cambios adicionales en el gráfico. Por el contrario, si la respuesta es incorrecta, se muestra una alerta indicando que cometió un error.

5.3.3 Revisión de sprint

Durante la revisión de este sprint, se verificó la correcta ejecución de las tareas planificadas y se llevó a cabo una evaluación de los componentes gráficos desarrollados. En esta revisión, se identificó un aspecto específico relacionado con la presentación visual y la retroalimentación de los usuarios, enfocado en el componente “*selectPoint*”. En esta instancia, se solicitó que al elegir un punto en dicho componente, este se destaca visualmente mediante un cambio de color o tamaño, con el objetivo de proporcionar una clara distinción del punto seleccionado por el estudiante en el gráfico.

5.4 Sprint 4

El enfoque principal del cuarto sprint se centra en la implementación de las pistas (*hints*). En esta sección, se describe en detalle cómo funcionan los *hints* y cómo se llevó a cabo la implementación.

5.4.1 Sprint backlog

- Los *hints* se deben desbloquear cuando el estudiante comete un error.
- El sistema debe notificar visualmente cuando un nuevo *hint* es desbloqueado.
- El sistema debe permitir al estudiante desplazarse entre la lista de todos los *hints* desbloqueados.

5.4.2 Product increment

La Figura 79 muestra un ejemplo de cómo se despliega el primer hint de un paso. Algunos elementos importantes que se pueden observar en la interfaz de los hints son:

- Cuerpo del hint: Aquí se muestra el contenido del hint, proporcionando información adicional o consejos para ayudar al estudiante a resolver el ejercicio.
- Texto indicador: Se muestra un texto que indica el número actual del hint y el total de hints desbloqueados, lo que brinda una referencia clara sobre la cantidad de hints disponibles.
- Botones de desplazamiento: Se incluyen botones para navegar al hint previo y al siguiente. Estos botones se activan o desactivan de manera dinámica según el hint actual y el total de hints desbloqueados, lo que evita que el estudiante se mueva fuera del rango de hints disponibles

Si la resistencia de cada sensor es de 100 ohm, es decir $R1=R2=R3 = 100$ ohm

1. ¿Cuál es la resistencia total del circuito de sensores en serie? ^

En un circuito en serie la resistencia total del circuito se calcula sumando las resistencias individuales

$$R_{serie} = R1 + R2 + R3$$

Calcular resistencia en serie

$$R_{serie} = \boxed{} + \boxed{} + \boxed{}$$

$$R_{serie} = \boxed{} \text{ ohm}$$

Pista:

Ingresá los valores de las resistencias y realiza la suma

Pista 1 de 1

Aceptar

Ayuda 0

Figura 79: Visualización de *Hint* inicial.

Cuando el estudiante responde incorrectamente un paso y hay hints por desbloquear, el sistema notifica al estudiante sobre la disponibilidad de un nuevo hint mediante el botón "Ayuda". En esta notificación, el botón se resalta con un texto y borde de color rojo, y el número indicador de hints desbloqueados se actualiza para reflejar la cantidad de nuevos hints disponibles. La Figura 80 ejemplifica cómo se muestra esta notificación cuando un nuevo hint es desbloqueado por el sistema.

Si la resistencia de cada sensor es de 100 ohm, es decir $R1=R2=R3 = 100$ ohm

1. ¿Cuál es la resistencia total del circuito de sensores en serie? ^

En un circuito en serie la resistencia total del circuito se calcula sumando las resistencias individuales

$$R_{serie} = R1 + R2 + R3$$

Calcular resistencia en serie ^

$$R_{serie} = \boxed{1} + \boxed{2} + \boxed{3}$$

$$R_{serie} = \boxed{4} \text{ ohm}$$

Aceptar

Ayuda 1

! 😞 Respuesta Incorrecta

Figura 80: Notificación al desbloquear nuevo *hint*.

Conforme el estudiante continúa respondiendo incorrectamente, el sistema gradualmente desbloquea más pistas hasta llegar a la última pista definida en la creación del ejercicio, la cual contiene la respuesta correcta esperada del estudiante.

Es importante destacar que el cuerpo de los hints admite la sintaxis *LaTeX*, lo que permite mostrar información de manera matemáticamente precisa y legible. Esta capacidad de utilizar *LaTeX* en los hints permite brindar explicaciones más detalladas y claras, especialmente en ejercicios que involucran conceptos matemáticos.

La Figura 81 ejemplifica cómo se utiliza esta capacidad en el último hint de un paso. En dicho hint, se presenta la expresión completa con los valores correctos resaltados en color verde. Esto facilita la identificación de los valores correctos que se esperaba que fueran ingresados.

Si la resistencia de cada sensor es de 100 ohm, es decir $R1=R2=R3 = 100$ ohm

1. ¿Cuál es la resistencia total del circuito de sensores en serie?

En un circuito en serie la resistencia total del circuito se calcula sumando las resistencias individuales

$$R_{serie} = R1 + R2$$

Calcular resistencia en serie

$$R_{serie} = [1] + [2] + [3]$$
$$R_{serie} = [4] \text{ ohm}$$

Pista:

La respuesta es:

$$R_{serie} = 100 + 100 + 100$$

$$R_{serie} = 300 \text{ ohm}$$

Pista 2 de 2

Aceptar

Ayuda 0

Figura 81: Visualización del último hint.

Para la implementación de los hints, al igual que con las alertas, se ha separado la lógica de la visualización. Para gestionar la lógica del despliegue de los hints, se implementó un custom hook llamado "*useHint*". Este *hook* se encarga de manejar el estado relacionado con los *hints*, como cuántos han sido desbloqueados, cuál es el hint actual, si se pueden desplazar a pistas anteriores o siguientes, entre otros.

El custom hook *useHint* recibe como parámetro los hints disponibles, los cuales son separados en dos arreglos: "*genericHints*" y "*specificHints*". El arreglo "*genericHints*" contiene los hints genéricos, mientras que el arreglo "*specificHints*" almacena los hints asociados a una respuesta específica.

El *custom hook* utiliza varios estados para mantener la información relevante:

- *idUnlockedHint*: Este estado se utiliza para hacer un seguimiento del índice del próximo hint a desbloquear. Se inicializa en 1 y se actualiza cada vez que se desbloquea un nuevo hint.
- *unlockedHints*: Es un estado que contiene un *array* de objetos, representando los *hints* desbloqueados hasta el momento. Se inicializa con el primer *hint*, ya que este siempre está desbloqueado desde el inicio.
- *currentHint*: Es un estado que indica el índice del *hint* actual que se está mostrando. Se inicializa en 0, que corresponde al primer *hint* desbloqueado. Este estado se actualiza cuando el estudiante navega entre los hints previos y siguientes.
- *totalHints*: Representa la cantidad total de hints disponibles en el ejercicio. Se inicializa con el valor de “*hints.length*”.
- *disabledPrevButton* y *disabledNextButton*: Son estados booleanos que indican si los botones de navegación hacia el *hint* previo y siguiente, respectivamente, están deshabilitados.
- *numHintsActivated*: Es un estado que lleva el registro del número de *hints* que el estudiante ha activado. Se utiliza para mostrar el número de *hints* desbloqueados en el botón de "Ayuda" y se actualiza cada vez que se desbloquea un nuevo *hint*.
- *unlockedAnswerIds*: Es un estado que contiene un arreglo con los *ids* de las respuestas para las que ya se han desbloqueado los *hints* específicos asociados. Esto se utiliza para evitar desbloquear el mismo *hint* específico varias veces.

En la Figura 82 se muestra el código del custom hook "*useHint*", en donde se define la lógica y los estados mencionados anteriormente. Adicionalmente, en el código se definen las funciones para interactuar con los *hints*, como “*nextHint*” y “*prevHint*” para desplazarse entre los *hints* previos y siguientes, y la función “*unlockHint*” para desbloquear nuevos *hints* en función de las respuestas del estudiante.

La función *unlockHint* toma como parámetro *answerId*, que representa el id de la respuesta asociada al *hint* específico que se desea desbloquear. Si *answerId* no se proporciona (es *undefined*), la función desbloqueará el siguiente *hint* genérico disponible.

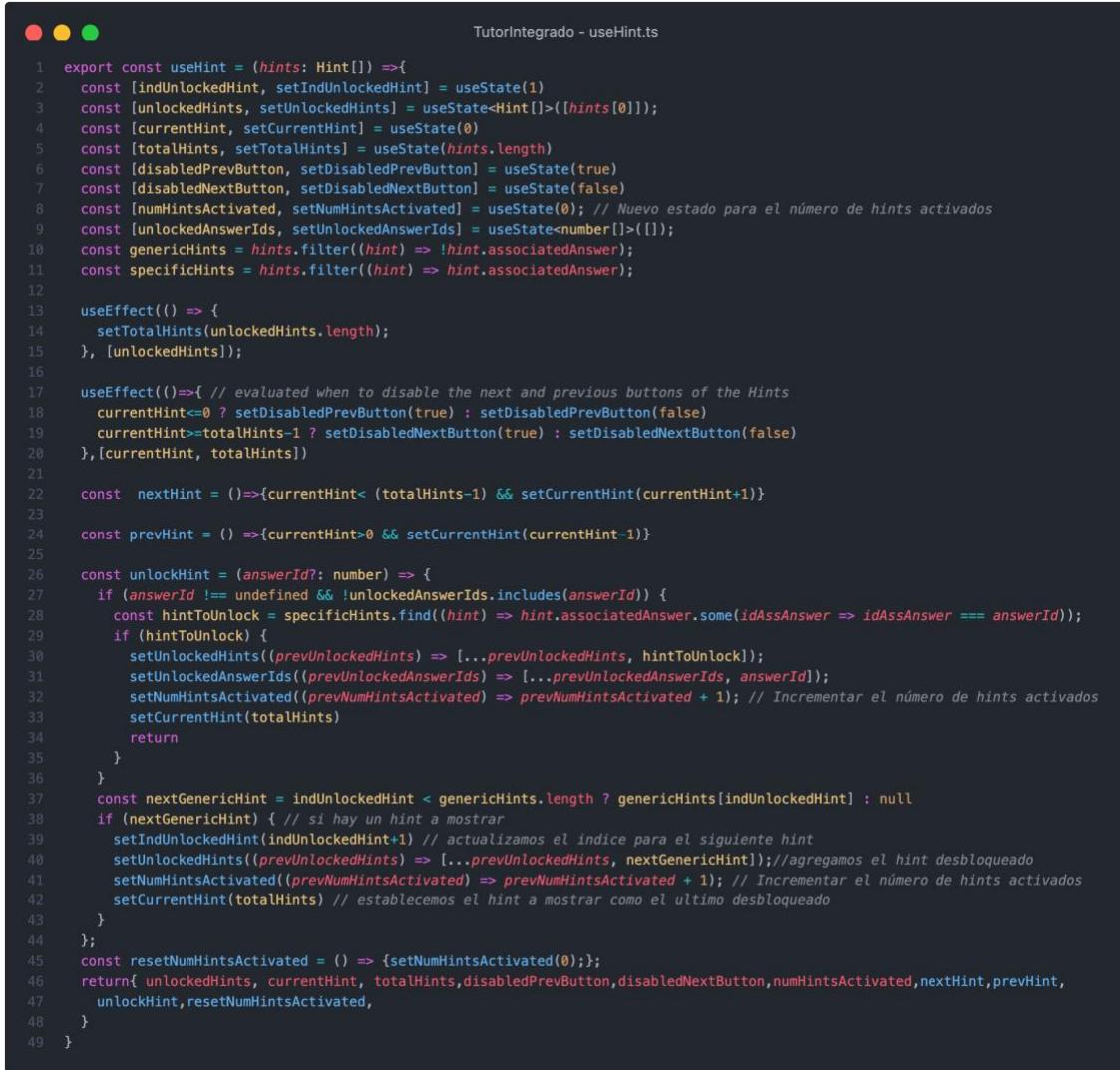
Para lo anterior, la función verifica si *answerId* no es *undefined* y si el id de respuesta no está incluido en el estado *unlockedAnswerIds*. Si se cumple esta condición, significa que el *hint* específico aún no ha sido desbloqueado para la respuesta proporcionada, y procede a buscar el *hint* específico correspondiente a esa respuesta.

Si se encuentra el *hint* específico, se agrega el *hint* al estado *unlockedHints*, y se agrega el id de respuesta al estado *unlockedAnswerIds* para evitar desbloquear el mismo *hint* específico varias veces.

En caso de no encontrar un *hint* específico para la respuesta proporcionada, la función busca el próximo *hint* genérico disponible en el arreglo *genericHints* y lo agrega al estado *unlockedHints*.

Luego de desbloquear un nuevo *hint*, la función actualiza el estado *idUnlockedHint* para hacer un seguimiento del próximo *hint* a desbloquear. También incrementa el estado *numHintsActivated* para llevar el registro del número de *hints* que se han desbloqueado.

Por último, la función establece el *hint* recién desbloqueado como el *hint* actual, permitiendo que este se visualice en la interfaz.



```

1  export const useHint = (hints: Hint[]) =>{
2    const [indUnlockedHint, setIndUnlockedHint] = useState(1)
3    const [unlockedHints, setUnlockedHints] = useState<Hint[]>([hints[0]]);
4    const [currentHint, setCurrentHint] = useState(0)
5    const [totalHints, setTotalHints] = useState(hints.length)
6    const [disabledPrevButton, setDisabledPrevButton] = useState(true)
7    const [disabledNextButton, setDisabledNextButton] = useState(false)
8    const [numHintsActivated, setNumHintsActivated] = useState(0); // Nuevo estado para el número de hints activados
9    const [unlockedAnswerIds, setUnlockedAnswerIds] = useState<number[]>([]);
10   const genericHints = hints.filter((hint) => !hint.associatedAnswer);
11   const specificHints = hints.filter((hint) => hint.associatedAnswer);
12
13  useEffect(() => {
14    setTotalHints(unlockedHints.length);
15  }, [unlockedHints]);
16
17  useEffect(()=>{ // evaluated when to disable the next and previous buttons of the Hints
18    currentHint<=0 ? setDisabledPrevButton(true) : setDisabledPrevButton(false)
19    currentHint>=totalHints-1 ? setDisabledNextButton(true) : setDisabledNextButton(false)
20  },[currentHint, totalHints]);
21
22  const nextHint = ()=>{currentHint< (totalHints-1) && setCurrentHint(currentHint+1)}
23
24  const prevHint = () =>{currentHint>0 && setCurrentHint(currentHint-1)}
25
26  const unlockHint = (answerId?: number) => {
27    if (answerId !== undefined && !unlockedAnswerIds.includes(answerId)) {
28      const hintToUnlock = specificHints.find((hint) => hint.associatedAnswer.some(idAssAnswer => idAssAnswer === answerId));
29      if (hintToUnlock) {
30        setUnlockedHints((prevUnlockedHints) => [...prevUnlockedHints, hintToUnlock]);
31        setUnlockedAnswerIds((prevUnlockedAnswerIds) => [...prevUnlockedAnswerIds, answerId]);
32        setNumHintsActivated((prevNumHintsActivated) => prevNumHintsActivated + 1); // Incrementar el número de hints activados
33        setCurrentHint(totalHints)
34        return
35      }
36    }
37    const nextGenericHint = indUnlockedHint < genericHints.length ? genericHints[indUnlockedHint] : null
38    if (nextGenericHint) { // si hay un hint a mostrar
39      setIndUnlockedHint(indUnlockedHint+1) // actualizamos el indice para el siguiente hint
40      setUnlockedHints((prevUnlockedHints) => [...prevUnlockedHints, nextGenericHint]);//agregamos el hint desbloqueado
41      setNumHintsActivated((prevNumHintsActivated) => prevNumHintsActivated + 1); // Incrementar el número de hints activados
42      setCurrentHint(totalHints) // establecemos el hint a mostrar como el ultimo desbloqueado
43    }
44  };
45  const resetNumHintsActivated = () => {setNumHintsActivated(0)};
46  return{ unlockedHints, currentHint, totalHints, disabledPrevButton, disabledNextButton, numHintsActivated, nextHint, prevHint,
47         unlockHint, resetNumHintsActivated,
48       }
49 }

```

Figura 82: Código del custom hook *useHint*.

Finalmente, el custom hook *useHint* retorna un objeto con las propiedades y funciones relevantes para interactuar con los *hints* y obtener información necesaria para su

despliegue, las cuales son requeridos por el componente encargado de la visualización de los mismos.

Para la visualización de los *hints* se ha creado el componente "*HintButton*", que gestiona la interacción y despliegue de las pistas disponibles en el ejercicio. Este componente utiliza el componente "*popover*" proporcionado por ChakraUI para desplegar la interfaz de las pitas.

El componente "*HintButton*" recibe varias propiedades esenciales para su funcionamiento, proporcionadas por el *custom hook* "*useHint*", tales como "*hints*", "*currentHint*", "*totalHints*", "*disabledPrevButton*", "*disabledNextButton*", "*numEnabledHints*" y las funciones "*nextHint*", "*prevHint*" y "*resetNumHintsActivated*". En la Figura 83, se puede apreciar la definición de este componente.



```
● ○ ● TutorIntegrado - hint.tsx
  1 interface Props{
  2   hints : Hint[]
  3   currentHint: number
  4   totalHints: number
  5   nextHint: () => void
  6   prevHint: () => void
  7   disabledPrevButton : boolean
  8   disabledNextButton : boolean
  9   numEnabledHints: number
 10   resetNumHintsActivated: ()=> void
 11 }
 12 const HintButton = (
 13   {hints,
 14   currentHint,
 15   totalHints,
 16   nextHint,
 17   prevHint,
 18   disabledPrevButton,
 19   disabledNextButton,
 20   numEnabledHints = 0,
 21   resetNumHintsActivated
 22 }: Props) => {
 23   const handleClick = () =>{
 24     resetNumHintsActivated()
 25   }
 26   return (
 27     <Popover placement="bottom" closeOnBlur={false}>
 28       // RESTO DEL CODIGO
 29     </Popover>
 30   )
 31 }
 32 export default HintButton
```

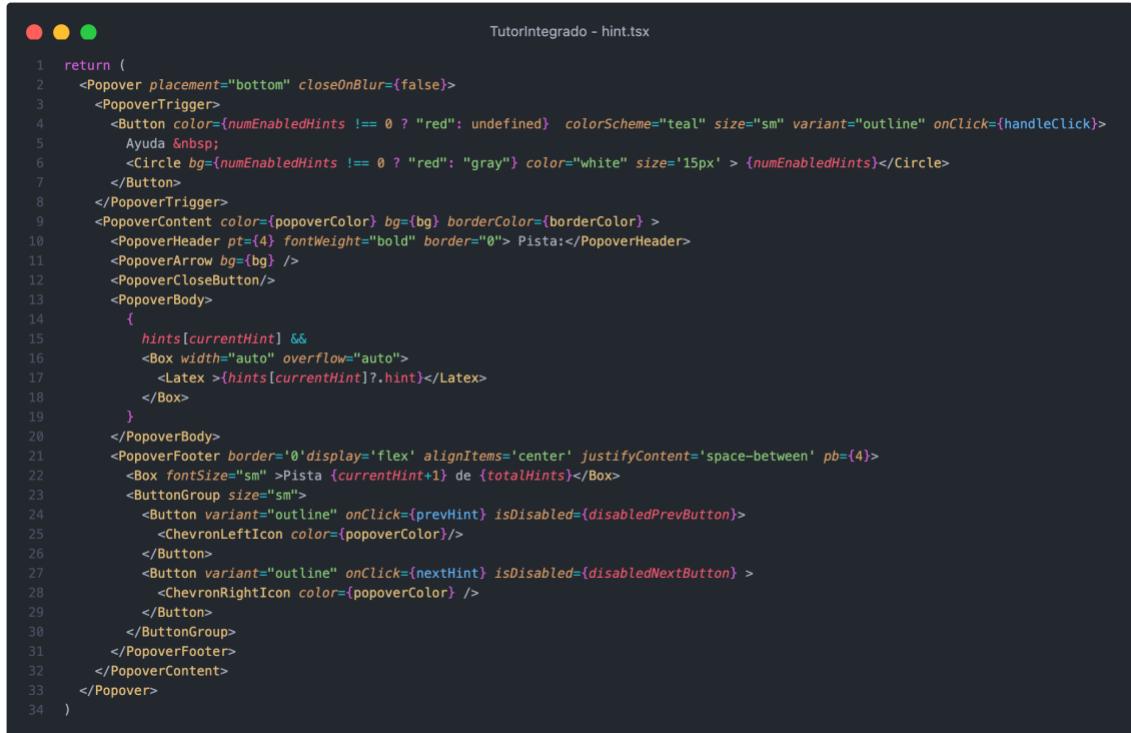
Figura 83: Definición de componente *HintButton*.

El componente "*HintButton*" retorna un conjunto de componentes, tal como se muestra en la Figura 84. Uno de los componentes destacados es "*popover*", que se activa al hacer clic en el botón de "Ayuda". Este "*popover*" despliega un recuadro que contiene la pista actual

obtenida de la propiedad "`hints[currentHint].hint`". Es decir, muestra la información y ayuda específica correspondiente a la pista actual del ejercicio.

Además, en el botón de "Ayuda", el componente muestra un círculo rojo con un número que indica la cantidad de pistas desbloqueadas. Este número se actualiza dinámicamente a medida que el estudiante desbloquea más pistas, gracias a la propiedad "`numEnabledHints`".

El componente también retorna dos botones, en los cuales se les asignan las funciones "`prevHint`" y "`nextHint`" a través de la propiedad "`onClick`". Estos botones permiten al estudiante desplazarse entre las pistas previas y siguientes, respectivamente.



```

● ● ● Tutorintegrado - hint.tsx
1  return (
2    <Popover placement="bottom" closeOnBlur={false}>
3      <PopoverTrigger>
4        <Button color={numEnabledHints !== 0 ? "red": undefined} colorScheme="teal" size="sm" variant="outline" onClick={handleClick}>
5          Ayuda &nbsp;
6          <Circle bg={numEnabledHints === 0 ? "red": "gray"} color="white" size='15px' > {numEnabledHints}</Circle>
7        </Button>
8      </PopoverTrigger>
9      <PopoverContent color={popoverColor} bg={bg} borderColor={borderColor} >
10        <PopoverHeader pt={4} fontWeight="bold" border="0"> Pista:</PopoverHeader>
11        <PopoverArrow bg={bg} />
12        <PopoverCloseButton/>
13        <PopoverBody>
14          {
15            hints[currentHint] &&
16            <Box width="auto" overflow="auto">
17              <Latex >{hints[currentHint].hint}</Latex>
18            </Box>
19          }
20        </PopoverBody>
21        <PopoverFooter border='0' display='flex' alignItems='center' justifyContent='space-between' pb=(4)>
22          <Box fontSize="sm" >Pista {currentHint+1} de {totalHints}</Box>
23          <ButtonGroup size="sm">
24            <Button variant="outline" onClick={prevHint} isDisabled={disabledPrevButton}>
25              <ChevronLeftIcon color={popoverColor}/>
26            </Button>
27            <Button variant="outline" onClick={nextHint} isDisabled={disabledNextButton}>
28              <ChevronRightIcon color={popoverColor} />
29            </Button>
30          </ButtonGroup>
31        </PopoverFooter>
32      </PopoverContent>
33    </Popover>
34  )

```

Figura 84: Retorno del componente *HintButton*.

5.4.3 Revisión de sprint

En la revisión de este sprint, se verificó que los *hints* se desbloquearan de manera adecuada cuando el estudiante cometiera errores y que el sistema notificara claramente la disponibilidad de nuevos hints. Además, se comprobó que se pudiera mover entre los hints desbloqueados de manera correcta.

En particular, se prestó atención a la presentación visual de los hints, asegurándose de que los consejos fueran legibles y que cualquier uso de la sintaxis *LaTeX* funcionara correctamente.

En la revisión, no se identificaron problemas significativos con la implementación de los *hints* y los componentes asociados. Por lo que se pudo validar que los objetivos del *sprint* se cumplieron satisfactoriamente.

5.5 Sprint 5

El enfoque principal de este sprint reside en satisfacer el último requisito visual destinado al tutor. En esta etapa, se aborda la implementación de la secuencialidad en la presentación de preguntas y pasos dentro de un ejercicio. El objetivo es establecer una estructura en la que tanto las preguntas como los pasos estén inicialmente bloqueados, a excepción de la primera pregunta y su primer paso. Esto significa que, en un principio, el estudiante solo podrá acceder y responder al primer paso de la primera pregunta. Una vez que se resuelva correctamente este paso, se irán desbloqueando gradualmente los pasos y preguntas subsiguientes. De esta manera, se logra una progresión secuencial que guía al estudiante a través del ejercicio de manera estructurada y coherente.

5.5.1 Sprint backlog

- El sistema debe permitir desbloquear secuencialmente las preguntas y los pasos a medida que el estudiante avanza y completa correctamente cada paso.
- Inicialmente, solo se mostrará la primera pregunta y su primer paso, permitiendo que el estudiante los visualice y resuelva.
- Una vez que el estudiante completa el último paso de una pregunta, se debe desbloquear automáticamente la siguiente pregunta y su primer paso.

5.5.2 Product increment

Para abordar las tareas definidas en este *sprint*, se decidió implementar un enfoque basado en el uso de estados globales para rastrear tanto la pregunta actual como el paso actual en el ejercicio. La utilización de un estado global se refiere a la adopción de variables o conjuntos de variables que pueden ser accedidos y modificados desde cualquier parte de la aplicación, lo que simplifica la gestión de información compartida.

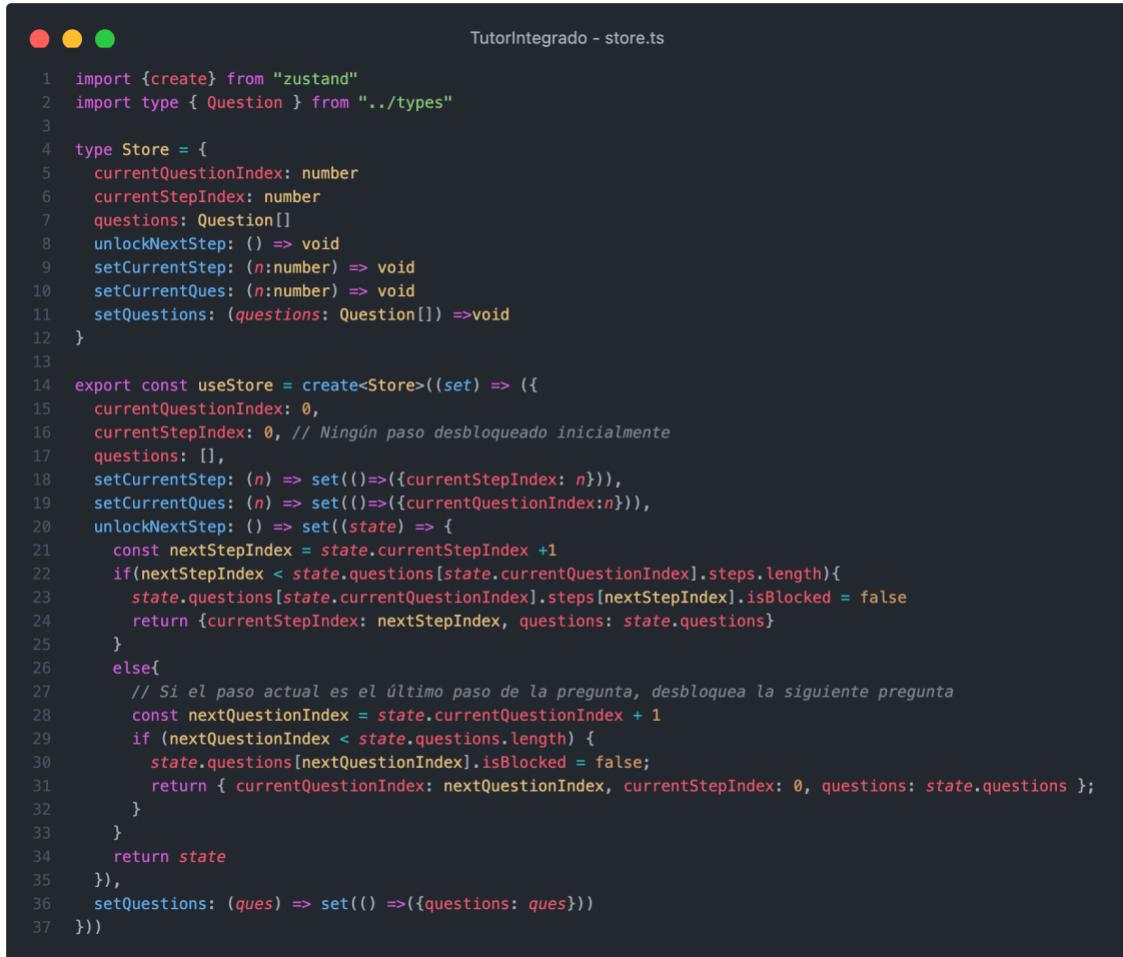
En este contexto, se optó por implementar "*Zustand*", una biblioteca que proporciona una solución efectiva para manejar el estado global en aplicaciones *React* (*Zustand*, s.f).

Para llevar a cabo la implementación del estado global utilizando "*Zustand*", se ha creado un archivo denominado "*store.ts*". En este archivo, se ha definido la constante "*useStore*", que constituye una instancia de "*Store*" en "*Zustand*".

En el contexto de *Zustand*, una *Store* es una entidad que encapsula y gestiona el estado de una aplicación de *React* de manera centralizada y accesible desde cualquier componente. Funciona como un contenedor para el estado y proporciona un conjunto de funciones para acceder y modificar ese estado de manera segura.

La *Store* en *Zustand* es creada mediante la función "*create*", que recibe una función *callback* como argumento. Esta función define las propiedades y métodos que conformarán el estado y las operaciones permitidas sobre él. Una vez creada la *Store*, puede ser utilizada como un *hook* por cualquier componente que necesite acceder a su estado compartido.

La Figura 85 presenta el código para la implementación del *hook* “*useStore*” utilizando *Zustand*.



```

1 import {create} from "zustand"
2 import type { Question } from "../types"
3
4 type Store = {
5   currentQuestionIndex: number
6   currentStepIndex: number
7   questions: Question[]
8   unlockNextStep: () => void
9   setCurrentStep: (n:number) => void
10  setCurrentQues: (n:number) => void
11  setQuestions: (questions: Question[]) =>void
12 }
13
14 export const useStore = create<Store>((set) => ({
15   currentQuestionIndex: 0,
16   currentStepIndex: 0, // Ningún paso desbloqueado inicialmente
17   questions: [],
18   setCurrentStep: (n) => set(()=>({currentStepIndex: n})),
19   setCurrentQues: (n) => set(()=>({currentQuestionIndex:n})),
20   unlockNextStep: () => set((state) => {
21     const nextStepIndex = state.currentStepIndex +1
22     if(nextStepIndex < state.questions[state.currentQuestionIndex].steps.length){
23       state.questions[state.currentQuestionIndex].steps[nextStepIndex].isBlocked = false
24       return {currentStepIndex: nextStepIndex, questions: state.questions}
25     }
26     else{
27       // Si el paso actual es el último paso de la pregunta, desbloquea la siguiente pregunta
28       const nextQuestionIndex = state.currentQuestionIndex + 1
29       if (nextQuestionIndex < state.questions.length) {
30         state.questions[nextQuestionIndex].isBlocked = false;
31         return { currentQuestionIndex: nextQuestionIndex, currentStepIndex: 0, questions: state.questions };
32       }
33     }
34     return state
35   }),
36   setQuestions: (ques) => set(() =>({questions: ques}))
37 }))

```

Figura 85: Código del *hook* *useStore* para el manejo de los estados globales.

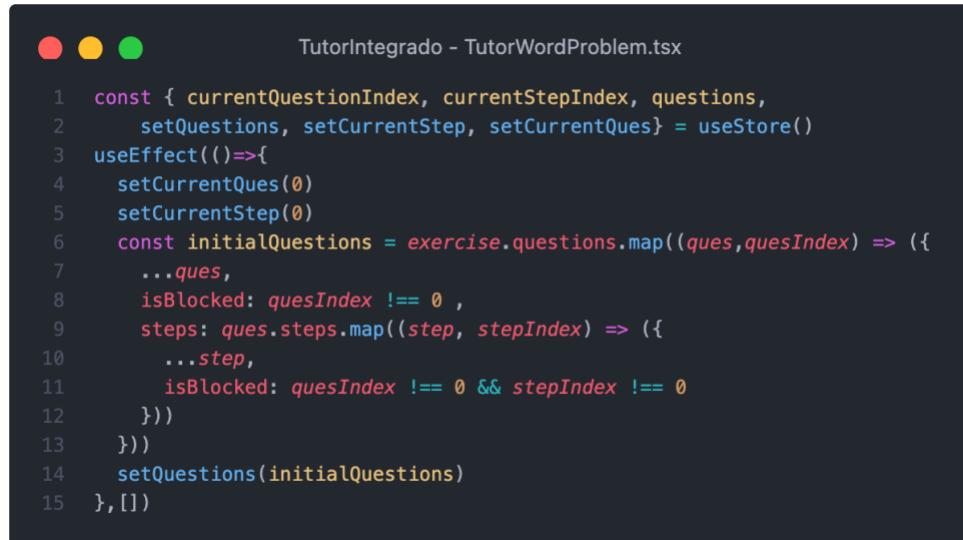
Este *hook* se encarga de gestionar diferentes aspectos de la secuencialidad en la presentación de las preguntas y pasos en el ejercicio. Las propiedades y funciones clave definidas en el *hook* *useStore* son las siguientes:

- *currentQuestionIndex*: Estado que almacena el índice de la pregunta actual que el estudiante está resolviendo.

- *currentStepIndex*: Estado que almacena el índice del paso actual de la pregunta actual.
- *questions*: Almacena un arreglo de objetos "Question" que representan las preguntas y sus pasos asociados en el ejercicio.
- *setCurrentStep*: Función que permite actualizar el valor de "currentStepIndex".
- *setCurrentQues*: Función que permite actualizar el valor de "currentQuestionIndex".
- *unlockNextStep*: Función que se encarga de desbloquear el siguiente paso una vez que el estudiante haya completado correctamente el paso actual. Si el paso actual es el último paso de la pregunta, se desbloqueará la siguiente pregunta junto con su primer paso.
- *setQuestions*: Función que actualiza el estado "questions" con el arreglo de preguntas proporcionado.

Una vez que el hook "*useStore*" ha sido definido, se vuelve accesible desde cualquier componente de la aplicación, lo que permite acceder a las propiedades y funciones previamente establecidas en dicho componente.

Para utilizarlo, simplemente es necesario importarlo en los componentes donde sea necesario su uso. En la Figura 86, se muestra una porción del código del componente "*TutorWordProblem*", donde se hace uso de este hook "*useStore*" para establecer los valores iniciales de los estados.



```

TutorIntegrado - TutorWordProblem.tsx

1  const { currentQuestionIndex, currentStepIndex, questions,
2      setQuestions, setCurrentStep, setCurrentQues } = useStore()
3  useEffect(()=>{
4      setCurrentQues(0)
5      setCurrentStep(0)
6      const initialQuestions = exercise.questions.map((ques, quesIndex) => ({
7          ...ques,
8          isBlocked: quesIndex !== 0 ,
9          steps: ques.steps.map((step, stepIndex) => ({
10             ...step,
11             isBlocked: quesIndex !== 0 && stepIndex !== 0
12         }))
13     }))
14     setQuestions(initialQuestions)
15 },[])

```

Figura 86: Inicialización de estados para el control de las preguntas y pasos.

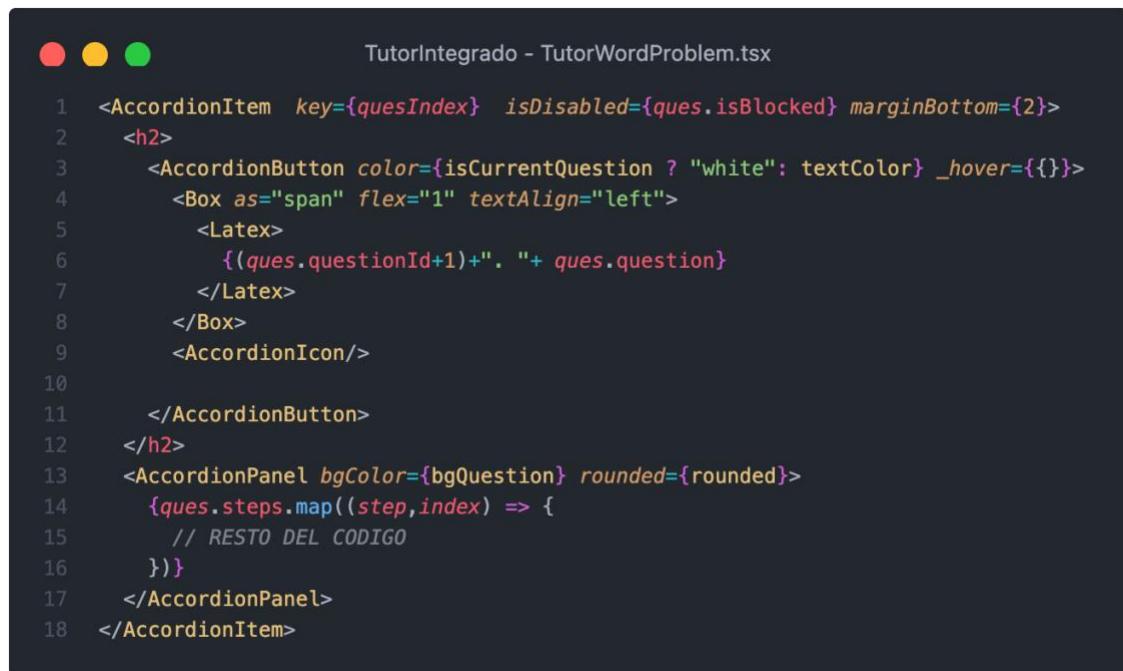
En este fragmento de código, se establecen los valores iniciales de la pregunta y el paso actual en cero mediante las funciones "*setCurrentQues*" y "*setCurrentStep*".

Adicionalmente, es importante señalar que se ha realizado una modificación en el objeto de preguntas original, agregando el atributo "*isBlocked*", tanto a las preguntas como a los pasos. Este atributo es una variable booleana que controla si una pregunta o paso deben estar bloqueados o no.

Para inicializar este atributo, se itera a través del arreglo de preguntas con el propósito de establecer un valor inicial para el atributo "*isBlocked*". En este proceso, se asigna el valor "*true*" al atributo, excepto para la primera pregunta y el primer paso de cada pregunta. El resultado de esta iteración se almacena en la constante "*initialQuestions*", la cual se utiliza como argumento en la función "*setQuestions*" para actualizar el estado "*questions*".

Una vez actualizado el estado, para bloquear tanto las preguntas como los pasos que no sean los actuales, se asigna el valor del atributo "*isBlocked*" al atributo "*isDisabled*" del componente encargado de mostrar tanto las preguntas como los pasos.

En la Figura 87 se muestra un ejemplo de cómo se utiliza el atributo "*isBlocked*" en el contexto de las preguntas. En este caso, si "*ques.isBlocked*" es "*true*" el atributo "*isDisable*" desactiva el componente "*AccordionItem*" evitando que el estudiante pueda interactuar con él. En caso contrario, si "*ques.isBlocked*" es "*false*" el componente se habilita permitiendo desplegar la pregunta.



```
TutorIntegrado - TutorWordProblem.tsx

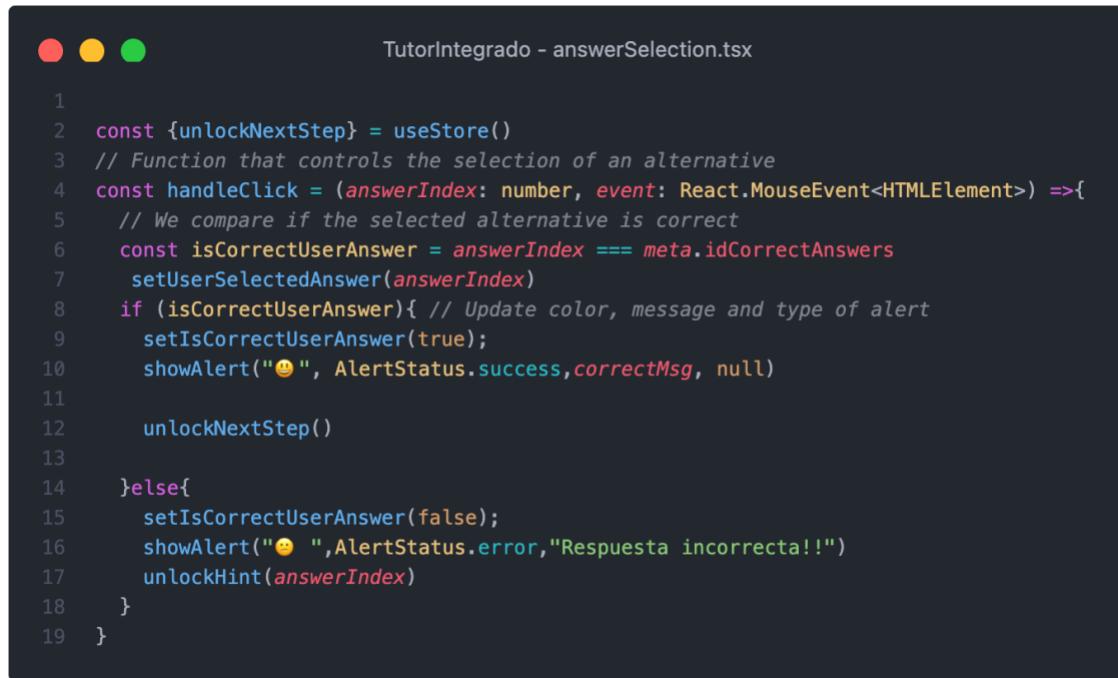
1 <AccordionItem key={quesIndex} isDisabled={ques.isBlocked} marginBottom={2}>
2   <h2>
3     <AccordionButton color={isCurrentQuestion ? "white": textColor} _hover={}>
4       <Box as="span" flex="1" textAlign="left">
5         <Latex>
6           {(ques.questionId+1)+"." + ques.question}
7         </Latex>
8       </Box>
9       <AccordionIcon/>
10
11     </AccordionButton>
12   </h2>
13   <AccordionPanel bgColor={bgQuestion} rounded={rounded}>
14     {ques.steps.map((step,index) => {
15       // RESTO DEL CODIGO
16     })}
17   </AccordionPanel>
18 </AccordionItem>
```

Figura 87: Ejemplo de uso del atributo "*isBlocked*".

Finalmente, para permitir el desbloqueo progresivo de los pasos y preguntas a medida que el estudiante avanza en el ejercicio, es necesario incorporar este *hook* en cada componente encargado de verificar la corrección de cada paso. Al confirmar que el estudiante ha

completado con éxito un paso en particular, debe invocarse la función "*unlockNextStep*" proporcionada por el *hook "useStore"*. Esto desencadenará la apertura del siguiente paso y, si se trata del último paso de una pregunta, también se desbloqueará la siguiente pregunta.

En la Figura 88 se muestra un fragmento de código del componente "*SelectionComponent*", en el cual se emplea el *hook "useStore"* para acceder exclusivamente a la función "*unlockNextStep*". Esta función se invoca únicamente cuando se verifica que la respuesta es correcta.



```
1
2 const {unlockNextStep} = useStore()
3 // Function that controls the selection of an alternative
4 const handleClick = (answerIndex: number, event: React.MouseEvent<HTMLElement>) =>{
5     // We compare if the selected alternative is correct
6     const isCorrectUserAnswer = answerIndex === meta.idCorrectAnswers
7     setUserSelectedAnswer(answerIndex)
8     if (isCorrectUserAnswer){ // Update color, message and type of alert
9         setIsCorrectUserAnswer(true);
10        showAlert("😊", AlertStatus.success,correctMsg, null)
11
12        unlockNextStep()
13
14    }else{
15        setIsCorrectUserAnswer(false);
16        showAlert("😢", AlertStatus.error,"Respuesta incorrecta!!!")
17        unlockHint(answerIndex)
18    }
19 }
```

Figura 88: Ejemplo de uso de la función "*unlockNextStep*".

Similarmente, se ha implementado el uso de la función "*unlockNextStep*" en todos los componentes desarrollados en este proyecto.

5.5.3 Revisión de sprint

Durante la revisión de este sprint, se realizó una evaluación de la implementación de la secuencialidad en la presentación de preguntas y pasos dentro del ejercicio. Se verificó que se cumpliera la condición de que los pasos y preguntas estuvieran bloqueados inicialmente, excepto por la primera pregunta y su primer paso, tal como se había definido en los requisitos.

Aunque durante esta revisión no se identificaron errores ni problemas en el funcionamiento de esta funcionalidad, se hizo hincapié en la importancia de proporcionar una claridad visual para identificar tanto la pregunta como el paso actual en el que el

estudiante está trabajando. Además, se planteó la solicitud de realizar visualmente la agrupación de los pasos correspondientes a cada pregunta, con el propósito de ofrecer una comprensión visual más sólida de la relación entre los pasos y las preguntas.

La retroalimentación recibida durante la revisión fue tomada en cuenta y se aplicaron los ajustes necesarios para mejorar la claridad visual y la organización de los pasos y preguntas en el ejercicio.

5.6 Sprint 6

El último *sprint* se centra en completar la funcionalidad relacionada con el almacenamiento de información sobre las acciones realizadas por los estudiantes en la plataforma, haciendo uso del mecanismo de reporte de acciones provisto por la plataforma.

5.6.1 Sprint backlog

- El tutor debe ser capaz de reportar las acciones realizadas por el estudiante y enviarlas al sistema central.

5.6.2 Product increment

Para cumplir con esta tarea, se reutilizaron las acciones que ya se habían utilizado en la plataforma hasta el momento. Estas acciones se describen detalladamente en la Tabla 25.

Tabla 25: Verbos para el reporte de acciones.

VERB	Parámetros	Descripción
<i>loadContent</i>	contentID	Reporte de la carga de un ejercicio.
<i>openStep</i>	contentID, stepId	Reporte al abrir un paso.
<i>closeStep</i>	contentID, stepId	Reporte al cerrar un paso.
<i>requestHint</i>	contentID, stepID, hintID, extra = {hint: texto del hint}	Reporte al solicitar una ayuda.
<i>tryStep</i>	contentID, stepID, result=[0,1], kcs=[], extra={}	Reporte al enviar una respuesta. “ <i>result</i> ” toma el valor 0 si la respuesta es incorrecta y 1 si es la respuesta es correcta.
<i>completeContent</i>	contentID, result = 1	Reporte al completar un ejercicio

Es importante destacar que en cada acción que requiere el parámetro “*stepId*”, se proporciona tanto el identificador del paso como el identificador de la pregunta. De esta

manera, "stepID" corresponde a un arreglo donde el primer elemento es el identificador de la pregunta y el segundo elemento es el identificador del paso.

El sistema de tutor cognitivo cuenta con un hook implementado llamado "useAction". Este hook permite llevar a cabo el reporte de acciones. Para usarlo, es necesario importarlo en el componente donde se desee utilizar. La Figura 89 ilustra un ejemplo de cómo se realiza la importación y definición de este hook en el componente principal de este proyecto.



```
● ● ● TutorIntegrado - TutorWordProblem.tsx
1 import { useAction } from "../../utils/action";
2 export const TutorWordProblem = ({
3   exercise,
4   topicId,
5 }: {
6   exercise: wpExercise;
7   topicId: string;
8 }) => {
9   const reportAction = useAction();
10 }
```

Figura 89: Importación y definición del hook *useAction*.

Una vez importado, el reporte de acciones puede ser ejecutado proporcionándole los atributos requeridos. En la Figura 90, se presenta un ejemplo de cómo se lleva a cabo el reporte de la acción "loadContent", que corresponde a la acción cuando se carga un nuevo ejercicio. Este reporte de acción se realiza dentro del hook *useEffect*, lo que significa que la acción se reportará cada vez que cambie el ejercicio. En esta acción en particular, se proporcionan como parámetros el nombre del verbo (*verbName*), el identificador del contenido (*contentID*), y el identificador del tópico (*topicID*).



```

1  useEffect(() => {
2    setCurrentQues(0); // Reset currentQuestionIndex to 0 for the new exercise
3    setCurrentStep(0); // Reset currentStepIndex to 0 for the new exercise
4    resetExpandedIndices();
5    resetExpandedStepIndices();
6    const initialQuestions = exercise.questions.map((ques, quesIndex) => ({
7      ...ques,
8      isBlocked: quesIndex !== 0,
9      steps: ques.steps.map((step, stepIndex) => ({
10        ...step,
11        isBlocked: quesIndex === 0 && stepIndex === 0,
12      })),
13    }));
14    setQuestions(initialQuestions);
15    reportAction({
16      verbName: "loadContent",
17      contentID: exercise.code,
18      topicID: topicId,
19    });
20  }, [exercise]);

```

Figura 90: Ejemplo de reporte de acción.

De manera similar, se realiza el reporte de las otras acciones en los componentes correspondientes, utilizando los parámetros adecuados.

La Figura 91 presenta el reporte, en el sistema, de las acciones definidas en la Tabla 25.

ID	VERB	TIMESTAMP	RESULT	USER	CONTENT	TOPIC	KCS	STEP ID	HINT ID
135532	completeContent	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	-	-	-
135531	tryStep	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	DesVarEcu Despejar variable en una ecuación	[3,3]	-
135530	tryStep	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	DivFrac Expresar división de fracciones como multiplicación	[3,2]	-
135529	tryStep	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	SRFrac Sumar y restar fracciones	[3,1]	-
135528	tryStep	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	ExpA Expresar algebraicamente	[3,0]	-
135527	requestHint	07/09/2023, 00:59 GMT-4	-	[30] Enzo marin	[wp2 word problem 2]	-	-	[3,0]	0
135526	tryStep	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	DivFrac Expresar división de fracciones como multiplicación	[2,2]	-
135525	tryStep	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	SRFrac Sumar y restar fracciones	[2,1]	-
135524	tryStep	07/09/2023, 00:59 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	SusFor Sustituir en fórmula	[2,0]	-
135523	tryStep	07/09/2023, 00:58 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	-	[1,0]	-
135522	tryStep	07/09/2023, 00:58 GMT-4	0	[30] Enzo marin	[wp2 word problem 2]	-	-	[1,0]	-
135521	tryStep	07/09/2023, 00:58 GMT-4	1	[30] Enzo marin	[wp2 word problem 2]	-	-	[0,0]	-
135520	openStep	07/09/2023, 00:58 GMT-4	-	[30] Enzo marin	[wp2 word problem 2]	-	-	[0,0]	-
135519	loadContent	07/09/2023, 00:58 GMT-4	-	[30] Enzo marin	[wp2 word problem 2]	-	-	-	-

Figura 91: Reporte de acciones en el sistema.

5.6.3 Revisión de sprint

La revisión del sprint implicó llevar a cabo pruebas utilizando una cuenta de prueba con el objetivo de verificar la adecuada funcionalidad del reporte de las acciones definidas en la Tabla 25. Durante esta fase, se aseguró que cada acción se reporte de manera precisa y que los parámetros requeridos se visualicen de manera correcta en sistema central.

6. VALIDACIÓN

Este capítulo aborda el quinto objetivo específico, el cual consiste en realizar una evaluación de la implementación propuesta con estudiantes de primer año de la facultad de ciencias de ingeniería.

6.1 Planificación

Para llevar a cabo la planificación de la validación, se organizó una reunión conjunta con los profesores patrocinante y co-patrocinante. Durante esta reunión se abordaron diversos aspectos, incluyendo la cantidad de estudiantes a involucrar, donde se estableció que 30 estudiantes serían suficientes para obtener resultados significativos. También se decidió dividir a los estudiantes en dos grupos según su nivel de conocimiento: uno de *menor conocimiento* y otro de *mayor conocimiento*. Para lo anterior, el profesor co-patrocinante se encargó de inscribir a estudiantes de diferentes carreras y específicamente de los cursos de Álgebra (*menor conocimiento*) y Cálculo 1 (*mayor conocimiento*).

Durante la reunión también se establecieron los detalles relativos a la distribución homogénea de ejercicios por cada grupo de estudiantes, así como la planificación de las sesiones y la definición de las encuestas y cuestionarios a aplicar.

6.1.1 Sesiones de evaluación

La evaluación se llevará a cabo en dos sesiones separadas, esto con el fin de obtener la mayor participación posible de estudiantes. La Tabla 26 presenta el diseño detallado de la planificación para las sesiones de evaluación.

Tabla 26: Planificación de sesiones de evaluación.

Actividad	Descripción	Duración
Introducción	<ul style="list-style-type: none">Bienvenida, presentación de estudiante tesista y profesores.	5 min
Explicación de la sesión	<ul style="list-style-type: none">Explicación de uso del sistemaExplicación de la sesión y pasos a seguir para responder cada encuesta.	10 min
Resolución de ejercicios	<ul style="list-style-type: none">Resolver primer ejercicio (15 min)<ul style="list-style-type: none">Responder encuesta satisfacción (5 min)Resolver segundo ejercicio (15 min)<ul style="list-style-type: none">Responder encuesta satisfacción (5 min)Resolver tercer ejercicio (15 min)<ul style="list-style-type: none">Responder encuesta satisfacción (5 min)	60 min
SUS y encuesta final	<ul style="list-style-type: none">Resolver encuesta SUS y encuesta de satisfacción final	10 min

6.1.2 Técnica de evaluación

En el marco de esta evaluación, a cada estudiante se le asignó la resolución de tres ejercicios específicos, los cuales fueron asignados mediante la técnica denominada "*latin square design*". El propósito fundamental del uso de dicha técnica es garantizar que cada ejercicio se aborde como primera, segunda y tercera opción en diferentes instancias, con el fin de minimizar posibles influencias de factores externos que puedan afectar los resultados. Estos factores incluyen el proceso de aprendizaje (ya que la experiencia podría mejorar al resolver el segundo ejercicio debido a la familiaridad adquirida al completar el primer ejercicio en el tutor) y la fatiga acumulada (la experiencia en el tercer ejercicio podría verse afectada negativamente debido al cansancio acumulado por resolver los ejercicios previos).

La Tabla 27 detalla la asignación de los ejercicios a cada grupo de estudiantes. Esto se hace considerando que existen cinco ejercicios sujetos a evaluación y que, en total, se estima la participación de treinta estudiantes en esta evaluación. A modo de ejemplo, el grupo "A" deberá resolver los ejercicios 1, 2 y 3.

Tabla 27: Asignación de ejercicios mediante la técnica *latin square design*.

Nº estudiantes	Grupo	Primer ejercicio	Segundo ejercicio	Tercer ejercicio
6	A	1	2	3
6	B	2	3	4
6	C	3	4	5
6	D	4	5	1
6	E	5	1	2

6.1.3 Encuestas de validación

La evaluación comprende preguntas diseñadas para evaluar tanto la usabilidad del sistema como la satisfacción de los estudiantes con los ejercicios propuestos. Para medir la satisfacción, se aplicará una encuesta después de resolver cada ejercicio, utilizando la escala Likert del 1 a 5. La encuesta incluye cuatro preguntas, detalladas en el ANEXO B.

En cuanto a la evaluación de la usabilidad, se empleará la Escala de Usabilidad del Sistema (SUS, por sus siglas en inglés). SUS es un cuestionario estandarizado que permite medir de forma universal la percepción de la usabilidad de un sistema (Pedrosa, 2022). El cuestionario SUS consta de diez preguntas específicas, las cuales se detallan en el ANEXO C.

Finalmente, para obtener una visión global de la satisfacción, tanto con el sistema como con los ejercicios en su conjunto, se utilizarán dos preguntas abiertas y una pregunta utilizando la escala Likert. Los detalles de estas preguntas se encuentran en el ANEXO D.

6.2 Preparación del sistema

Previo a la realización de las sesiones de validación, el estudiante tesista se encargó de crear cuentas de prueba y llevar a cabo la asignación y creación en el sistema de los cinco grupos previamente definidos mediante la técnica "*latin square design*". Además, se preparó una versión específica del sistema destinada exclusivamente para las sesiones de evaluación. En esta versión se realizaron ajustes y modificaciones, entre los cuales se incluyó la ocultación de los tópicos que no están relacionados con este proyecto en las cuentas de prueba. También se implementó la creación de una página de inicio (ver Figura 92) y una breve página con instrucciones generales para orientar a los estudiantes durante las sesiones (ver Figura 93).

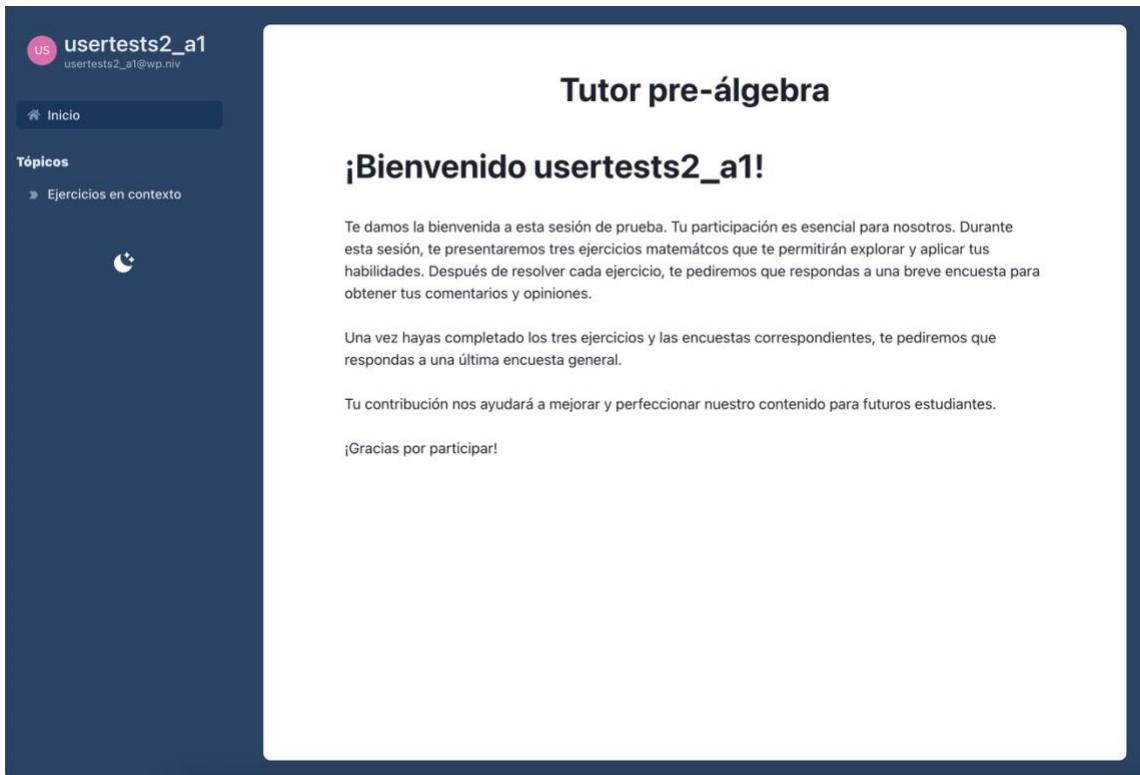


Figura 92: Página de inicio para las sesiones de evaluación.

The screenshot shows a user interface for an evaluation session. On the left, there's a sidebar with a user icon, the text 'usertests2_a1', and an email address 'usertests2_a1@wp.niv'. Below this are links for 'Inicio' and 'Tópicos' (with 'Ejercicios en contexto' under it). There's also a small circular icon with a play symbol. The main content area has a title 'Perteneces al Grupo A' and a section 'Instrucciones generales:' containing a numbered list of steps. Step 1 is about starting exercises, step 2 is about completing the questionnaire, step 3 is about advancing to the second exercise, step 4 is about completing the questionnaire again, step 5 is about advancing to the third exercise, step 6 is about completing the questionnaire once more, and step 7 is about responding to a general questionnaire covering pages 5 to 7. Below the steps is an observation note about returning to the page to continue exercises if needed. At the bottom, there's a message about being ready to start, a reference 'wp1"', and a prominent blue button labeled 'Comenzar!'. The overall background is dark blue.

Figura 93: Pagina de instrucciones para comenzar sesiones de evaluación.

Adicionalmente, se implementó un algoritmo para la asignación de los ejercicios en función del grupo al que pertenezca cada cuenta de prueba. Este algoritmo garantiza que se presenten a cada usuario los ejercicios correspondientes de acuerdo con la asignación definida mediante la técnica "*latin square design*".

La Figura 94 ilustra la interfaz de presentación del primer ejercicio para un usuario del grupo A. En este caso, el ejercicio inicial es el "1". La figura también muestra el botón "Siguiente", el cual permite a los usuarios avanzar a los siguientes ejercicios una vez que hayan concluido con el ejercicio actual. En el contexto particular de un usuario del grupo A, los ejercicios subsiguientes serían los ejercicios "2" y "3".

The screenshot shows a digital interface for an exercise. At the top, it says "Ejercicio 1". Below that, there is a table with the following data:

SECTOR	CO ₂ NETO
Energía	$8,18231 \cdot 10^{13}$ gr
Procesos Industriales	$4,3602 \cdot 10^{12}$ gr
Usos de la tierra (Agricultura, Silvicultura y otros)	$-3.9915 \cdot 10^{13}$ gr
Residuos	$3 \cdot 10^8$ gr

Below the table, it says "Tabla: Niveles de CO₂ en gramos por sector en Chile durante el año 2013".

Text below the table states: "Los números positivos representan emisiones de GEI, mientras que los negativos representan absorciones de GEI."

A note says: "A partir de la tabla anterior nos gustaría identificar:"

Question 1: "1. ¿Qué sector emite más CO₂, contribuyendo mayormente a la contaminación?"

Text in a box: "Recordatorio: La notación científica describe números basados en potencia de 10. Su forma es: $m \cdot 10^n$ "

Text: "Donde m es un número decimal cuya parte entera tiene una sola cifra distinta de cero y 10^n es una potencia de exponente entero"

Text: "Selección la alternativa correcta"

Other questions listed are: 2. ¿Qué sector absorbe CO₂, disminuyendo la contaminación?; 3. Convierte los valores de CO₂ de los sectores en potencias de 10⁹; 4. ¿Cuál es el total de emisiones?; 5. ¿Cuál es el balance entre las emisiones y absorción de CO₂?; 6. ¿Cuál es el porcentaje que aporta el sector de energía en las emisiones totales?

At the bottom right is a blue button labeled "Siguiente".

Figura 94: Presentación de ejercicio para la sesión de evaluación.

6.3 Resultados

Siguiendo la planificación diseñada para las sesiones de evaluación, se llevaron a cabo las dos sesiones en las cuales participaron 9 estudiantes en la primera sesión y 19 estudiantes en la segunda sesión, de los cuales 11 estudiantes corresponden al curso de Álgebra, mientras que 17 corresponden al curso de Cálculo. Los resultados de estas sesiones se describen a continuación.

6.3.1 Resultados Test por ejercicio

El objetivo de este test es evaluar la satisfacción de los estudiantes con respecto a cada ejercicio resuelto y comparar estos resultados entre los grupos de Álgebra y Cálculo. Esto nos permitirá analizar la percepción de los ejercicios en dos grupos que representan diferentes niveles de conocimiento, específicamente, un grupo de menor conocimiento (Álgebra) y otro de mayor conocimiento (Cálculo).

Los resultados detallados de esta encuesta se encuentran en el ANEXO E. Para el análisis, se aplicó estadística descriptiva, calculando el promedio obtenido en cada ejercicio para cada grupo de Álgebra y Cálculo

La Tabla 28 presenta los resultados promedio de la pregunta 1 (Creo que este ejercicio es útil para aprender matemáticas), en donde se compara las respuestas de los grupos de Álgebra y Cálculo por separado.

Tabla 28: Promedio de la pregunta 1 por grupo y ejercicio.

Ejercicio	Álgebra	Cálculo
1	4,3	4,6
2	4,8	4,4
3	4,0	4,3
4	4,9	4,6
5	4,6	4,4

En términos generales, tanto los estudiantes de Álgebra como los de Cálculo consideran que estos ejercicios son útiles para el aprendizaje de las matemáticas. Además, no se observan grandes diferencias entre los dos grupos (ver Figura 95), lo que sugiere que ambos grupos encuentran igual valor en estos ejercicios en términos de utilidad para el aprendizaje.

1. Creo que este ejercicio es útil para aprender matemáticas.

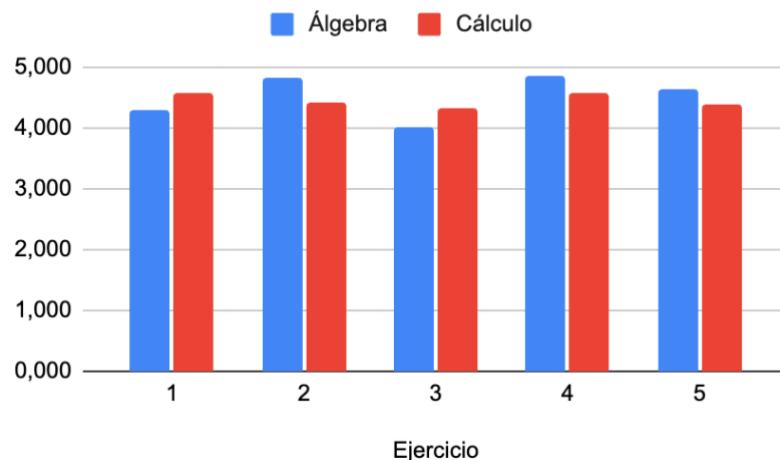


Figura 95: Resultados promedio de la pregunta 1 por ejercicio.

La Tabla 29 presenta los resultados promedio de la pregunta 2 (con este ejercicio aprendí algo nuevo), en donde se compara las respuestas de los grupos de Álgebra y Cálculo por ejercicio.

Tabla 29: Promedio de la pregunta 2 por grupo y ejercicio.

Ejercicio	Álgebra	Cálculo
1	4,3	3,6
2	4,5	4,7
3	3,6	3,1
4	3,3	3,3
5	4,0	4,1

Los resultados indican que, en promedio, tanto los estudiantes de Álgebra como los de Cálculo sienten que aprendieron algo nuevo a través de estos ejercicios. Sin embargo, es interesante notar que en los ejercicios 1 y 3 (ver Figura 96), los estudiantes de Álgebra obtuvieron puntajes más altos que los estudiantes de Cálculo. Esto sugiere que, en estos ejercicios específicos, los estudiantes de Álgebra percibieron un mayor valor en términos de aprender algo nuevo en comparación con los estudiantes de Cálculo.

2. Con este ejercicio aprendí algo nuevo.

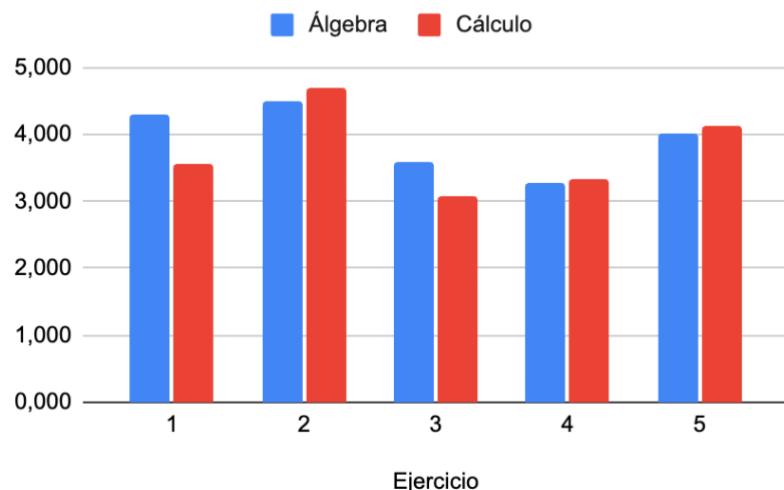


Figura 96: Resultados promedio de la pregunta 2 por ejercicio.

La Tabla 30 presenta los resultados promedio de la pregunta 3 (Creo que este ejercicio, con enunciado y aplicación real, es mejor para aprender que un ejercicio explícito), en donde se compara las respuestas de los grupos de Álgebra y Cálculo por ejercicio.

Tabla 30: Promedio de la pregunta 3 por grupo y ejercicio.

Ejercicio	Álgebra	Cálculo
1	4,3	4,4
2	4,3	4,5
3	4,0	4,5
4	4,6	4,6
5	4,1	4,1

En promedio, ambos grupos consideran que los ejercicios con enunciados y aplicaciones reales son mejores para aprender que los ejercicios puramente matemáticos. Las puntuaciones son similares en ambos grupos y no se observan grandes diferencias (ver Figura 97).

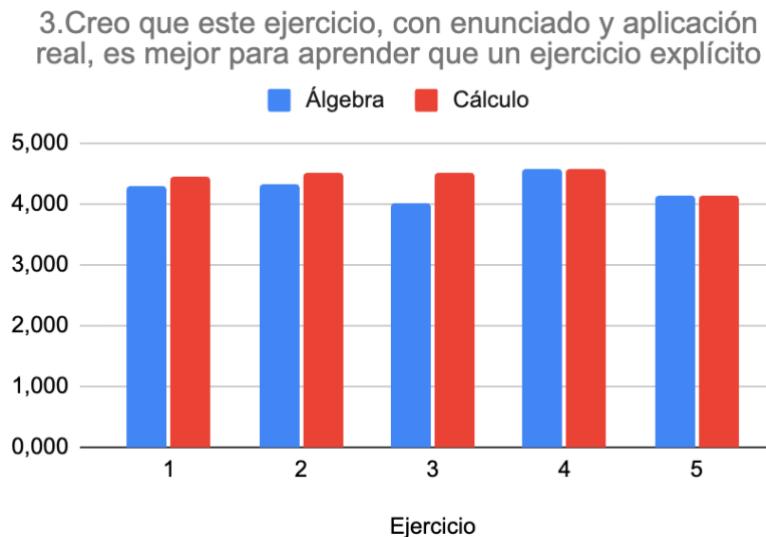


Figura 97: Resultados promedio de la pregunta 3 por ejercicio.

La Tabla 31 presenta los resultados promedio de la pregunta 4 (Resolver este ejercicio me ayuda a reflexionar sobre la importancia de aprender matemáticas), en donde se compara las respuestas de los grupos de Álgebra y Cálculo por ejercicio.

Tabla 31: Promedio de la pregunta 4 por grupo y ejercicio.

Ejercicio	Álgebra	Cálculo
1	4,4	3,6
2	4,3	4,5
3	4,2	3,8
4	4,4	3,9
5	4,6	4,3

En cuanto a la pregunta cuatro, se observa una diferencia en la que el grupo de Álgebra obtiene puntuaciones ligeramente más altas en esta pregunta en todos los ejercicios, excepto en el segundo (ver Figura 98). Esto podría sugerir que los estudiantes de Álgebra sienten que resolver estos ejercicios les ayuda más a reflexionar sobre la importancia de aprender matemáticas en comparación con los estudiantes de Cálculo.

4. Resolver este ejercicio me ayuda a reflexionar sobre la importancia de aprender matemáticas

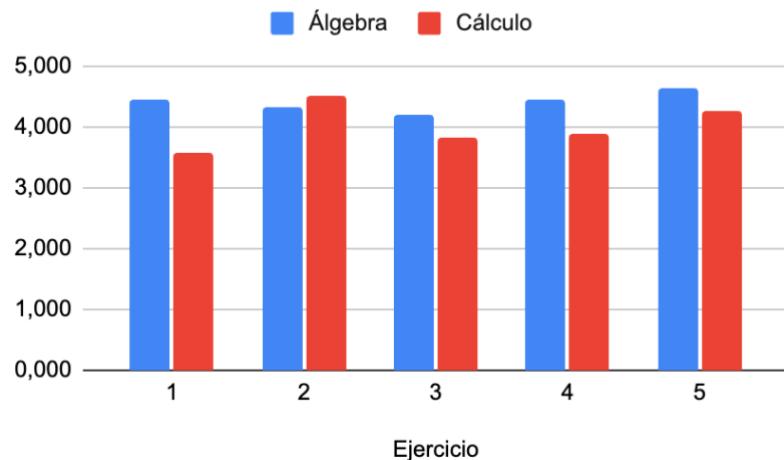


Figura 98: Resultados promedio de la pregunta 4 por ejercicio.

Estos resultados indican, en términos generales, que tanto los estudiantes de Álgebra como los de Cálculo tienen una percepción positiva de los ejercicios en relación con su utilidad, su capacidad para aprender algo nuevo, su preferencia por ejercicios con enunciados y aplicaciones reales, y su capacidad para reflexionar sobre la importancia de aprender matemáticas.

Además, es importante destacar que no se observan grandes diferencias entre las respuestas de los dos grupos, lo que sugiere que tanto los estudiantes con menor conocimiento matemático (Álgebra) como los de mayor conocimiento (Cálculo) valoran de manera similar estos tipos de ejercicios contextualizados. Por lo tanto, la percepción de los ejercicios parece no variar en función del nivel de conocimiento matemático de los estudiantes. Estos hallazgos respaldan la efectividad de los ejercicios contextualizados como herramientas de aprendizaje en ambos grupos estudiados.

6.3.2 Resultados Test usabilidad (SUS)

En cuanto al test de usabilidad, los resultados obtenidos en ambas sesiones se encuentran detallados en el ANEXO E. A estos resultados se les calculó su contribución (cuya tabla es presentada en el ANEXO E) y se obtuvo un puntaje SUS promedio de 85.74 sobre un total de 100.

Con base en este resultado, se puede concluir que la plataforma de aprendizaje propuesta ha obtenido una calificación de usabilidad sólida y positiva por parte de los estudiantes participantes en las sesiones de evaluación. Un puntaje promedio de 85.74 en la escala SUS sugiere que los usuarios encuentran que la plataforma es fácil de usar y eficiente en general.

Estos resultados son alentadores y respaldan la eficacia de la implementación propuesta en términos de su usabilidad. Además, demuestran que la inclusión de ejercicios en contexto no afecta negativamente la usabilidad de la plataforma.

6.3.3 Resultados Test Satisfacción

Para llevar a cabo el análisis del test de satisfacción, que incluye dos preguntas abiertas y una pregunta en escala Likert, se procedió inicialmente a codificar las respuestas de las preguntas abiertas. Esto se hizo con el propósito de agrupar las respuestas (frases y/o conceptos que se repiten) en categorías que pudieran ser cuantificadas, lo que facilitaría un análisis más preciso de los datos recopilados. Los resultados detallados de este análisis se encuentran disponibles en el ANEXO E.

La Figura 99 y la Figura 100 presenta las categorías resultantes, para los aspectos positivos y negativos respectivamente, mediante el uso de nubes de etiquetas.



Figura 99: Nube de etiquetas de aspectos positivos.

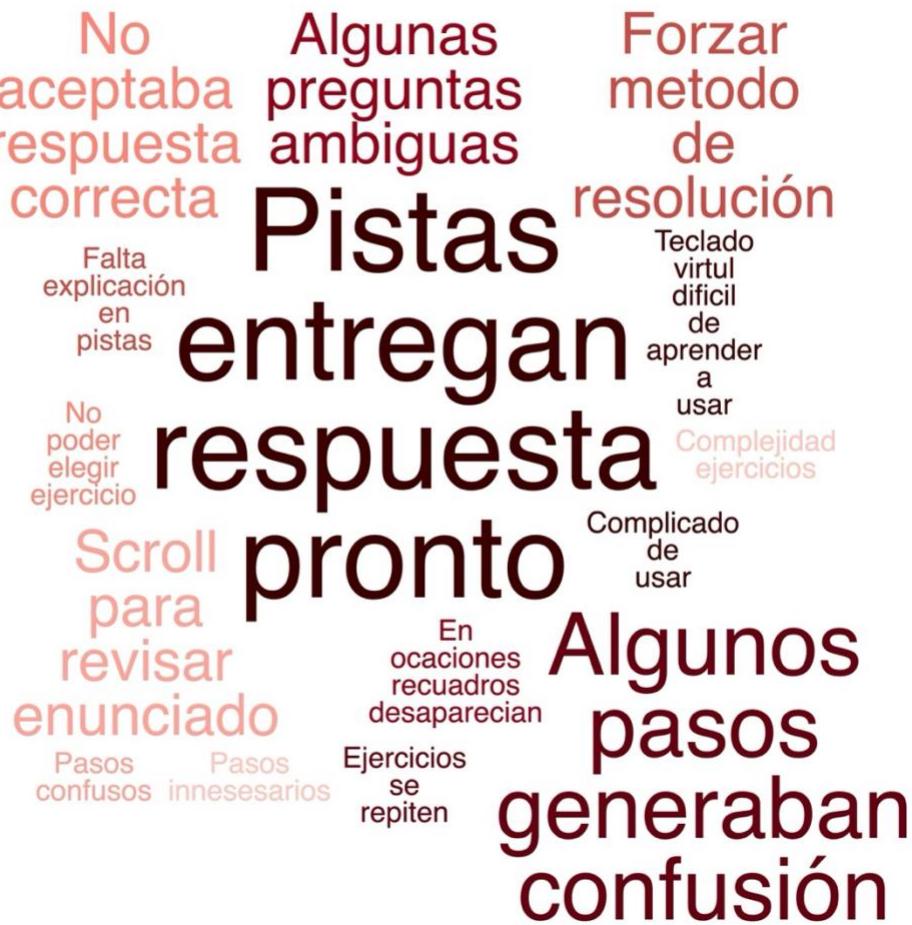


Figura 100: Nube de etiquetas de aspectos negativos.

Es importante destacar de manera significativa la presencia de una gran cantidad de aspectos positivos en comparación con los aspectos negativos (59 contra 25). Esto indica que la mayoría de los estudiantes expresaron opiniones favorables con respecto a los ejercicios.

Esta predominancia de aspectos positivos en las respuestas es un indicador alentador y sugiere que los estudiantes apreciaron en gran medida los ejercicios. Sin embargo, también es importante considerar los aspectos negativos mencionados por algunos estudiantes para identificar áreas de mejora y perfeccionamiento, tanto en la plataforma como en los ejercicios.

A continuación, se describen las categorías claves identificadas.

Principales Aspectos Positivos:

- **Facilita la aplicación de las matemáticas:** Los estudiantes valoraron positivamente que los ejercicios estuvieran contextualizados, lo que les ayudó a comprender cómo aplicar conceptos matemáticos en situaciones del mundo real. Este aspecto fue mencionado ocho veces y se percibe como una fortaleza de la plataforma.
- **Pistas:** Algunos estudiantes también destacaron la utilidad de las pistas, especialmente cuando cometían errores comunes. Esta categoría igualmente fue mencionada 8 veces.
- **Bien explicados:** Gran parte de los estudiantes apreciaron la claridad y la calidad de las explicaciones proporcionadas en los ejercicios. Esto les ayudó a comprender mejor los conceptos matemáticos. Esta categoría fue mencionada 6 veces como aspecto positivo.
- **Dinámica paso a paso:** Los estudiantes valoraron la presentación de los ejercicios en un formato de resolución paso a paso. La cual fue mencionada 6 veces como aspecto positivo.
- **Potencia debilidades:** Algunos estudiantes mencionaron que los ejercicios les ayudaron a identificar sus debilidades en matemáticas y cómo mejorarlas y/o reforzarlas. Esta categoría fue mencionada 6 veces como un aspecto positivo.
- **Ejercicios cotidianos:** Algunos estudiantes expresaron que apreciaban la relación de los ejercicios con situaciones cotidianas. Mencionando que les permitió ver la utilidad de las matemáticas en su vida diaria. Esta categoría fue mencionada 5 veces.

Principales aspectos Negativos:

- **Pistas entregan respuesta pronto:** Algunos estudiantes señalaron que las pistas proporcionadas revelaban la respuesta final demasiado temprano en el proceso de resolución, lo que podría reducir el desafío y la oportunidad de aprendizaje. Esta categoría fue mencionada 4 veces.
- **Algunos pasos generaban confusión:** La ambigüedad en la explicación de algunos pasos generó confusión entre algunos estudiantes, lo que indica la necesidad de mejorar aún más la claridad en ciertas explicaciones de los ejercicios. Esta categoría fue mencionada 3 veces como aspecto negativo.
- **No aceptaba respuesta correcta:** Hubo comentarios sobre el sistema que no aceptaba respuestas correctas. Sin embargo, es importante señalar que estos casos

específicos se debieron a problemas en el ingreso de elementos matemáticos desde el teclado físico, como el símbolo de elevado, lo que resultaba en respuestas finales mal formateadas, pero visualmente correctas. Este aspecto fue mencionado 2 veces.

- **Algunas preguntas ambiguas:** Se hizo referencia a la ambigüedad en ciertas preguntas como un aspecto negativo en dos ocasiones. Aunque no se especificaron las preguntas particulares, se destaca la importancia de abordar esta ambigüedad, ya que podría llevar a confusiones y respuestas incorrectas por parte de los estudiantes.
- **Scroll para revisar enunciado:** La necesidad de hacer *scroll* para volver a consultar el enunciado en ciertas situaciones se mencionó dos veces como un aspecto negativo. Esta observación sugiere que podría resultar incómodo y afectar la fluidez de la experiencia de aprendizaje de los estudiantes.

Además de los aspectos negativos mencionados anteriormente, es relevante destacar que algunos comentarios deben ser descartados. Por ejemplo, un estudiante mencionó la incapacidad de elegir entre los ejercicios a resolver como un aspecto negativo. Sin embargo, es importante tener en cuenta que este estudiante ya había participado en una evaluación previa destinada a evaluar otros aspectos del proyecto Fondecyt, que están fuera del alcance de este proyecto y encuesta. Por lo tanto, la capacidad de selección de contenido no era un aspecto que se buscara evaluar en este contexto específico.

La retroalimentación proporcionada por los estudiantes constituye una fuente valiosa de información para perfeccionar la plataforma y hacerla aún más efectiva y educativa. El análisis detallado de los aspectos positivos y negativos contribuirá a guiar futuras actualizaciones y mejoras en los ejercicios, la claridad de las pistas, la estructura de los pasos de resolución y otros aspectos críticos de la experiencia de aprendizaje.

Finalmente, con respecto a la pregunta en escala Likert (La cual consulta si recomendaría estos ejercicios a sus compañeros), esta se analizó utilizando estadísticas descriptivas para calcular el promedio de las respuestas de los estudiantes. Los detalles completos de los resultados de esta pregunta se encuentran en el ANEXO E.

Los resultados de este análisis se proporcionan en la Tabla 32.

Tabla 32: Promedio por grupo de la pregunta 3 del test de satisfacción.

Grupo	Promedio
Algebra	4,6
Calculo	4,4

Estos resultados sugieren que la mayoría de los estudiantes están dispuestos a recomendar estos ejercicios a sus compañeros, lo que refuerza la percepción generalmente positiva obtenida a través de la encuesta. Además, se destaca que no existe grandes diferencias entre los grupos de estudiantes de álgebra y cálculo, lo que sugiere que ambos grupos tienen una opinión, homogénea, favorable sobre la plataforma en términos de recomendación. Esto indica que la efectividad y utilidad de los ejercicios se perciben de manera consistente en ambos grupos, independientemente del nivel educativo.

Estos hallazgos son alentadores y respaldan la idea de que los ejercicios en contexto tienen un potencial significativo para mejorar la experiencia de aprendizaje en matemáticas. No obstante, es esencial seguir abordando los aspectos negativos mencionados para continuar mejorando la plataforma y hacerla aún más efectiva en el futuro.

7. CONCLUSIONES Y TRABAJO FUTURO

El presente trabajo de título se ha centrado en fortalecer la plataforma de Tutores Cognitivos mediante la implementación de nuevos ejercicios interactivos y contextualizados, con un enfoque especial en aspectos socio-medioambientales. Esto se ha traducido en la creación de cinco ejercicios que abarcan diversos temas matemáticos y los conectan con situaciones del mundo real.

La implementación de estos ejercicios contextualizados también ha implicado el desarrollo de cuatro componentes novedosos (*mathComponent*, *selectionComponent*, *selectPoint* y *linearFit*) que amplían las capacidades interactivas de la plataforma. Estos nuevos componentes permiten abordar una mayor variedad de tópicos matemáticos y ofrecer una experiencia más dinámica e interactiva a los estudiantes.

La metodología SCRUM, adaptada para este proyecto, ha demostrado ser una elección acertada. El enfoque iterativo ha facilitado la recepción continua de retroalimentación, lo que ha posibilitado realizar mejoras incrementales en cada *sprint*. Esto ha contribuido significativamente al perfeccionamiento tanto de los ejercicios como de los nuevos componentes.

A lo largo del desarrollo del proyecto, se enfrentaron desafíos técnicos significativos. La adaptación e implementación del componente *MathLive*, utilizado para el ingreso de respuestas matemáticas, ha demandado un profundo entendimiento de su funcionamiento interno. La integración precisa de esta herramienta en la plataforma fue un proceso minucioso que requirió mucho tiempo y esfuerzo.

Otro desafío técnico estuvo relacionado con la integración de los componentes gráficos utilizando *JSXGraph*. Aunque esta herramienta ofrece una amplia flexibilidad para crear gráficos interactivos, su implementación requería una comprensión profunda de la biblioteca asociada.

Además, la adaptación de los ejercicios para abordar los aspectos socio-medioambientales planteó desafíos específicos en términos de la investigación y la incorporación de datos reales y contextuales. Esto implicó la revisión de fuentes confiables y la selección cuidadosa de datos que respaldaran los ejercicios de manera significativa.

A pesar de estos desafíos, los resultados de la validación confirman el éxito del trabajo. Los estudiantes han demostrado una alta aceptación y satisfacción con los nuevos ejercicios implementados. Además, la encuesta SUS ha arrojado un puntaje elevado en usabilidad, lo que evidencia la facilidad de uso y la eficiencia del sistema desarrollado.

7.1 Trabajo futuro

En cuanto al trabajo futuro, indudablemente, las recomendaciones y los aspectos negativos identificados durante la etapa de validación se consideran como áreas de enfoque clave

para mejorar tanto la plataforma como los ejercicios, con el objetivo de hacerlos aún más efectivos y enriquecedores.

Además, se espera ampliar significativamente la cantidad de ejercicios contextualizados en la plataforma. Lo anterior, con el propósito de abordar una gama aún más amplia de tópicos matemáticos y situaciones del mundo real.

Finalmente, la incorporación del componente "*JSXGraph*" en el proyecto como elemento gráfico ofrece la oportunidad de desarrollar una amplia gama de componentes interactivos que abarquen diversos temas matemáticos. Esto permitiría complementar los componentes creados en el presente proyecto con nuevos elementos que incluyan, por ejemplo, ecuaciones cuadráticas y otros conceptos matemáticos relevantes para el aprendizaje de los estudiantes.

8. REFERENCIAS

- Adebayo, S. (s.f.). Chakra UI is a simple, modular and accessible component library that gives you the building blocks you need to build your React applications. Accedido el 4 de junio, 2023, desde <https://chakra-ui.com/>
- Aleven, V., McLaren, B., & Sewall, J. (2009). Scaling Up Programming by Demonstration for Intelligent Tutoring Systems Development: An Open-Access Web Site for Middle School Mathematics Learning (Vol. 2). IEEE Transactions on Learning Technologies.
- Álvarez A., P., Arriaza L., J., Cisternas G., R., Olmos T., Y., Ríos, J.C., & Tauler C., F. (s.f.). *Matemática Básica para la Inserción a la Ingeniería* (Vol. 6).
- Arias S., F. J., Jiménez B., J. A., & Ovalle C., D. A. (2009). Modelo de planificación instruccional en sistemas tutoriales inteligentes. Revista Avances en Sistemas e Informática, 6(1),155-164. Accedido el 19 de abril, 2023, desde: <https://www.redalyc.org/articulo.oa?id=133112608016>
- Catalano, M. T. (2010). College Algebra in Context: A Project Incorporating Social Issues. Numeracy Advancing Education in Quantitative Literacy, 3. Accedido el 20 de abril, 2023, desde <http://dx.doi.org/10.5038/1936-4660.3.1.7>
- CortexJs. (s.f.). Accedido el 30 de Julio, 2023, desde <https://cortexjs.io/mathlive/>
- Del Olmo Muñoz, J., González Calero, J. A., Diago, P., Arnau, D., & Arevalillo Herráez , M. (2023). Intelligent tutoring systems for word problem solving in COVID-19 days: could they have been (part of) the solution? ZDM – Mathematics Education, 55, 35-48. Accedido el 20 de abril, 2023, desde <https://doi.org/10.1007/s11858-022-01396-w>
- Drumond, C. (s.f.). Scrum: qué es, cómo funciona y como empezar. Atlassian. Accedido el 14 de octubre, 2023, desde <https://www.atlassian.com/es/agile/scrum>
- GitHub – TutorCognitivo. Accedido el 8 de octubre, 2023, desde <https://github.com/enzomarin/TutorCognitivo/tree/ejercicios-contexto/src/components/tutorWordProblems/Components>
- JSXGraph. (s.f.). Dynamic Mathematics with JavaScript. Accedido el 27 de Julio, 2023, desde <https://jsxgraph.uni-bayreuth.de/wp/index.html>
- Next.Js by Vercel. (s.f.)- The React Framework. Accedido el 4 de junio, 2023, desde <https://nextjs.org/>
- Pedrosa, M. (29 de septiembre de 2022). ¿La usabilidad puede medirse? Escala SUS y test de usuario. flat101. Accedido el 26 de Agosto, 2023, desde <https://www.flat101.es/blog/diseño-ux/la-usabilidad-puede-medirse-escala-sus-y->

test-de-
usuario/#:~:text=El%20sistema%20de%20escala%20de,con%20la%20interfaz%
20de%20estudio.

React - A JavaScript library for building user interfaces (s.f.). Accedido el 4 de junio, 2023, desde <https://reactjs.org/>

React - reusing logic with custom hooks. Accedido el 19 de julio, 2023, desde <https://es.react.dev/learn/reusing-logic-with-custom-hooks>

Rehkopf, M. (s.f.). Todo lo que necesitas saber sobre los sprints de scrum. Atlassian. Accedido el 14 de octubre, 2023, desde <https://www.atlassian.com/es/agile/scrum/sprints>

TypeScript: JavaScript with syntax for types. (s.f.). Accedido el 4 de junio, 2023, desde <https://www.typescriptlang.org/>

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4), 197–221. <https://doi.org/10.1080/00461520.2011.611369>

Walkington, C., Clinton, V., Ritter, S. N., & Nathan, M. J. (2015). How Readability and Topic Incidence Relate to Performance on Mathematics Story Problems in Computer-Based Curricula. *Journal of Educational Psychology*, 1051–1074. Accedido el 20 de abril, 2023, desde <https://doi.org/10.1037/edu0000036>

Zustand. (s.f.). Accedido el 7 de Agosto, 2023, desde <https://zustand-demo.pmnd.rs/>

9. ANEXOS

ANEXO A: Tablas de artículos seleccionados.

Código del artículo
[A02]
Nombre del artículo
How Readability and Topic Incidence Relate to Performance on MathematicsStory Problems in Computer-Based Curricula.
¿Cuál es el contexto del artículo?
<p>El artículo se centra en una investigación de cómo varios indicadores de legibilidad y el tema o contexto de los ""word problems"" están asociados con la tendencia de los estudiantes a dar respuestas correctas y solicitar pistas en un Tutor cognitivo.</p> <p>Las categorías clave de legibilidad y temas que se relacionaron con las medidas de resolución de problemas incluyen la dificultad de las palabras, la longitud del texto, el uso de pronombres, la similitud de oraciones y la familiaridad con el tema.</p>
¿Qué datos o detalles útiles nos brinda este artículo?
<p>El artículo detalla cómo el tema o contexto en un problema textual (word problem) puede traer beneficios al momento de resolver los ejercicios por parte de los estudiantes, permitiendo construir más fácilmente un modelo de situación que represente las acciones y relaciones en un problema narrativo. Además, nombra ciertas características en el texto que pueden provocar el interés situacional de los estudiantes.</p>
Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados
No aplica.
Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo
No aplica.
¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?
<p>Si, menciona que resolver estos tipos de problemas pueden ser un esfuerzo desafiante porque implica navegar por varios tipos diferentes de información. También mencionan que Natham, Kintsch y Young (1992) propusieron un modelo de resolución de problemas matemáticos narrativos en el que los alumnos coordinan tres niveles adicionales de representación:</p> <ul style="list-style-type: none">a) La base de texto o la información proposicional proporcionada en el texto del problema presentado como una red de relacionesb) El modelo de situación o una representación mental de las relaciones, acciones y eventos en el problema que se conecta con el conocimiento previo del lector para llenar los vacíos dejados por una historia dispersa.c) El modelo de problema de operandos matemáticos formales como números y variables <p>Es a través de la coordinación de estos niveles que un alumno pasa de un modelo superficial a la creación de sentido que media en la formación de una respuesta significativa.</p>

¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?
No aplica.
¿El artículo describe los tipos de feedback entregados a los estudiantes?
No aplica.
¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?
No aplica.
Otros hallazgos relevantes
No aplica.

Código del artículo
[A03]
Nombre del artículo
Analyzing and generating mathematical models: An Algebra II CognitiveTutor design study.
¿Cuál es el contexto del artículo?
Este artículo se centra en la evaluación de una herramienta de modelado matemático introducido en un Tutor Cognitivo de Álgebra 2: Esta herramienta tiene como objetivo que los estudiantes aprendan a escribir modelos generales de forma lineal, enfocando en problemas del mundo real.
¿Qué datos o detalles útiles nos brinda este artículo?
Nos brinda un ejemplo de Tutor cognitivo que implementa problemas del mundo real que puedan ser modelados por una ecuación lineal.
Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados
El tutor de Álgebra 2 está construido en torno de un modelo cognitivo del conocimiento de resolución de problemas que los estudiantes están adquiriendo. El modelo refleja la teoría ACT-R del conocimiento de habilidades al suponer que las habilidades para resolver problemas se pueden modelar como un conjunto de reglas de producción independientes. El modelo cognitivo permite al tutor rastrear la ruta de solución del estudiante a través de un espacio de resolución de problemas complejos, brindando retroalimentación sobre cada acción de resolución de problemas y consejo sobre la resolución de problemas según sea necesario.
Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo
El sistema presenta la declaración del problema en una ventana donde se describe una situación problemática y hace cinco preguntas sobre la situación. Además presenta una hoja de trabajo en blanco (tabla). Al comienzo del problema esta hoja se encuentra en blanco y el estudiante debe ir respondiendo las preguntas planteadas completando la hoja de trabajo. Además, los estudiantes (1) deben identificar cantidades o

variables relevantes en el problema y etiquetar las columnas en consecuencia; (2) ingresar las unidades apropiadas en la primera fila de la hoja de trabajo; (3) ingresar una fórmula simbólica para cada cantidad en la segunda fila; y (4) responder las preguntas en las filas sucesivas de la tabla.

¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?

No aplica.

¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?

No aplica.

¿El artículo describe los tipos de feedback entregados a los estudiantes?

No aplica.

¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?

No aplica.

Otros hallazgos relevantes

No aplica.

Código del artículo

[A04]

Nombre del artículo

Constraint-based Student Modeling in Probability Story Problems with Scaffolding Techniques.

¿Cuál es el contexto del artículo?

El contexto del artículo gira en torno a MAST, un sistema de tutoría inteligente basado en la web para problemas de probabilidades. Centrándose en el modelado basado en restricciones (CBM), el cual es una técnica de modelado del conocimiento del estudiante que se basa en la teoría de Ohlsson de aprender de los errores de desempeño. Aunque CBM se emplea típicamente con problemas de palabras en diferentes dominios, se limita a verificar la respuesta del estudiante sintáctica y semánticamente en función de restricciones definidas. Por lo general, no llega a tomar en consideración la posibilidad de que el alumno cometa un error debido a una mala interpretación del problema de la historia. Un objetivo de este trabajo es ampliar CBM para abordar los errores de interpretación errónea de los estudiantes.

¿Qué datos o detalles útiles nos brinda este artículo?

Explican cómo realizaron el modelado del estudiante basado en restricciones en problemas narrativos (Story problems). Básicamente diagnostican la falta de comprensión del estudiante sobre un tema determinado y reaccionan en consecuencia. También definen las técnicas del modelado de los estudiantes las cuales se pueden clasificar en:

1) técnicas de modelado a corto plazo: los errores en la solución del estudiante a un problema dado se diagnostican para proporcionar ayuda relevante. 2) técnica de modelado a largo plazo: Considera las interacciones de los estudiantes con el sistema a lo largo del tiempo para construir un modelo más preciso.
Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados
No aplica.
Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo
No aplica.
¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?
No aplica.
¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?
No aplica.
¿El artículo describe los tipos de feedback entregados a los estudiantes?
Presenta un tipo de apoyo (andamiaje) proporcionado durante el proceso de aprendizaje. Este apoyo está destinado a ser individualizado según los requerimientos específicos del estudiante. Además, está diseñado para estimular un nivel más profundo de aprendizaje. Dos técnicas de andamiaje: 1) Preguntas de andamiaje: secuencia de preguntas para ayudar a los estudiantes a desarrollar la comprensión del ejercicio. En estas preguntas el problema se divide en subproblemas. 2) Retroalimentación de andamiaje: se proporciona al estudiante un conjunto de pistas incrementales hasta que sea capaz de resolver el problema correctamente. Además, MAST amplía la retroalimentación del andamiaje al proporcionar sugerencias visuales en forma de palabras clave resaltadas en el problema al detectar un error del alumno.
¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?
No aplica.
Otros hallazgos relevantes
El sistema aborda errores debido a una mala interpretación del problema al ubicar palabras clave que pueden no haber sido consideradas por el estudiante. Cada restricción en el sistema está representada por la siguiente tupla: <Cr, Cs, WSD, PSD, H, [axioma]> WSD: descripción semántica de las palabras clave que debe tener en cuenta el estudiante para satisfacer la restricción PSD: Descripción del segmento del problema en que se encuentran las palabras clave. Además cada restricción va acompañada de una o más pistas H para guiar al alumno a satisfacer las restricciones en caso de error. También, algunas restricciones pueden ir acompañadas de un axioma de seguimiento.

Código del artículo
[A05]
Nombre del artículo
LIM-G: Learner-initiating instruction model based on cognitive knowledge for geometry word problem comprehension.
¿Cuál es el contexto del artículo?
Este documento se centra en un modelo de instrucción iniciada por el alumno (LIM-G) para ayudar a los alumnos a comprender los problemas textuales de geometría. De esta manera el alumno puede solicitar ayuda sobre cualquier problema geométrico que le interese y no necesariamente en los ejercicios que implementa la plataforma.
¿Qué datos o detalles útiles nos brinda este artículo?
Nos ofrece una idea de la estructura de conocimiento cognitivo que utilizar para comprender los problemas textuales de geometría y así poder generar ayuda específica a los estudiantes que la soliciten.
Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados
En este artículo, el conocimiento cognitivo significa el conocimiento necesario para comprender los problemas textuales de geometría. Para esto utilizan un árbol como estructura de conocimiento cognitivo, este contiene nodos jerárquicos de conocimiento esquemático, conceptos de problemas y conocimiento lingüístico. El conocimiento esquemático representa categorías de problemas textuales de geometría. El concepto de problema consiste en el conocimiento lingüístico necesario para comprender las descripciones lingüísticas de este concepto de problema.
Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo
Se han utilizado tres formas principales de representación para presentar problemas textuales (word problems): representación textual, esquemática y telegráficas. Siendo la representación esquemática en la que se obtuvo un desempeño significativamente mejor.
¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?
El artículo menciona que algunos investigadores encontraron que los problemas verbales son difíciles para los estudiantes porque involucra muchas habilidades, como la comprensión de lectura, la escritura de ecuaciones, la identificación de variables en el texto y el cálculo aritmético (Cummins et al., 1988; Mayer, 1992; Stem, 1993). Además algunos estudios muestran que los estudiantes no logran resolver problemas verbales porque solo pueden reconocer la información superficial de los problemas en lugar de comprender completamente las implicaciones de los problemas.
¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?
El artículo menciona que el proceso de resolución de problemas matemáticos verbales consta de dos pasos, la representación del problema y la solución del problema. Para la representación del problema, un alumno

necesita transformar la descripción de un problema en su representación mental interna en dos etapas: la traducción e integración del problema. La traducción de problemas extrae conceptos geométricos de la descripción textual del problema utilizando conocimientos lingüísticos y semánticos. El conocimiento lingüístico se utiliza para comprender los significados de las palabras en la descripción textual, mientras que el conocimiento semántico significa conocimiento fáctico en el mundo.

La integración de problemas requiere que un alumno conecte las oraciones en una descripción de problemas y produzca una representación coherente. En esta etapa, se necesita un conocimiento esquemático de la clasificación del problema para integrar las piezas de información proporcionadas por el problema.

¿El artículo describe los tipos de feedback entregados a los estudiantes?

No aplica.

¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?

No aplica.

Otros hallazgos relevantes

No aplica.

Código del artículo

[A06]

Nombre del artículo

Domain-specific knowledge representation and inference engine for an intelligent tutoring system.

¿Cuál es el contexto del artículo?

Este artículo describe un Sistema de Tutor Inteligente que se enfoca en la etapa del proceso de resolución de problemas, específicamente en la traducción de problemas verbales a notación simbólica.

¿Qué datos o detalles útiles nos brinda este artículo?

El artículo propone un motor de inferencia específico de dominio y un mecanismo de representación del conocimiento. Como resultado, el sistema puede simultáneamente:

- a) Representar todas las posibles soluciones algebraicas a un problema verbal dado.
- b) Realizar un seguimiento de las acciones del estudiante.
- c) Determinar unívocamente el estado actual del proceso de resolución.
- d) Construir un modelo de estudiante.
- e) Proporcionar remediación automática adaptativa.

Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados

Su enfoque principal es el método cartesiano para la resolución de problemas algebraicos. Este puede ser descrito por una secuencia de acciones ordenadas:

- (1) asignar n letras para representar n cantidades desconocidas;
- (2) definir el resto de las incógnitas mediante expresiones algebraicas ;

- (3) establecer un conjunto de n ecuaciones sobre la base de que n cantidades pueden expresarse de dos maneras diferentes; y
 (4) resolver el conjunto de n ecuaciones

Además, el conocimiento algebraico de un problema verbal lo representan como una función de cantidades conocidas, cantidades desconocidas y las relaciones entre ellas. Esto lo describen mediante el uso de gráficos trinomiales, específicamente un hipergrafo.

Por último, para permitir un almacenamiento eficiente, toda la información se representa en formato XML.

Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo

La GUI está implementada en dos secciones. El lado izquierdo se utiliza para definir nuevas variables, ya sea ingresando una letra o una expresión que relacione otras cantidades definidas previamente. Si se usa una letra, el alumno también debe seleccionar un significado de cantidad de una lista desplegable. Las posibles descripciones se extraen de la base de conocimiento y permiten que el componente del dominio vincule únicamente la letra a la cantidad correcta. El estudiante construye las expresiones algebraicas utilizando un componente gráfico similar a una calculadora que contiene un botón para cada operador aritmético., y otro por cada cantidad que ya se haya definido.

Las ecuaciones se construyen usando la calculadora en un panel lateral derecho. Este contiene el signo igual y los mismos botones que el componente utilizado para definir nuevas cantidades. Además para ayudar a identificar visualmente las cantidades ya definidas, estas también se muestran en una tabla que incluye tanto sus valores como las descripciones asociadas.

¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?

No aplica.

¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?

Básicamente el estudiante debe ir agregando en una tabla las distintas variables y expresiones necesarias para la resolución del problema, luego en la misma tabla debe ir asignando a cada variable o expresión un significado disponible en una lista desplegable.

¿El artículo describe los tipos de feedback entregados a los estudiantes?

El ITS proporciona tanto recomendaciones a pedido (pistas) como ayuda no solicitada cuando el estudiante comete un error.

Para proporcionar sugerencias, primero se calcula automáticamente la próxima acción factible. El artículo también proporciona el algoritmo utilizado para este propósito.

Cuando se brinda la ayuda solicitada y el estudiante aún no puede avanzar en el proceso, se sigue una estrategia de refinamiento.

El contenido del mensaje se genera dinámicamente mediante el uso de planillas parametrizadas prediseñadas. Una plantilla puede incluir texto, campos especiales y declaraciones condicionales.

Campos especiales: palabras claves que se traducen dinámicamente en mensajes verbales significativos utilizando la inf. contenida en la descripción del XML.

Sentencias condicionales: se utilizan con fines de adaptación.

¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?

Para poder realizar un seguimiento del proceso de resolución, tanto el hipergrafo como la GUI deben actualizarse después de cada entrada válida del estudiante. El proceso de resolución se considera finalizado cuando se han tratado todos los bordes del hipergrafo y, por lo tanto, se han eliminado del hipergrafo. Este

modo de operación le permite al estudiante tomar cualquiera de las muchas rutas de resolución válidas que arrojan una solución correcta.

Otros hallazgos relevantes

El artículo presenta y detalla el uso de una plantilla dinámica utilizada para responder a una solicitud de pista cuando el algoritmo para generar recomendaciones sugiere la definición de una cantidad en términos de otras cantidades existentes.

Además las sugerencias emitidas por el algoritmo también se pueden usar para calcular resoluciones alternativas para un problema textual dado. El ITS presentado en el artículo incorpora un motor de resolución de problemas que utiliza iterativamente este algoritmo y actualiza la hipergrafía como si el usuario hubiera ingresado la acción sugerida. Como resultado, se muestra una secuencia de pasos que arroja una solución correcta del problema.

Código del artículo

[A07]

Nombre del artículo

Fundamentals of the design and the operation of an intelligent tutoring system for the learning of the arithmetical and algebraic way of solving word problems

¿Cuál es el contexto del artículo?

El contexto de este artículo recae en un Sistema de Tutor Inteligente que tiene como objetivo supervisar la resolución de problemas textuales. Además, este sistema permite al usuario seguir cualquier camino de resolución. Es decir aborda el problema de las múltiples rutas posibles en la resolución de un “word problem” algebraico

¿Qué datos o detalles útiles nos brinda este artículo?

Este artículo es un excelente ejemplo donde se modelan e implementan ejercicios textuales en un Tutor inteligente.

Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados

El sistema planteado en este artículo utiliza el método cartesiano (MC) para describir la secuencia de pasos necesarios para la resolución de un problema algebraico textual.

De esta manera el MC se puede descomponer en:

- 1) La lectura analítica del enunciado del problema para transformarlo en una lista de cantidades y relaciones entre cantidades.
- 2) Elegir una cantidad (o varias cantidades) que se designa con una letra (o varias letras diferentes)
- 3) Escribir expresiones algebraicas para designar las demás cantidades, utilizando la letra (o letras) introducida en el segundo paso y las relaciones encontradas en la lectura analítica realizada en el primer paso.
- 4) Escribir una ecuación (o tantas ecuaciones independientes como el número de letras introducidas en el segundo paso), a partir de la observación de que dos expresiones algebraicas (no equivalentes) escritas en el tercer paso designan la misma cantidad.
- 5) Transformando la ecuación en una forma canónica.
- 6) La aplicación de la fórmula o el algoritmo de solución a la ecuación en forma canónica.
- 7) La interpretación del resultado en términos del planteamiento del problema

Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo

El diseño de la interfaz tiene las siguientes características: (1) proporciona una clara separación de procedimientos de los pasos de MC; (2) permite resoluciones tanto aritméticas como algebraicas; (3) proporciona soporte para supervisar cada paso de resolución (4) las expresiones matemáticas solo se pueden introducir mediante botones, se crea un botón para cada nueva cantidad que el estudiante define; (5) se proporcionan comentarios inmediatos para acciones incorrectas y también ayuda a pedido; (6) los mensajes se expresan en lenguaje natural.

Una vez seleccionado el problema, el sistema muestra el enunciado y solo le da la posibilidad al usuario de presionar el botón etiquetado como "definir nueva cantidad". Luego se activa el panel de definición de cantidad. En este panel, el usuario debe seleccionar el nombre de una cantidad entre las que se muestran en una lista desplegable, y elegir entre tres acciones posibles: Asignar número, asignar letra o asignar expresión.

¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?

El artículo destaca que la competencia en la resolución algebraica de problemas textuales depende de tres componentes principales: La competencia en el uso del lenguaje natural en que se plantea el problema; la competencia en el uso del lenguaje algebraico en que se representará la ecuación; y la competencia para traducir el texto expresado en lenguaje natural a lenguaje algebraico.

¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?

Las acciones asociadas con el proceso de traducción corresponden a los primeros cuatro pasos del MC. De esta manera, para representar la lectura analítica de un problema se utilizan los hipergrafos, en donde las relaciones entre cantidades se representan como aristas con tres vértices. Cada vértice representa una cantidad. Cada borde representa una de las cuatro operaciones típicas entre dos cantidades (esto es, +,-,*,/). Ademas las cantidades conocidas se representan mediante circulos negros (vertices oscuros) y las cantidades desconocidas se representan mediante cuadros vacíos (vertices claros). Se permiten aristas con cualquier número de vértices cuando sea necesario.

Por último para facilitar la comprensión de la representación de los hipergrafos en este artículo usan líneas verticales para los bordes que representan relaciones aditivas y líneas horizontales para los bordes que representan relaciones multiplicativas.

¿El artículo describe los tipos de feedback entregados a los estudiantes?

Las descripciones de variables en el archivo XML se utilizan para generar mensajes semánticos significativos utilizando un lenguaje vernáculo. Otra información, como reglas de ayuda predefinidas o listas que incluyen errores comunes cometidos por los estudiantes, forman parte del componente de tutoría. Estos han sido codificados en el sistema como una secuencia de declaraciones condicionales.

¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?

La ruta de resolución depende de una serie de decisiones del usuario. Estos son principalmente a) el número de cantidades/variables que están representadas por letras; b) cuáles son estas cantidades; y c) el orden en que se determinan los valores de cantidad. Esto produce múltiples caminos de resolución válidos que pueden conducir a expresiones algebraicas muy diferentes para la misma cantidad o a diferentes ecuaciones. Básicamente utilizan hipergrafos para representar todas las posibles soluciones de un problema

Otros hallazgos relevantes

El sistema admite la incorporación de nuevos problemas sin la necesidad de ser programados. Esto permite al usuario introducir un enunciado del problema, junto con las lecturas analíticas que corresponden al problema en cuestión. La aplicación también acepta una lista de posibles descripciones para cada cantidad. Una vez que se han introducido los datos requeridos, la aplicación puede construir un archivo XML, que se almacena automáticamente en la base de datos

Código del artículo
[A08]
Nombre del artículo
An intelligent tutoring system incorporating a model of an experienced human tutor.
¿Cuál es el contexto del artículo?
Este artículo se centra en una investigación sobre lo que dificulta la habilidad de traducir una situación del mundo real al lenguaje del álgebra y reportan el descubrimiento de una habilidad "oculta" en la simbolización de los problemas algebraicos.
¿Qué datos o detalles útiles nos brinda este artículo?
Básicamente aporta con un método para ayudar al estudiante a extraer del enunciado del problema las variables y ecuaciones necesarias para la correcta resolución del problema.
Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados
No aplica.
Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo
No aplica.
¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?
No aplica.
¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?
Básicamente el sistema analiza la entrada del estudiante, y realiza una serie de preguntas al estudiante como lo haría un tutor humano. Para esto utiliza tres estrategias de tutoría diferentes: 1)Articulación concreta: se suele utilizar tres preguntas del tipo Q_compute, Q_explain y Q_generalize. 2) Variable introducida: esta estrategia utiliza preguntas Q_decomp y Q_substitute. 3) Explicar primero en lenguaje natural: Q_represents_what and Q_explain_verbal.
Además el sistema tiene reglas de error para aplicar a preguntas que no sean preguntas de simbolización.
¿El artículo describe los tipos de feedback entregados a los estudiantes?

No aplica.
¿ Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?
No aplica.
Otros hallazgos relevantes
No aplica.

Código del artículo
[A09]
Nombre del artículo
Intensive scaffolding in an intelligent tutoring system for the learning of algebraic word problem solving.
¿Cuál es el contexto del artículo?
El contexto de este artículo recae en el análisis del efecto del "andamiaje" en el aprendizaje de resolución de "word problems" algebraicos. El cual se utiliza en los TC para guiar o brindar asistencia a los estudiantes al momento de resolver ejercicios matemáticos.
¿Qué datos o detalles útiles nos brinda este artículo?
Presenta un ITS para el aprendizaje de la resolución de problemas verbales (word problems) algebraicos que utiliza las siguientes estrategias de andamiaje conceptual: 1) Guiar la resolución restringiendo las acciones potenciales que un usuario puede realizar en cada etapa 2) Suavizar la entrada en la resolución del problema al hacer visible el nombre de las cantidades involucradas. 3) Proporcionar mensajes de notificación cuando una acción no es correcta. 4) Proporcionar ayuda para superar las lagunas conceptuales. Además comentan que demasiado andamiaje puede implicar un menor aprendizaje y definen dos tipos de andamiaje: Andamiaje intensivo: andamiaje conceptual ilimitado. Implica que cuando el estudiante demanda ayuda siempre se le da, permitiendo así continuar con la resolución, informando sobre cómo realizar el siguiente paso. Andamiaje no intensivo: El estudiante recibe apoyo, pero nunca se le dice exactamente qué hacer.
Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados
El sistema utiliza el Método cartesiano (MC) como esqueleto sobre el que estructura el AWPS (álgebra word problems solving) obligando al alumno a completar un paso antes de pasar al siguiente. De esta manera la plataforma proporciona implícitamente conocimiento conceptual sobre el MC. En particular, el ITS ha sido diseñado para que solo se permitan ciertas operaciones en cada paso del proceso de resolución de problemas.
Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo
Cuando se selecciona un problema, el programa muestra el enunciado, y sólo permite al usuario la posibilidad de realizar tres acciones diferentes: asignar un número, asignar una letra o asignar una expresión. Estas tres

opciones se refieren, respectivamente: a dar un valor a una cantidad conocida; asignar una letra a una cantidad desconocida; y asignar una expresión aritmética o algebraica a una cantidad desconocida.

Las expresiones se construyen utilizando un componente gráfico similar a una calculadora. Este componente contiene un botón para cada operador aritmético y otro para cada cantidad previamente definida. Además, la construcción de la ecuación o ecuaciones solo se permite una vez que se ha asignado una expresión algebraica a cada cantidad desconocida del enunciado.

¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?

No aplica.

¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?

Para andamiar (guiar) la identificación de las cantidades necesarias para resolver un problema y permitir la inferencia de esquemas conceptuales, el tutor ofrece un menú desplegable con nombres de cantidades. De esta forma, el tutor intencionadamente hace visibles todas las magnitudes que intervienen en la resolución del problema.

El tutor en un recuadro le permite al alumno seleccionar entre expresión, valor o letra. Luego presenta un Input en donde el alumno debe poner el valor correspondiente y luego debe seleccionar la descripción dentro de un menú desplegable.

¿El artículo describe los tipos de feedback entregados a los estudiantes?

Cualquier acción del usuario es validada automáticamente por el sistema. En cada paso, el sistema puede identificar si una acción es correcta o no. Cuando un usuario comete un error, el sistema genera un mensaje simple que informa al usuario que la acción no es correcta y lo insta a intentarlo nuevamente. En principio, no se proporciona ayuda adicional con el tipo de error.

¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?

No aplica.

Otros hallazgos relevantes

El ITS también proporciona sugerencias a pedido, las cuales tienen como objetivo llenar los vacíos conceptuales que impiden al estudiante avanzar en el proceso de resolución. Las sugerencias bajo demanda siguen una estructura de tres niveles.

En el primer nivel: El sistema proporciona ayudas para ayudar al estudiante con el método de resolución y no contiene ninguna información relacionada con el contenido del problema.

En el segundo y tercer nivel, los mensajes de ayuda se contextualizan dentro del problema que se está resolviendo.

Finalmente, si el usuario no puede continuar y demanda más ayuda, el sistema le explica exactamente qué hacer para continuar con la resolución.

Código del artículo

[A10]

Nombre del artículo

Systematic literature review of STEM self-study related ITS.
¿Cuál es el contexto del artículo?
El artículo se centra en realizar una revisión sistemática de ITS de varios sectores educativos.
¿Qué datos o detalles útiles nos brinda este artículo?
Principalmente nos aporta con algunos tipos y características de la retroalimentación en los ITS.
Describir los elementos o características claves del modelamiento de los "word problems" o problemas contextualizados
No aplica.
Describir cómo implementan los "word problems" o ejercicios contextualizados en un tutor cognitivo
No aplica.
¿El artículo describe competencias y/o habilidades asociadas a la solución de los "word problems" o ejercicios contextualizados?
No aplica.
¿Cómo enfrentan el problema de transformar el enunciado en un conjunto de variables, expresiones, etc. Para ayudar a la resolución del problema?
No aplica.
¿El artículo describe los tipos de feedback entregados a los estudiantes?
El artículo realiza una investigación para explorar cuáles son las formas de retroalimentación y ayuda que se presentan en los ITS, entre los cuales identificaron los siguientes métodos de ayuda: <ul style="list-style-type: none"> - Hints - Tips - Explanation of a concept - The indication of next steps to solve a problem - Feedback on learner's answer correctness - Adapt help based on student performance - Step by step assistance for problem-solving - Motivational messages / encouragement - Adapt learning material or tutoring strategy to student knowledge - Ask student questions of why/how they reached the solution or why did he opt an input En cuanto a los métodos de feedback identificados : <ul style="list-style-type: none"> - Sugiere materiales de aprendizaje (web, libros, videos, etc) - Comentarios personalizados basados en las características del alumno - Explicación del concepto o de un paso(s) de solución - Señalar los errores de los estudiantes - Dar la respuesta correcta - Mostrar ejemplos resueltos - Comparar la respuesta del estudiante con la respuesta del sistema
¿Como el artículo se enfrenta con el espacio de soluciones posibles al resolver estos tipos de problemas?

No aplica.

Otros hallazgos relevantes

No aplica.

ANEXO B: Encuesta por ejercicio

Retroalimentación primer ejercicio

1. Creo que este ejercicio es útil para aprender matemáticas.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

2. Con este ejercicio aprendí algo nuevo.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

3. Creo que este ejercicio, con enunciado y aplicación real, es mejor para aprender que un ejercicio explícito que sólo tiene expresiones matemáticas para calcular.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

4. Resolver este ejercicio me ayuda a reflexionar sobre la importancia de aprender matemáticas

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

ANEXO C: Encuesta SUS

Test de Usabilidad

Esta encuesta tiene por objetivo conocer tu opinión respecto de varios aspectos relacionados con la usabilidad del sistema. Tus respuestas sólo serán usadas como insumo para plantear mejoras del sistema.

1. Creo que usaría este sistema frecuentemente.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

2. Encuentro este sistema innecesariamente complejo.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

3. Creo que el sistema fue fácil de usar.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

5. Encontré que las diversas funciones de este sistema estaban bien integradas.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

6. Pensé que había demasiada inconsistencia en este sistema.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

7. Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

8. He encontrado el sistema muy complicado de usar.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

9. Me sentí muy seguro/a usando el sistema.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

10. Necesité aprender muchas cosas antes de ser capaz de usar este sistema.

1	2	3	4	5		
Totalmente en desacuerdo	<input type="radio"/>	Totalmente de acuerdo				

ANEXO D: Encuesta satisfacción

Test de Satisfacción.

Esta encuesta tiene como propósito conocer tu nivel de satisfacción con los ejercicios resueltos.

1. Menciona mínimo 1 y máximo 3 aspectos que te gustaron de los ejercicios vistos

1. _____

2. _____

3. _____

2. Menciona mínimo 1 y máximo 3 aspectos que no te gustaron o te parecieron confusos de los ejercicios vistos.

1. _____

2. _____

3. _____

3. Recomendaría estos ejercicios a compañeros/as.

1 2 3 4 5

Totalmente en desacuerdo



Totalmente de acuerdo

ANEXO E: Resultados de encuestas

Disponible en:

https://docs.google.com/spreadsheets/d/1Wh1KCz5QT8_LecxNPLUEa4XJKkU4Cf7KpvIMTw-8qx0/edit?usp=sharing