



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

DISEÑO, DESARROLLO Y USABILIDAD DE SOFTWARE MATEMÁTICO PARA EL DESARROLLO DEL PENSAMIENTO CRÍTICO EN EL AULA

VICENTE JOSÉ GARDULSKI BUDINICH

Tesis para optar al grado de
Magíster en Ciencias de la Ingeniería

Profesor Supervisor:
MIGUEL NUSSBAUM VOEHL

Santiago de Chile, (Junio, 2018)

© 2018, Vicente Gardulski



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

**DISEÑO, DESARROLLO Y USABILIDAD DE
SOFTWARE MATEMÁTICO PARA EL
DESARROLLO DEL PENSAMIENTO CRÍTICO
EN EL AULA**

VICENTE JOSÉ GARDULSKI BUDINICH

Tesis presentada a la Comisión integrada por los profesores:

MIGUEL NUSSBAUM VOEHL

CARLOS BONILLA MELÉNDEZ

PABLO CHIUMINATTO MUÑOZ

IGNACIO CASAS RAPOSO

Para completar las exigencias del grado de
Magíster en Ciencias de la Ingeniería
Santiago de Chile, (Junio, 2018)

A mis Padres, hermanos y amigos,
que me apoyaron mucho.

INDICE GENERAL

| | Pág. |
|--|------|
| INDICE DE TABLAS | iv |
| INDICE DE FIGURAS..... | v |
| RESUMEN..... | vi |
| ABSTRACT | viii |
| 1. Motivación | 1 |
| 1.1 Habilidades del siglo XXI..... | 1 |
| 1.1.1 Aprendizaje en el siglo XXI..... | 2 |
| 1.1.2 Habilidades requeridas en el siglo XXI y tecnología en el aula | 4 |
| 1.2 Interfaces humano-computador..... | 6 |
| 2. Resumen del trabajo realizado | 9 |
| 3. Plan de trabajo..... | 10 |
| 4. Integración de contenidos mediante herramientas tecnológicas | 11 |
| 2.1.1 Método de Nielsen | 14 |
| 2.1.2 El método de Wharton | 16 |
| 3. Tareas implementadas | 18 |
| 3.1 Metodología | 18 |
| 3.2 Descripción del software..... | 19 |
| 3.3 Proceso de desarrollo | 24 |
| 3.3.1 Diseño de los problemas | 24 |
| 3.3.2 Desarrollo de los prototipos | 26 |
| 3.3.3 Aplicación de la prueba de diagnóstico | 28 |
| 3.3.4 Implementación y testeo mediante evaluación conjunta..... | 30 |
| 4. Resultados y discusión | 35 |
| BIBLIOGRAFIA | 48 |
| A N E X O S | 52 |
| Anexo 1 : Estructura y Motor de la aplicación (Gonzalez, 2017)..... | 53 |
| 1.1.1 Arquitectura. | 53 |
| 1.1.2 Diagrama de clases UML..... | 54 |
| 1.1.3 Implementación de la calculadora..... | 57 |
| 1.1.4 Implementación de registro de datos..... | 59 |
| Anexo 2 : Ejercicios del pre test (Cáceres, 2017) | 63 |

INDICE DE TABLAS

| | Pág. |
|--|------|
| Tabla 1: Descripción de los valores del pre-test | 30 |
| Tabla 2: ANOVA | 30 |

INDICE DE FIGURAS

| | Pág. |
|---|------|
| Figura 1: Modelo de aprendizaje propuesto | 12 |
| Figura 2: Modelo de scaffolding | 21 |
| Figura 3: Capturas de pantalla de la aplicación por sesiones | 22 |
| Figura 4: Esquema complejizado de recorrido en la aplicación | 27 |
| Figura 5: Pauta conjunta de evaluación | 3° |
| Figura 6: Caso de uso de pauta conjunta de evaluación | 32 |
| Figura 7: Prototipos antes y después de la aplicación de la rúbrica | 32 |
| Figura 8: Comparación de maquetas antes y después de la evaluación | 34 |
| Figura 9: Tablas de resumen de tiempo por grupo en segundos | 37 |
| Figura 10: Resumen preguntas de satisfacción | 38 |
| Figura 11: Captura del registro generado por la aplicación | 40 |
| Figura 12: Resumen acciones grupo control y experimental | 40 |
| Figura 13: Porcentajes de acciones inapropiadas por grupo | 44 |

RESUMEN

Actualmente es tendencia la investigación de la integración de habilidades al curriculum, especialmente desde el marco de las llamadas habilidades del siglo XXI. La consecuencia de cambiar la orientación de los programas e incluir estas habilidades por medio de nuevas tecnologías implica transformación no solo de la evaluación de estas competencias, sino de la manera en como estas se implementan.

El presente trabajo presenta un modelo de evaluación de usabilidad, diseño y desarrollo de un *software* basado en el uso de tablets en el aula. Esta aplicación fue diseñada siguiendo las estrategias del Problem Based Learning, y a las habilidades del siglo XXI; competencias centrales para desempeñarse exitosamente como ciudadano y profesional en la era de las tecnologías informáticas.

El software se desarrolló con el objetivo de potenciar el aprendizaje de los estudiantes (sujetos) mediante interacciones con el software, que facilitan la indagación y razonamiento crítico. Esto se logró gracias al trabajo e interacción de los alumnos con el software que les brindó una retroalimentación adaptativa. Para evaluar la implementación experimental, se trabajó en dos versiones del software, con y sin retroalimentación. Se determinó la validez de los contenidos en el dominio en el que se aplicó por medio de un pre y post test de medición. Estos grupos se construyeron para apoyar el análisis de la metodología de enseñanza-aprendizaje que corresponde a la segunda etapa de esta investigación.

La validez de la implementación del software se midió a través de un constructo de pautas de evaluación de usabilidad y fue refrendado durante la etapa de desarrollo. Luego, se

pudo examinar la diferencia en la implementación de los diferentes grupos y su caracterización, disminuyendo el sesgo de adaptación al software.

Específicamente, en esta tesis se expone el desarrollo de las pautas de diseño de usabilidad que ayudaron en la metodología de implementación del software. Estas pautas sirvieron para determinar los cambios de diseño en base a las heurísticas y workflows de uso de las metodologías planteadas. Estos resultados se compararon con los tiempos de ejecución, la caracterización de las acciones que los alumnos realizaron en la intervención y las observaciones de satisfacción que se realizaron. Esto sirvió para cumplir con el objetivo de esta tesis, esto fue medir el impacto que tienen los cambios de diseño en base a la rama de usabilidad denominada User Experience.

También se encontraron diferencias entre las implementaciones en los grupos aplicados. Estas señalan las influencias que puede tener la retroalimentación en el desempeño que obtienen los sujetos en la aplicación. Se caracterizó de la misma manera las diferentes componentes de usabilidad en base al constructo de pautas propuesto. Estas diferencias deberían ser consideradas en futuras investigaciones, por posibles influencias de la adaptación a la tecnología sobre la evaluación de habilidades en la escuela.

Esta Tesis tuvo el apoyo del proyecto FONDECYT / CONICYT 1150045

ABSTRACT

Nowadays, research into the integration of skills into the curriculum is trend, especially within the framework of the so-called skills of the 21st century. The consequence of changing the orientation of programs and including these skills through new technologies implies transformation not only in the evaluation of these competencies, but also in the way they are implemented.

This paper presents a model of usability evaluation, design and development of a software based on the use of tablets in the classroom. This application was designed following the strategies of Problem Based Learning, and the skills of the 21st century; core competencies to successfully perform as a citizen and professional in the age of information technology.

The software was developed with the aim of enhancing the learning of the students (subjects) through interactions with the software, which facilitate research and critical thinking. This was achieved through the students' work and interaction with the software that provided them with adaptive feedback. To evaluate the experimental implementation, two versions of the software were developed, with and without feedback. The validity of the contents in the domain in which it was applied was determined by means of a pre and post-test.

The validity of the software implementation was measured through a construct of usability evaluation guidelines and was endorsed during the development stage. Then, the difference in the implementation of the different groups and their characteristics could be examined, reducing the bias of adaptation to the software.

Specifically, this thesis presents the development of usability design guidelines that helped in the methodology of software implementation. These guidelines were used to determine the design changes based on the heuristics and workflows of use of the proposed methodologies. These results were compared with the execution times, with the characterization of the actions that the students carried out in the intervention and with the observations of satisfaction that were made. This served to fulfill the objective of this thesis, this was to measure the impact of design changes based on the usability branch called User Experience.

Differences were found between implementations in the applied groups. These indicate the influences that feedback has on the performance of the subjects in the application. The different usability components were characterized in the same way based on the proposed guideline construct. These differences should be considered in future research, because of possible influences of adaptation to technology on the assessment of skills in school.

This Thesis had the support of FONDECYT / CONICYT 1150045 project

Keywords: User Experience, Usability, 21st century skills

1. MOTIVACIÓN

1.1 Habilidades del siglo XXI

Durante los últimos años las tecnologías de la información y comunicación (TIC) han transformado varios aspectos de nuestra sociedad de manera radical; desde como las personas se comunican hasta la forma en que trabajamos. Incluso, actualmente, la interacción con la tecnología ha influido en como la sociedad se desenvuelve y comunica en el día a día. Se está frente a una revolución tecnológica que podría ponerse al nivel de invenciones como la imprenta o procesos como la revolución industrial.

Los niños que han crecido en la última década han vivido esta transformación desde el día cero, los denominados “nativos digitales” (Prensky, 2001). Su mundo se ha configurado alrededor de invenciones como la televisión, los teléfonos celulares, la Internet y los juegos interactivos. Estos hacen que hoy en día ellos sean capaces de escuchar música, trabajar y chatear, todo al mismo tiempo.

Como lo describe Prensky, los padres de estos nativos digitales se convierten en inmigrantes digitales. Dado que al no haber nacido bajo el milenio tecnológico se ven confundidos con las maneras que sus hijos se comunican. A los padres se les hace difícil entender la capacidad que sus hijos tienen de desarrollar múltiples tareas al mismo tiempo—mientras cumplen con sus trabajos o deberes escolares (Prensky, 2001).

Todos estos fenómenos no sólo provocan que actualmente existan nuevas maneras de comunicarse y relacionarse muy alejadas a las prácticas tradicionales de las décadas anteriores; también tiene un impacto en la manera en cómo hoy se sociabiliza y se aprende. Junto con las tecnologías debieran cambiar también las estrategias de enseñanza, ya que

las múltiples formas de aprender de los alumnos están evolucionando aceleradamente. Así, si las prácticas tradicionales de enseñanza no cambian y aprovechan las nuevas tecnologías, estas solo serán un accesorio caro que no tendrá ningún impacto en la calidad de la educación actual.

1.1.1 Aprendizaje en el siglo XXI

A la fecha de hoy la Internet cumplirá más de 30 años desde sus primeros pasos comerciales en Estados Unidos. Desde esa fecha se ha visto cómo las tecnologías que solían ser análogas se han transformado a tecnologías digitales; y las herramientas para la educación no son un caso aislado.

Estos cambios en las tecnologías no solamente están influenciando el aprendizaje de las masas, sino también cómo los profesores están enseñando en el aula. Estudios actuales postulan que sería beneficioso hacer las clases desde una perspectiva constructivista con la ayuda de la utilización de las TIC, para así mejorar los resultados (Veletsianos, 2016). Pero debe considerarse que en este contexto el constructivismo está mirado desde la perspectiva del aprendizaje y no de la enseñanza. Es más, a lo largo de esta década se han encontrado varios ejemplos en los que esta perspectiva no se cumple (Pedró, 2009) y varios son los docentes que se rehúsan a usar las tecnologías en la sala de clases.

Autores como Prensky, Oblinger y Pedró (Prensky, 2001; Oblinger y Oblinger, 2005; Pedró, 2009) señalan que los estudiantes de hoy en día representan la primera generación que ha crecido rodeada de tecnologías. Por ese motivo, cuentan con características y capacidades respecto a las TIC que las diferencian de generaciones anteriores. (Moreno, 2016).

Aprender y enseñar a alumnos que crecieron en la era digital no solo significa un reto para los profesores, que deben adaptarse al manejo tecnológico a la par de sus alumnos. También significa adaptar la infraestructura y la organización de las escuelas para reflejar los cambios. Autores como Ledesma han estudiado estos cambios en el aula y han propuesto una serie de medidas que se hacen necesarias para atravesar estos cambios, por ejemplo: contar con un plan nacional de desarrollo de TIC en el aula; tener una infraestructura adecuada para desarrollar este plan; preparar a los profesores para la adopción de las TIC en el aula; y por último tener una visión de la integración de las TIC en la sala de clases (Ledesma, 2005).

Como es natural pensar, estos pre-requisitos por sí solos no son lo suficientes para generar mejoras en la integración de las TIC en el aula, porque la integración de las tecnologías con el aprendizaje no solo requiere estos factores, sino que también una adaptación en los métodos de enseñanza por parte de los profesores. Los hallazgos de estudios como el de Ledesma indican que deben hacerse cambios en las culturas de enseñanza-aprendizajes para responder a los cambios y desafíos de la era digital (P. 7 Ledesma, 2005). Bajo este razonamiento los planes de educación deben construirse para responder y avanzar con la nueva era y con los nativos digitales. Específicamente deben adaptarse para cumplir con las competencias que los alumnos deben tener para poder desarrollarse en estos contextos tecnológicos.

Así, cobra sentido encontrar una forma de evaluar las tecnologías que se implementan en el aula y las habilidades que estas fomentan en términos del logro y de la adaptación-recepción que tienen los alumnos de estas. Para esto se necesita definir un método que facilite el desarrollo de software con foco en el aprendizaje del alumno, sin tener sesgos

de uso por la adaptación y dificultad que significa su utilización, para su posterior evaluación.

1.1.2 Habilidades requeridas en el siglo XXI y tecnología en el aula

Frente a la necesidad de determinar cuáles son las habilidades que los alumnos deben manejar para tener éxito en el siglo XXI, se organiza el *Partnership for 21st Century Skills (P21 – <http://www.p21.org>)*. Los miembros de esta organización tienen la misión orientar los planes de educación, en pos del desarrollo de las habilidades que permitan formar mejores profesionales que se adapten a los cambios tecnológicos que se están viviendo (P. 45 Trilling, 2009). Hay autores como Levy y Murnane que han hecho análisis del impacto de los cambios tecnológicos y de cómo la formación de estas nuevas habilidades está incidiendo en el cambio de la oferta y la demanda de trabajos y profesionales (Levy et. Al, 2012).

El argumento que exponen los autores mencionados anteriormente se basa en las diferencias que tienen las maneras de pensar de los humanos, versus el razonamiento que tienen las máquinas. La demanda por trabajadores que desarrollan tareas que pueden ser mecanizadas está disminuyendo, y así continuará a medida que las máquinas sean capaces de imitar esas capacidades humanas de manera más precisa. Sin embargo, la demanda por expertos en razonamiento y comunicación con estas máquinas, habilidades que los humanos tenemos y los computadores no han podido imitar todavía, va en aumento; y continuará creciendo a medida que se complejizan los sistemas de inteligencia artificial y automatización.

Es en el punto anterior donde se hacen cruciales las habilidades propuestas por el P21, ya que la demanda por profesionales que manejen la comunicación e interacción con las máquinas seguirá aumentando con el tiempo. Así, la concientización de la implementación de currículums que potencien el aprendizaje de estas habilidades y su interacción con la tecnología cobra sentido (Levy et. Al, en Moss 2006), y se hace de interés estudiar la integración que tienen las herramientas tecnológicas en el aula.

Implementar estas nuevas habilidades, junto con los contenidos del currículum tradicional, implica un cambio en el paradigma de enseñanza que aún no ha sido completamente definido. Cambiar la orientación del currículum en pos de estos nuevos objetivos implica cambiar la manera en que este se implementa, y con ello, las formas y recursos que se tienen para enseñar. Ya lo señalan autores como Benito:

"Aunque sería lógico pensar que el desarrollo de las Tecnologías de la Información y la Comunicación (TIC) incidiría en la mejora sustancial en el ámbito de la Educación [y de la implementación de las habilidades del siglo 21], la realidad muestra que en la actualidad ni el uso de estos recursos en la escuela ha llegado a ser masivo, ni parece haber cambiado sustancialmente las pautas de formación del profesorado, ni la forma en que éste entiende los procesos de aprendizaje."

(Benito, 2009).

Una de las principales barreras que ha tenido la implementación de las TIC en el aula es debido a los sesgos de la enseñanza y evaluación tradicional que predomina en el sistema. Estudios han demostrado que, pese a que el sistema facilita la integración de tecnologías en el aula, padres y profesores todavía se sienten inseguros acerca de estos nuevos métodos y de la capacidad que tienen para generar los mismos logros en los exámenes nacionales estandarizados. A eso se le suma el miedo de que el uso de tecnologías afecte el

rendimiento de manera negativa, en comparación a los métodos de enseñanza tradicional (Grünbaum, 2004).

Lo que se debe entender en este punto, es que implementar estas habilidades en el currículum necesariamente significa tener una mayor integración de la tecnología en el aula. Esto no solo comprende dejar de lado los prejuicios que genera el paradigma tradicional, sino que buscar una manera de integrar la tecnología de manera natural en el aula. Es razonable pensar en que parte de la adaptación que debe surgir va de mano con la formación de los evaluadores y su preparación para usar las herramientas tecnológicas, en conjunto con la orquestación de las clases. Pero pese a que existe la necesidad de implementar estas nuevas metodologías, las implementaciones no siempre son sencillas (Reimers & Chung, 2016).

Así, surge la necesidad de integrar los contenidos actuales de los currículos de aprendizaje en la línea de integración y potenciamiento de los *21st Century Skills*. Para esto se debe idear un modelo de aprendizaje que considere estas capacidades en una plataforma tecnológica.

1.2 Interfaces humano-computador

Parte importante de los estudios de integración de tecnología y la interacción que se tiene con ellas se basa en corrientes de investigación de interfaces humano computador. Esta no es sino el estudio de cómo la tecnología influencia el trabajo y las actividades que desarrollamos con ella. Asociada a las interfaces, existe una disciplina de diseño denominada *User-Centered Design*, o diseño centrado en el usuario, la que se enfoca en

cómo diseñar tecnología que se integre de la manera más natural posible con el usuario. Uno de los aspectos claves de esta disciplina es la “usabilidad”, que, por lo general, está medida en términos de eficiencia, efectividad y satisfacción que tiene el usuario en su interacción con los sistemas. Autores como Dix señalan la importancia de estos conceptos, en conjunto con cómo los usuarios se sienten a medida que interactúan con el sistema (Dix, 2009). Citando la definición ISO de usabilidad, se define como:

“La medida en que un producto puede ser utilizado por usuarios específicos para alcanzar objetivos específicos con eficacia, eficiencia y satisfacción en un contexto específico de uso”

(ISO P. 8, 1998).

Los aspectos claves de esta definición son la efectividad, la eficiencia y la satisfacción que presenta el usuario al interactuar con el software. Esta evaluación cobrará importancia al tratar de relacionar la experiencia del usuario y el impacto que esta tiene en los logros, ya que lo que se busca es desarrollar un sistema que no tenga los sesgos antes planteados (de adopción de tecnologías y resistencia al cambio) para poder tener un progreso que no dependa de la usabilidad de la herramienta desarrollada.

Durante la última década se ha visto cómo varios intentos por integrar cambios en las metodologías de enseñanza-aprendizaje mediante nuevas tecnologías en el aula han fallado al momento de implementarse (Janssen, 2013). Autores como Hargreaves y Fink atribuyen las fallas a que “Los cambios en educación son fáciles de proponer, difíciles de implementar y extraordinariamente difíciles de mantener” (P. 6, 2006). Bajo este contexto es que se sitúa la integración de TIC actualmente en la sala de clases y al mismo tiempo es que cobra importancia evaluar el grado de aceptación que estas tecnologías tienen; tanto

con los profesores como con los alumnos. En este punto es crucial relacionar la efectividad, eficiencia y satisfacción, que tienen las aplicaciones en el aula, con el logro y avance que consiguen los alumnos mediante el uso de estas herramientas.

2. RESUMEN DEL TRABAJO REALIZADO

El trabajo realizado por el autor de esta tesis se efectuó en conjunto a un equipo de investigación que realizó un conjunto de actividades a decir:

- Selección de estrategia de enseñanza-aprendizaje para incorporar al software
- Esto es trabajo compartido del equipo
- Desarrollo del software. Trabajo del equipo con apoyo de programador externo
- Testeo del software realizado por el autor de esta tesis
- Pruebas de usabilidad realizadas por el autor de esta tesis
- Intefgracion del software en el aula. Trabajo conjunto del equipo
- Modelo de evaluación y análisis de los resultados del modelo de enseñanza-aprendizaje. Realizado por el resto del equipo
- Evaluación y análisis de resultados de testeo realizado por el autor de esta tesis

3. PLAN DE TRABAJO

A continuación se detalla el plan de trabajo ejecutado por el equipo de investigación y el autor de esta tesis. Este se dividió en las etapas de:

- Selección de estrategia de enseñanza- aprendizaje
- Desarrollo del software y testeo
- Integración en el aula y mejora continua del software
- Medición y evaluación

4. INTEGRACIÓN DE CONTENIDOS MEDIANTE HERRAMIENTAS TECNOLÓGICAS

Una de las principales definiciones de integración de acuerdo con el *Webster's New World Dictionary*, es "ser o llegar a ser completo", "unir partes a un todo". Así, integrar se refiere a completar/articular las partes para formar un todo. Entonces el término de integración de tecnologías se refiere a su uso como parte integral del currículum y no como un apéndice (Ilabaca, 2003).

Por otra parte, una de las definiciones más populares sobre el currículum es la que hace Stenhouse en 1987 en su libro *Diseño y Desarrollo curricular* (Stenhouse, 1987). El autor define el currículum como “[...] el proyecto que determina los objetivos de la educación escolar, es decir, los aspectos del desarrollo y de la incorporación a la cultura que la escuela trata de promover y propone un plan de acción adecuado para la consecución de esos objetivos” (DCB, P. 21 en Zabalza, 1987).

Dado lo anterior, es de importancia que para la integración de las TIC no solo se tome en consideración el contenido de las asignaturas del currículum, sino también su relación con el modelo educativo que se imparte en el aula.

Esta investigación tuvo como objetivo principal la implementación de un software que sirve para enseñar los contenidos de un grupo de alumnos de quinto básico. El principal foco es medir la usabilidad y cuantificar la adaptabilidad que los usuarios tuvieron a la aplicación para correlacionar esto con el logro obtenido. Así el software se implementó bajo una metodología de aprendizaje orientado al logro, perteneciente al Problem Based Learning (Wood, 2003), sobre la que se hace referencia más adelante en esta investigación.

Uno de los primeros pasos del equipo fue definir el grupo sobre el que se realizaría el estudio y el programa educativo sobre el que se realizarían las actividades. El modelo de Aprendizaje propuesto por el equipo se basa en la indagación como el proceso de reflexión que lidera el aprendizaje (o logro mencionado anteriormente). En la siguiente figura se ilustra el proceso.

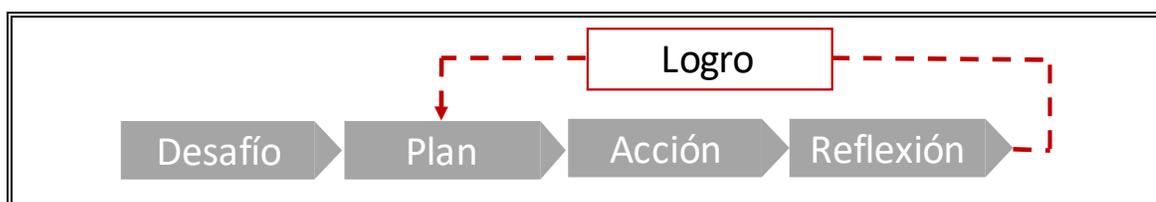


Figura 1: Modelo de aprendizaje propuesto

La metodología propuesta se basa en la realización de desafíos por parte de los alumnos para la creación de un plan, que permita obtener la solución del problema propuesto. El sistema presenta una serie de desafíos de dificultad incremental, los que pretenden generar un logro en el aprendizaje a través de la reflexión que genera el usuario al realizar sus acciones.

Uno de los desafíos que se tuvo en la implementación de esta aplicación fue la interacción que el usuario tenía con ella y cómo esta incide en el logro y aprendizaje mismo del alumno. Así en esta tesis se propone generar un método para la incorporación de las TIC en el sistema educativo de nivel básico por medio de pautas de diseño y testeo, facilitando la intervención con el software en el aula y su mejora continua. Esto con el fin de evaluar el impacto que tiene la usabilidad del software tanto cuantitativa como cualitativamente.

En primer lugar, se realizó un estudio de las herramientas actuales para el testeo de la usabilidad de las aplicaciones, para luego generar una comparación y proponer un método de integración de estas con la mejora continua de la aplicación a desarrollar. Esto con el fin de disminuir la brecha de adaptabilidad que existe por parte del usuario con el sistema y su posible injerencia en el logro que presenten en el modelo de aprendizaje propuesto. En la siguiente sección se detallan las herramientas utilizadas en base a los paradigmas de usabilidad que se tienen actualmente.

En segundo lugar, se generó un test de diagnóstico para determinar la importancia que tiene el nivel de dificultad de los problemas y descartar que el logro estuviera relacionado con la usabilidad de la herramienta. Los resultados de este test y sus conclusiones se detallan más adelante. Por último, se desarrolló un post-test, para medir el avance que pudieron tener los alumnos mediante esta experiencia, los resultados de esta última prueba y su relación con el *User Experience* se detallan en la segunda etapa de esta investigación.

4.1 Modelos de evaluación de usabilidad

Como se mencionó anteriormente, es de interés de los grupos de desarrollo familiarizarse con el usuario y aplicar técnicas que permitan el fácil manejo e interacción con las interfaces que se crean. Actualmente existen dos grandes categorías de evaluación de usabilidad; estas son la usabilidad por medio de parámetros de diseño y heurísticas propuestas por Nielsen (que no requieren entrenamiento para usarse); y por otra parte la usabilidad en base a la realización efectiva de ciertas metas o tareas propuestas por Wharton (que requieren un conocimiento de la teoría para poder ser implementados) (Hollingsed, 2007). Pese a que estos métodos fueron desarrollados sobre la década de los

90, todavía son aplicables en los contextos de interacción actuales y su uso prevalece en el desarrollo de las aplicaciones contemporáneas.

2.1.1 Método de Nielsen

El método de evaluación heurística propuesta por Nielsen se basa en una serie de patrones o requisitos de diseño que el sistema debe cumplir. (Nielsen, 1994) Luego un pequeño grupo de expertos evalúa la usabilidad de las interfaces e implementa los cambios por medio de estos principios y guías. Estas se contrastan con el funcionamiento y diseño, de la aplicación en cuestión en su contexto. Así diferentes opiniones o evaluadores pueden determinar la usabilidad de la interfaz de diferente manera, según su entendimiento de los conceptos. Como la describe el autor, la usabilidad se puede definir en términos generales en tres aspectos: eficiencia, eficacia y satisfacción; que se detallan a continuación (Sasse P. 615, 1999).

Eficiencia: la eficiencia de la interfaz del sistema se plantea a través de la rapidez de adaptación que tiene el usuario de aprender a interactuar con el sistema. Esto se mide mediante la capacidad del usuario de realizar la tarea, que guarda relación con cumplir las expectativas de uso que el usuario tiene de estas. La medición de la eficiencia se hace específicamente con el grupo de heurísticas correspondientes a: *Relación del sistema con el mundo real, Consistencia y estándares, Ayuda y documentación* (Hornbæk, 2006). Éstas están asociadas a los tiempos de uso que tiene el usuario en el sistema.

Eficacia: la eficacia se refiere a cuánto debe el usuario memorizar para poder interactuar de manera eficiente con la interfaz. Para su evaluación el autor propone la clasificar la

efectividad del sistema mediante la observación de tareas que debe realizar el usuario para cumplir con la meta. La medición de la eficacia se hace específicamente con el grupo de heurísticas correspondientes a: *Control del usuario y libertad, Visibilidad del estado del sistema, Flexibilidad y eficiencia de la interfaz* (Hornbæk, 2006). Estas están asociadas a la tasa de éxito que tiene el usuario en el sistema.

Satisfacción: En este último grupo se encuentra una medida subjetiva que evalúa la comodidad y la aceptabilidad que tiene la interfaz por parte del usuario final. La medición de la satisfacción Específicamente se hace mediante el grupo de heurísticas correspondiente a: *Prevención de errores, Reconocimiento de errores, Reporte, diagnóstico y vuelta a la normalidad después de un error, Estética y el diseño minimalista*. Estas se pueden cuantificar por medio de encuestas, entrevistas focalizadas con los usuarios, o con métricas que guarden relación con las heurísticas antes mencionadas (Hornbæk, 2006).

Si bien esta categorización evalúa posibles interacciones sobre la interfaz, no evalúa la influencia de esta en la resolución de problemas o procedimientos o su aplicabilidad en contextos orientados. El punto crítico de este tipo de evaluaciones es que los diseñadores de la aplicación se desenvuelven en un contexto externo al del usuario final. De esta manera las observaciones de usabilidad podrían ser lejanas a las necesidades reales que los usuarios tienen. En este caso, los alumnos, al usar la aplicación en el aula. Para evaluaciones que requieren de una evaluación en contexto orientado, una de las herramientas más usadas es el *Cognitive Walkthrough* de Wharton.

2.1.2 El método de Wharton

El *Cognitive Walkthrough* propuesto por Wharton en la década de los noventa, busca identificar problemas de usabilidad por medio de la evaluación de uso que dan al sistema los nuevos usuarios. Para esto, mide la usabilidad de la interfaz por medio de la facilidad de uso que esta tiene. Esto lo lleva a cabo mediante un modelo de evaluación de aprendizaje y uso basado en posibles tareas que desarrolla el usuario. (Wharton, 1994). Este método puede ser usado a lo largo del proceso de desarrollo de la interfaz y tiene 4 fases (Polson, 1992).

1. El usuario determina una meta a ser cumplida en el sistema
2. El usuario determina las acciones posibles
3. El usuario selecciona la acción que lo llevará al resultado
4. El usuario realiza la acción y obtiene retroalimentación de la aplicación

Una de las críticas a este sistema es la repetitiva manera en que se llevan a cabo las observaciones en las pautas. Esto deja la identificación de problemas en un rango de procesos muy limitado, ya que se basa en la experiencia de grupos de usuarios según el observador y no en una pauta abstracta e independiente a la observación (Hollingsed, 2007). Esto es de fácil corrección si se trabaja con grupos de usuarios pequeños y se lleva a cabo una inspección en conjunto, desde las tareas más simples a las más complejas, extrapolando aquellas que no tienen que ver con el general de la muestra.

Si bien este último punto es corregible, existe una limitante con respecto a la evaluación de las actividades en sí. Estas se basan en que el método no provee detalles de qué es lo que hace a una acción más visible a los usuarios y qué tipos de acciones consideran más los usuarios al momento de realizar la tarea, es decir su categorización. Así la clasificación de las tareas es un componente fundamental a la hora de tomar acciones sobre las tareas a modificar en base al estudio (Frankze, 1995).

El método de observación descrito en este punto cobra relevancia al evaluar las componentes de usabilidad comentadas anteriormente, sobre todo la componente de satisfacción que tienen los usuarios en sistemas como el que se busca implementar. Así, si se combina el método de observación propuesto por Wharton y las heurísticas de satisfacción propuestas por Nielsen se logra generar un método que puede medir la satisfacción, efectividad, eficiencia y el logro en términos comparables a la experiencia del usuario. Dicho método permite evaluar la experiencia orientada de forma objetiva, al mismo tiempo de categorizar las acciones y los errores que se presentan en la muestra según la visión del usuario final.

En la siguiente sección se postula un método conjunto de evaluación dado por la integración de los dos métodos descritos anteriormente. Esto permite evaluar de manera sistemática la satisfacción, eficiencia y eficacia que tienen los usuarios en términos objetivos y empíricos. Este método se describe en el punto 3.3.3 de la siguiente sección.

3. TAREAS IMPLEMENTADAS

3.1 Metodología

El principal desafío que se tuvo fue el implementar el modelo cognitivo de aprendizaje descrito en la sección 1. Esto porque el proceso mismo de testeo y validación del software se hizo mediante el modelo de mejora continua propuesto. El proceso general se dividió en tres partes; la creación de los problemas, el desarrollo de software y la validación/mejora continua. Esta última una etapa fue decisiva, ya que se usó para validar el software y obtener resultados cuantitativos de su uso, al mismo tiempo que se mejoró la usabilidad.

En el proceso de desarrollo del software se consideró tanto la confección e implementación de la interfaz con el usuario, como la interacción que este tenía con ella. Esta debía estar alineada con el modelo de enseñanza-aprendizaje, para asegurar que el proceso cognitivo no tuviera sesgos atribuibles a la tecnología.

Además, fue importante analizar la interacción y el uso de la interfaz por parte de los alumnos. Para esto fue necesario el desarrollo de un sistema de evaluación de usabilidad en ambas componentes; que permitiera iterar sobre el diseño del software al mismo tiempo de su desarrollo y testeo en el aula (prueba experimental con el grupo), para así facilitar la mejora continua del software.

Este desarrollo conjunto se pensó con el objetivo de obtener datos de usabilidad, pero con foco específico en las componentes de satisfacción, eficiencia y eficacia que presentan los usuarios frente a la aplicación. Con estos datos se procedió a determinar la adaptabilidad en términos del *User-experience*. Este proceso se llevó a cabo simultáneamente con la

realización de la actividad y la orquestación en la sala de clases; estos resultados se pueden ver en la sección de análisis.

Para determinar la componente de dificultad que presentan los problemas construidos, se aplicó un pre-test que sirvió para determinar los conocimientos y dominios que los usuarios de la aplicación manejaban. Esto con la finalidad de tener un diagnóstico y poder hacer ajustes en términos de contenido. Además, para comparar el logro obtenido después de la actividad, se aplicó un post-test, que es parte de la segunda etapa de esta investigación. En esta tesis se abordará el uso del test de diagnóstico y de las pruebas de usabilidad y mejora continua, como herramienta para delimitar el sesgo que existe por la dificultad de los problemas. La exposición y discusión de los datos obtenidos se aborda en la sección 4.2.

3.2 Descripción del software

El equipo de investigación estructuró la aplicación en base a los contenidos asignados a las unidades de matemáticas que se acordaron previamente con el colegio y en el contexto de los grupos de testeo. Estas correspondieron a las unidades de geometría, conteo y álgebra de la asignatura de matemática para el ciclo de quinto básico. La aplicación se estructuró en 6 diferentes sesiones con 11 etapas de diferente dificultad cada una y de manera creciente. El desarrollo de estas se hizo por medio de *Unity*. Este es un motor multipropósito para la creación de juegos que soporta aplicaciones 2D y 3D y está basado en el lenguaje de programación C#. (González, 2017). Detalles de esto se pueden ver en el anexo 1 de esta tesis.

Para efectos de estudio del impacto en el aprendizaje del modelo propuesto se desarrollaron dos aplicaciones. Una de estas fue aplicada sobre el grupo control, con dificultad lineal. Sobre el grupo experimental fue aplicada una versión sustentada en el *scaffolding* (Sawyer, 2005) propuesto en el modelo de aprendizaje, en el cual se les dio retroalimentación según el tipo de error que cometían en la aplicación. El desarrollo de estas dos versiones permitió al equipo comparar el rendimiento de los grupos control y experimental en base al modelo de aprendizaje propuesto, estudio que se abordará en la segunda etapa de esta investigación; junto a ellos se pueden encontrar detalles del modelo de *scaffolding* implementado y el modelo de aprendizaje detallado.

La aplicación fue diseñada para permitir el uso autónomo de esta por parte de los usuarios, contemplando un set de instrucciones y tutorial introductorio a los comandos necesarios para desarrollar las actividades de esta. El funcionamiento de las etapas fue similar en cada sesión, donde los alumnos debían resolver la pregunta planteada para luego verificar la respuesta y si esta era correcta pasar al segundo nivel según estuvieran en el modelo con *scaffolding* o no. Un factor diferenciador entre los dos grupos fue la retroalimentación que recibieron de la aplicación. Para el caso del grupo control esta fue limitada, en cambio para el grupo experimental, esta retroalimentación fue de acuerdo con los errores que cometieron y el nivel de dificultad de la etapa correspondiente.

En la figura 2 se puede ver el modelo de *scaffolding* con los diferentes caminos que sigue el usuario a medida que avanza en la aplicación. Este está basado en el sistema de retroalimentación adaptativo mencionado anteriormente. El sistema hará que el usuario parta en una versión autónoma de la aplicación (*Autonomus type* en la figura), la cual no tiene retroalimentación personalizado.

Si el usuario es capaz de resolver la etapa Autónoma sin necesidad de retroalimentación, entrará a la segunda etapa en modo autónomo (sin mayor retroalimentación). Si se equivoca pasará a la etapa con retroalimentación guiada, donde recibe una mayor cantidad de instrucciones y retroalimentación referidas al error. Luego si acierta volverá a la etapa autónoma hasta pasar al segundo nivel, sino pasará a la versión con retroalimentación dirigida. En esta versión de la etapa se le indicará en un ejemplo simplificado específicamente lo que debe realizar. Las definiciones de la retroalimentación de estas etapas se pueden encontrar en extenso en la segunda etapa de esta investigación. Para fines de esta etapa se verán como diferentes niveles del software con grados de dificultad distintos.

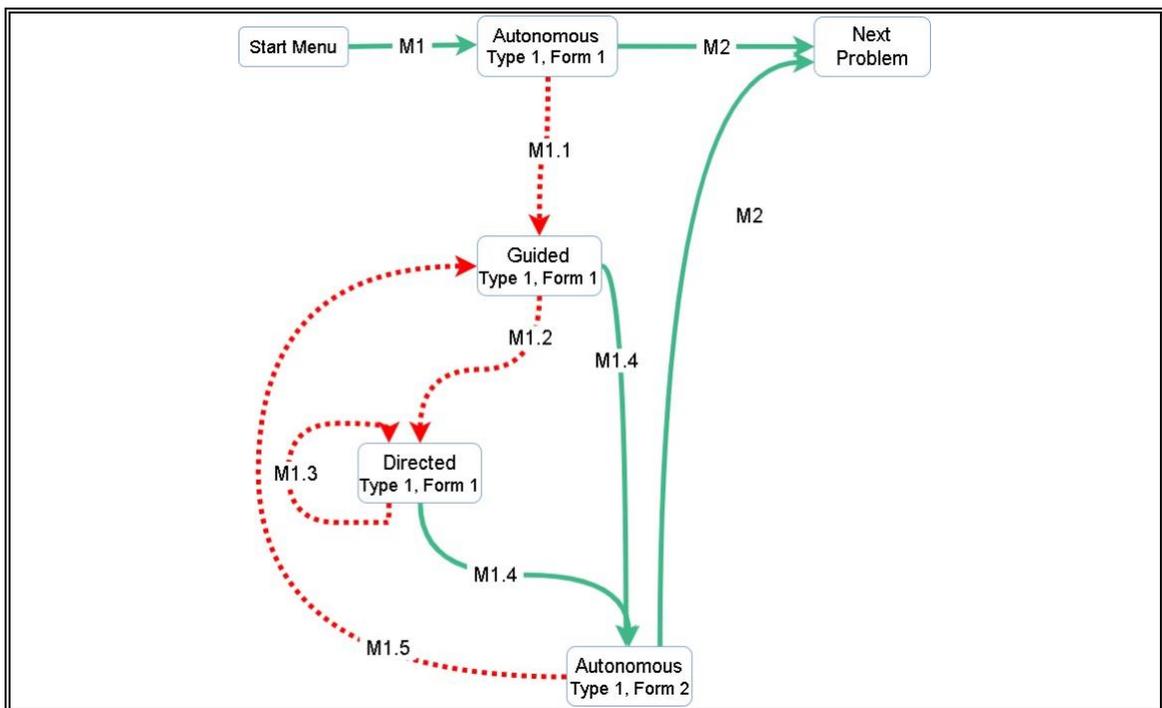


Figura 2: modelo de scaffolding

En la figura 3 se pueden ver las diferentes etapas que componen la unidad de conteo. Estas cuatro capturas de pantalla corresponden a los primeros cuatro problemas de la sesión de conteo, en diferentes versiones del dominio. La justificación de la elección de estos problemas se puede encontrar, de la misma forma, en la segunda etapa de esta investigación.



Figura 3: capturas de pantalla de la aplicación por sesiones

Por otra parte, la aplicación cuenta con una base de datos propia que registró cada acción que los usuarios cometieron en cada sesión. Esta base de datos interna de la aplicación, o log, registró los movimientos hechos por los usuarios en cada etapa y en cada nivel; los tiempos utilizados, las respuestas ingresadas y su posición en el *scaffolding* para cada acción cometida. Esta base de datos se manejó de manera independiente para cada alumno

con el fin de poder hacer seguimiento al progreso que tuvieron entre sesiones, análisis que se mostrará en la sección 4 de este capítulo.

3.3 Proceso de desarrollo

Uno de los principales desafíos planteados fue la mejora continua del diseño del software, teniendo en cuenta la metodología de enseñanza-aprendizaje detrás de la aplicación. Esto considerando los posibles sesgos que los usuarios pudieran tener en cuanto al uso del software. El proceso de diseño del software se basó en tres etapas; diseño, maqueteo y validación-testeo.

Para llevar a cabo estas etapas en primer lugar se hizo un estudio de las aplicaciones presentes hoy en el mercado con relación al *Problem Based Learning* y el aprendizaje orientado al logro. Luego se procedió a armar la base de los problemas y posteriormente se confeccionaron y aplicaron las rúbricas de usabilidad con el método heurístico y experimental explicados anteriormente. A continuación se describen las tres etapas de desarrollo.

3.3.1 Diseño de los problemas

Esta etapa comenzó con el estudio de las aplicaciones actualmente disponibles en el mercado, que tuvieran relación con la metodología de aprendizaje planteada. Uno de los factores decisivos para esta búsqueda fue el foco que debía cumplir el software en términos de habilidades a tratar en el aula. Así se desarrolló una búsqueda que tuviera énfasis en el desarrollo del pensamiento crítico mediante la indagación.

Uno de los principales instrumentos de medición y calificación en ciencias y matemáticas es la prueba de *Trends in International Mathematics and Science Study (TIMSS)*, implementada por la *International Association for the Evaluation of Educational Achievement (IEA)*. (TIMSS, 2007)

Esta prueba ha sido aplicada en más de 50 países y se caracteriza por medir contenido, rendimiento, expectativas y perspectivas sobre la población en la cual se aplica. Estas últimas habilidades son afines a los propósitos de diseño del equipo por lo que sirvieron de inspiración.

Así se buscaron problemas del área matemática, que no tuvieran una forma de solución única y generaran debate en el aula. Uno de los principales factores al momento de buscar el diseño de los problemas fue la manera en que se iba a orientar al alumno en el proceso de aprendizaje. El *scaffolding*, según la noción de Vygotsky (1978), corresponde a dar ayuda y orientar al usuario novicio por medio de mentorías o soporte brindado por parte de los usuarios más expertos. (Sawyer, 2005; P. 123).

Como lo define el autor, el *scaffolding* en software puede proveer soporte cognitivo para los alumnos situándolos en contextos de práctica más auténticos (en comparación a una clase teórica o discusión). Específicamente el software puede contribuir en cambiar los aspectos de la práctica teórica visibles y más reales. Esta visibilidad es la que potencia y fomenta la discusión entre los alumnos, la que lleva a la indagación, que es el paso buscado en el modelo de aprendizaje y que se aborda en específico en el la segunda etapa de esta investigación (Sawyer, P. 124).

Uno de los softwares con fines similares fue el presentado por la autora Catherine Chase (Chase, 2009). Su trabajo se basa en el aprendizaje por medio del Problem Based Learning orientado este al logro. Para esto construye una aplicación que considera un modelo adaptativo de *scaffolding* más simple que el que presenta esta tesis. La diferencia es que la aplicación cuenta con un avatar o personaje construido por el usuario, que lo guía durante el transcurso de las etapas. Así este personaje o *Teachable Agent* como lo

denomina el autor, es el encargado de dar retroalimentación y orientar al usuario en el logro de la tarea.

Si bien el fin pedagógico es similar al planteado por el equipo de investigación, la forma dista de la metodología planteada en esta tesis. Ya que para lograr el propósito, Chase usa como herramienta mapas conceptuales, y el testeo de la aplicación lo hace sobre grupos que ya tienen conocimientos del área y necesitan repasar o aprender habilidades nuevas dentro del mismo dominio. Los mapas conceptuales aparecen entonces como expresión de lo ya aprendido y no como motivo de aprendizaje (Vacek, 2009). Así el trabajo de Chase con los *Teachable Agents* sirvió de inspiración para orientar el diseño de la aplicación y el scaffolding en esta primera etapa, pero con un foco distinto, ya que la muestra de esta investigación debía aprender nuevos conocimientos y dominios por medio del *software* a desarrollar.

Así se utilizó la base de los problemas de la prueba estandarizada TIMMS como inspiración. Esto con el fin de obtener un set de problemas correspondiente a la variante de *Problem Based Learning*—o aprendizaje mediante problemas. Luego del diseño se procedió a implementar prototipos y testearlos en grupos control menores a la muestra real, para afinar la usabilidad de los prototipos antes de la intervención. En la siguiente sección se describe el trabajo y desarrollo de estas tareas.

3.3.2 Desarrollo de los prototipos

La etapa de desarrollo del *software* se realizó en dos partes. La primera etapa se realizó justo después de la etapa de diseño, y tenía como fin evaluar la factibilidad de creación y uso de los problemas diseñados en un entorno digital. Para esto se definieron los

problemas que cronológicamente calzaban con los contenidos del segundo ciclo de matemáticas de quinto básico y se comenzó su diseño.

En esta etapa fue crucial tomar en cuenta la orquestación de la actividad para poder generar maquetas que estuvieran alineadas con la experiencia de usuario y el uso del software. El desarrollo se hizo por medio de 3 etapas las que abarcaron: (1) describir y caracterizar al usuario, (2) identificar las áreas donde los usuarios necesitarían soporte para generar una mejor adaptación y aprendizaje en el caso de errores, y, por último, (3) desarrollar el *scaffolding* de manera conceptual y práctica para poder corregir los errores identificados en el paso (2) (Quintana et al., 2003 en Sawyer 2005). Estos tres pasos dieron como resultado el esquema mostrado en la figura 4.

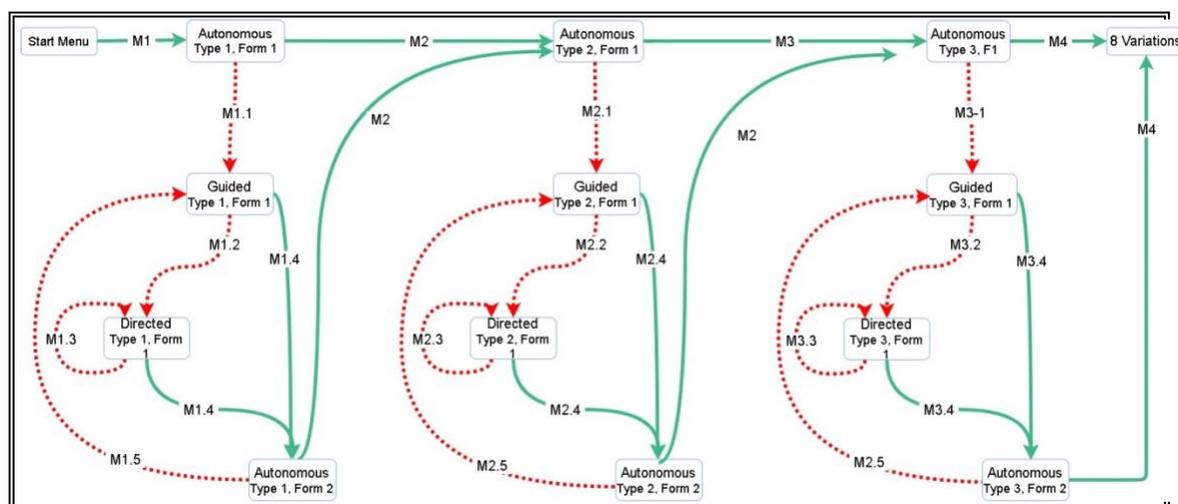


Figura 4: esquema completizado de recorrido en la aplicación

En este esquema se puede ver que, para cada clase, se definió una serie de etapas con niveles de aprendizaje para cada usuario. Estos fueron hechos para dar retroalimentación

al usuario dependiendo de los errores que cometió en cada etapa, dándole soporte basado en las necesidades de cada alumno con la intención de promover que estos logren la meta o tarea a cumplir para cada etapa en el software (Sawyer, 2005).

Parte importante de esta sección fue la definición de herramientas que los alumnos usarían para resolver los casos. Por medio de la creación de situaciones y casos de uso para cada usuario se definieron las herramientas necesarias y su aplicación para cada nivel. Estos casos de uso guardaron relación con la interacción posible que los alumnos tendrían con el software y sirvieron para caracterizar las pautas de evaluación que se explican en la sección a continuación.

Para evaluar estos diseños preliminares se usaron las pautas de evaluación conjunta de heurísticas y contexto mencionadas anteriormente. En la siguiente sección se muestra el pre-test, las pautas de observación y se detalla el proceso de implementación e intervención en detalle.

3.3.3 Aplicación de la prueba de diagnóstico

El pre-test diseñado por el equipo de investigación contenía 8 problemas similares a los que se usaron en las clases (sesiones experimentales) con ayuda del software. Estos pueden verse en el anexo 2 de esta tesis y son parte de la segunda etapa de esta investigación (Cáceres, 2017). Primero se les pidió a los estudiantes que resolvieran el problema y luego que explicaran la estrategia de resolución de problemas que adoptaron. A los estudiantes se les otorgaron dos puntos por su explicación y dos por obtener la respuesta correcta. El objetivo de la prueba era, por lo tanto, medir la capacidad de los

estudiantes para transferir los conocimientos relacionados con los temas a tratar en las sesiones experimentales y así poder determinar si la aplicación—específicamente el *user-experience*— significó o no un sesgo en el aprendizaje.

El Alfa de Cronbach (Cronbach, 1951), indicador estadístico que mide la fiabilidad de una escala de medida, se calculó para determinar la validez interna de la prueba. Como el Alfa de Cronbach depende de la muestra, se calculó para cada medida tomada. Para el grupo Control el valor fue de 0.7 y para el grupo con retroalimentación Elaborada fue 0.78. Según George & Mallery (2003), un Alfa de Cronbach de más de 0,7 indica validez interna.

Por otra parte, se utilizó la prueba de Shapiro-Wilk (Saphiro, Wilk, 1965), test estadístico que se usa para contrarrestar la normalidad de datos de un conjunto, para comprobar la distribución de los residuos en el pre-test. Los resultados para el pre-test fueron $p = 0.13$ (>0.05) para el grupo con retroalimentación Elaborada y $p = 0.0083$ (<0.05) para el grupo Control. Esto sugiere que los resultados deben interpretarse con cautela.

La igualdad de varianzas fue verificada usando la prueba de Bartlett, test estadístico utilizado para saber si las muestras vienen de poblaciones con la misma varianza. El resultado de esta prueba fue de 0.96 (>0.05), lo que indica que las variaciones son homogéneas. La falta de interacción entre la co-variable (es decir, el resultado en la pre-prueba) y el grupo del estudiante se verificó usando un ANOVA, con el tipo de retroalimentación como variable independiente y la co-variable como término de interacción.

El término de interacción devolvió $F(1,60) = 2,57$, $p = 0,347$ ($>. 05$), es decir, no significativo. Por lo tanto, se puede descartar cualquier interacción entre el resultado del

test previo y el grupo de estudiantes. En las siguientes figuras se detallan los resultados del pre test y los análisis para ambos grupos. El análisis en extenso de esta prueba y el detalle de la construcción tanto del pre test como del post test se abarcan en la segunda etapa de esta investigación.

Tabla 1: Descripción de los valores para el pre-test

| Grupo | n | Pre-test (SD) |
|--------------------------------|----|------------------|
| Grupo Control | 33 | 17.84 (4.54) |
| Retroalimentación Elaborado | 31 | 18.87 (4.97) |

*p<0.05

Tabla 2: ANOVA

| | Suma de cuadrados | df | Media cuadrática | F | p |
|----------|----------------------|----|---------------------|-------|------------------------|
| Pre-test | 427.5 | 1 | 427.5 | 34.66 | $1.8 \cdot 10^{-7}$ ** |
| Grupo | 52.5 | 1 | 52.5 | 4.257 | 0.0434 * |
| Residuos | 752.2 | 61 | 12.3 | | |

* p<0.05

** p>0.001

3.3.4 Implementación y testeo mediante evaluación conjunta

El proceso de desarrollo constó de dos etapas principalmente, la primera fue la creación de la aplicación con los testeos de usabilidad de los prototipos. La segunda etapa fue la de implementación de la aplicación en el Colegio Terra Australis, específicamente en los cursos de quinto básico del establecimiento. Estos contaron con un total de 64 alumnos (n=31 y n =33, para quinto A y C respectivamente). Para esto se utilizó la rúbrica de

evaluación conjunta. Este testeo se llevó a cabo con un grupo de teteo piloto del software; lo que sirvió para obtener algunas observaciones de usabilidad y hacer ajustes a la rúbrica de evaluación.

En la figura 7 se muestran imágenes de estos primeros prototipos y de las observaciones hechas en la rúbrica. Esta permitió evaluar los tres aspectos de usabilidad mencionados anteriormente (eficiencia, eficacia y satisfacción) y ajustar los prototipos según los criterios. Posteriormente se usaron las observaciones de la rúbrica y los datos recolectados para estudiar el avance en términos de usabilidad a lo largo del uso del software.

La segunda parte del proceso de diseño fue la incorporación de las correcciones a estos prototipos gracias a las observaciones de la rúbrica de evaluación. En esta segunda etapa se perfeccionó de manera iterativa la aplicación, y para cada sesión se aplicaron los criterios de evaluación de usabilidad antes descrito para así mejorar las sesiones posteriores.

| D | nombre | finalidad | prioridad | Workflow | Problema | Heurística as problema en la interfaz |
|----------|---|-----------|-----------|----------|----------|---------------------------------------|
| 0 | entender el desafío | | | | | |
| 0.1 | entender las instrucciones | | | | | |
| 0.2 | entender el feedback | | | | | |
| 0.3 | entender y usar la interfaz | | | | | |
| 0.4 | entender y completar el ingreso de respuestas | | | | | |
| 1 | completar nivel dirigido | | | | | |
| 1.1 | completar el tutorial | | | | | |
| 1.2 | entender las instrucciones | | | | | |
| 1.3 | realizar las acciones que se dictan | | | | | |
| 1.4 | ingresar correctamente el resultado | | | | | |
| 1.5 | ingresar correctamente la estrategia | | | | | |
| 2 | completar nivel guiado | | | | | |
| 2.1 | completar el tutorial | | | | | |
| 2.2 | entender las instrucciones | | | | | |
| 2.3 | entender la interfaz | | | | | |

Figura 5: pauta conjunta de evaluación

| D | nombre | finalidad | prioridad | Workflow | Problema | Heurística asociada | problema en la interfaz | Trabajo a realizar |
|-----|---|---|-----------|---|--|--|--|--|
| 0 | entender el desafío | | | | | | | |
| 0.1 | entender las instrucciones | Poder desarrollar los desafíos de la manera adecuada mediante la lectura de las instrucciones | | 5 Ver la pantalla del desafío, encontrar el botón de ayuda, presionarlo, leer las instrucciones, llevar a cabo el desafío | Las instrucciones no siempre son reconocidas | Velocidad de la lectura y entendimiento de las instrucciones | Hay que desplegarlas | Mostrar las instrucciones al abrir la etapa |
| 0.1 | entender el feedback | Poder entender los errores cometidos mediante la retroalimentación que otorga el sistema | | 3 Leer la notificación de respuesta incorrecta, leer el error asociado, leer el correcto funcionamiento | El feedback no se muestra | Cantidad de errores enmendados gracias al feedback | el feedback no se muestra | mostrar el feedback asociado al error que se cometió |
| 0.2 | entender y usar la interfaz | Poder desarrollar la actividad mediante la interfaz propuesta de manera ágil | | 4 Presionar los botones de la interfaz, realizar la acción asociada al botón, llegar al resultado esperado de dicho botón | No se hace una introducción a la interfaz | Velocidad del comienzo del uso de la interfaz | Algunos de los movimientos o elementos no son naturales para los niños | Hacer un tutorial o pantalla de bienvenida de la aplicación explicando la interfaz antes de empezar (puede ser optativo tomar el tutorial para el usuario) |
| 0.3 | entender y completar el ingreso de respuestas | Poder ingresar la respuesta al problema planteado | | 5 Buscar la sección de respuesta, presionarla, ingresar la respuesta, validarla | | Cantidad de clicks erróneos antes de poder ingresar la respuesta | | |
| 0.4 | | | | | | | | |

Figura 6: caso de uso de pauta conjunta de evaluación

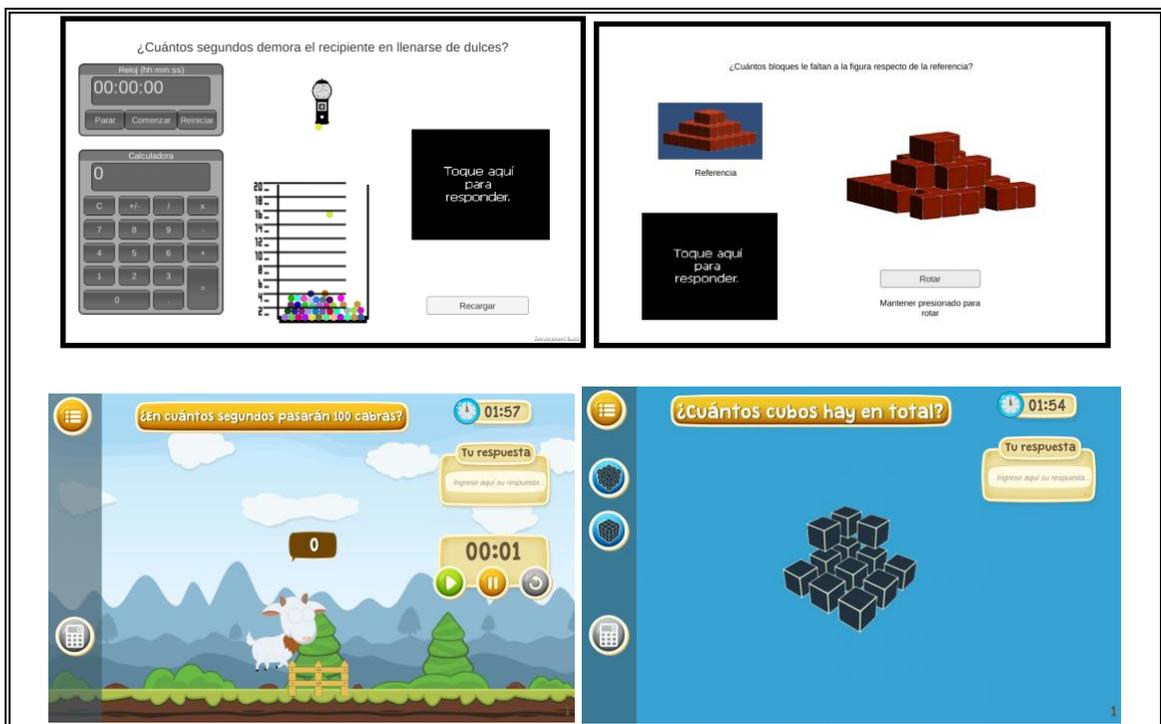


Figura 7: prototipos antes y después de la aplicación de la rubrica

En primera instancia se definió, para los problemas de conteo, los principales métodos que podrían usarse para lograr la meta del problema propuesto. Para esto se usaron las pautas que pueden observarse en la figura 5 y 6. En estas se buscaron posibles aplicaciones y se definieron las posibles acciones para cada tipo de usuario según su nivel de *expertise*. Por ejemplo, para el caso particular de los problemas de conteo, la primera observación que se realizó fue en cuanto al sistema de medida de distancias que se tenía. En un principio este se implementó en el software como el sistema de medida que se aplica en los mapas modernos (donde se selecciona un punto de inicio y uno de llegada y el software muestra el largo de la línea trazada).

Luego de la creación del caso de uso se determinó que este método no era intuitivo y que su diseño no facilitaba el uso de la herramienta. Así los usuarios expertos, o con menor adversidad a la tecnología en este caso, reportaron menor tasa de fallo en el uso del software (realizaron acciones que se alinean en pos de cumplir con el propósito del workflow), versus los usuarios novicios. Información que se pudo obtener de las observaciones hechas en las pautas de evaluación mencionadas anteriormente.

La observación de la interacción del usuario sugirió un paradigma más cercano a la realidad del aula, y se determinó la inclusión de reglas gráficas en la aplicación, lo que marco un menor porcentaje de error y un mejor recibimiento en el aula. A cada observación se le asignó una prioridad con valor entre 1 y 5, de menos importante para el workflow—o procedimiento a realizar por el alumno—a mas importante. Esto sirvió para caracterizar las observaciones de satisfacción que se discuten en la sección de exposición y análisis de resultados. En la figura 8 se puede ver la maqueta inicial versus la maqueta con la implementación de reglas del ejemplo mencionado.

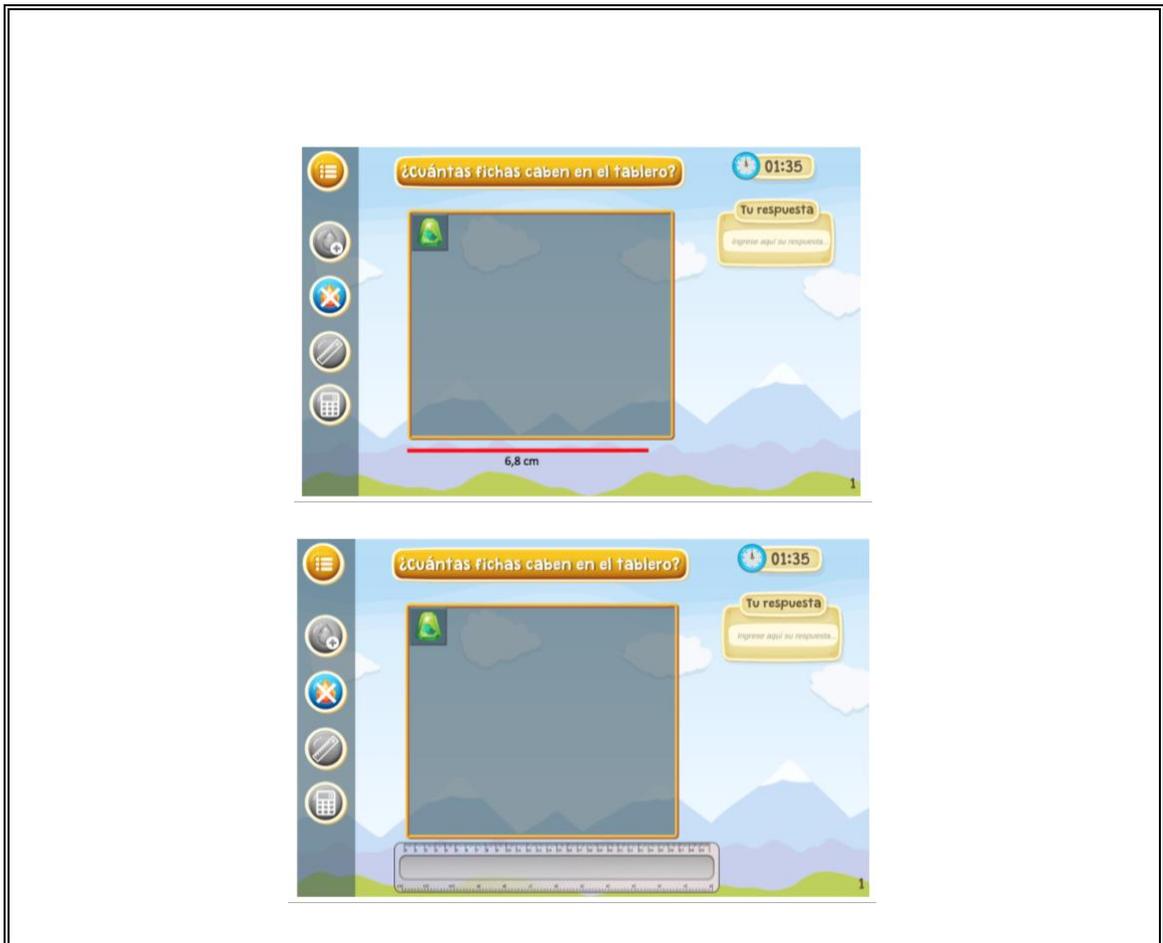


Figura 8: comparación de maquetas antes y después de la evaluación

4. RESULTADOS Y DISCUSIÓN

4.1 Efectividad de la herramienta

Para analizar el impacto de la intervención en los alumnos, como se mencionó anteriormente, se aplicó un pre-test en los estudiantes del grupo piloto. Estos individuos no presentaban ningún conocimiento previo de la herramienta ni de los contenidos que esta evalúa. El análisis de estos resultados demostró que la intervención fue más efectiva en el grupo que presentaba el modelo de *scaffolding*; lo anterior sugiere que, al menos a corto plazo, el sistema de retroalimentación que fue diseñado permitió a los estudiantes ser más eficientes en la resolución de los problemas. Los detalles de este punto de la intervención pueden verse en la segunda etapa de esta investigación.

Por otro lado, los resultados expuestos del pre-test, según lo mostrado en la sección 3.3.3, señalan que se puede descartar cualquier interacción entre el resultado del test previo y el grupo de estudiantes. Así para efectos de los análisis de esta sección se puede descartar cualquier interferencia de contenido o fondo con los resultados de desempeño expuestos, ya que los alumnos tienen manejo en el dominio que se les planteó el problema. Detalles de los análisis del proceso de enseñanza-aprendizaje pueden encontrarse en la segunda etapa de esta investigación, en esta etapa se discutirán los resultados que corresponden a la usabilidad del software.

4.2 Estudio de usabilidad

Autores dedicados a los estudios de usabilidad, como Bevan, han categorizado las componentes de usabilidad antes mencionadas (Nielsen y Wharton) para términos de

análisis y estudio. Según el autor se tienen tres dimensiones importantes (Bevan, 2006), acordes a las ya descritas en el sistema propuesto en las secciones anteriores:

- Eficiencia: "recursos gastados" la rapidez con la que un usuario puede realizar el trabajo es fundamental para la productividad empresarial.
- Satisfacción: grado de satisfacción de las expectativas. La satisfacción es un factor de éxito para cualquier producto con uso discrecional; es esencial para mantener la motivación de los usuarios.
- Eficacia: "precisión e integridad" la realización de tareas sin errores es importante tanto en las aplicaciones comerciales como de consumo. Se puede decir que la eficacia de un producto depende de la precisión con la que se lleven a cabo las tareas y se alcancen los objetivos para los que está diseñado.

Para poder hacer estimaciones fiables de los resultados de satisfacción, son necesarios al menos ocho o diez participantes; las muestras más grandes ofrecen un valor más significativo de la tasa de éxito (Bevan, 2006). En este estudio, la evaluación ha sido realizada por todos los estudiantes presentes en la fase experimental del curso; correspondientes a un total de 78 alumnos.

Para medir eficiencia se compararon los tiempos medios de uso por cada etapa, por cada sesión, medida cuantificada por el log de la aplicación en segundos. Los resultados de estas y la evolución por sesión se puede ver en la figura 9 presentada a continuación. Esta muestra el tiempo promedio de resolución de etapas por sesión, por grupo. Estos grupos se construyeron y analizaron de esta forma para guardar relación con lo expuesto en la

segunda parte de esta tesis (validación del esquema de enseñanza-aprendizaje con el logro).

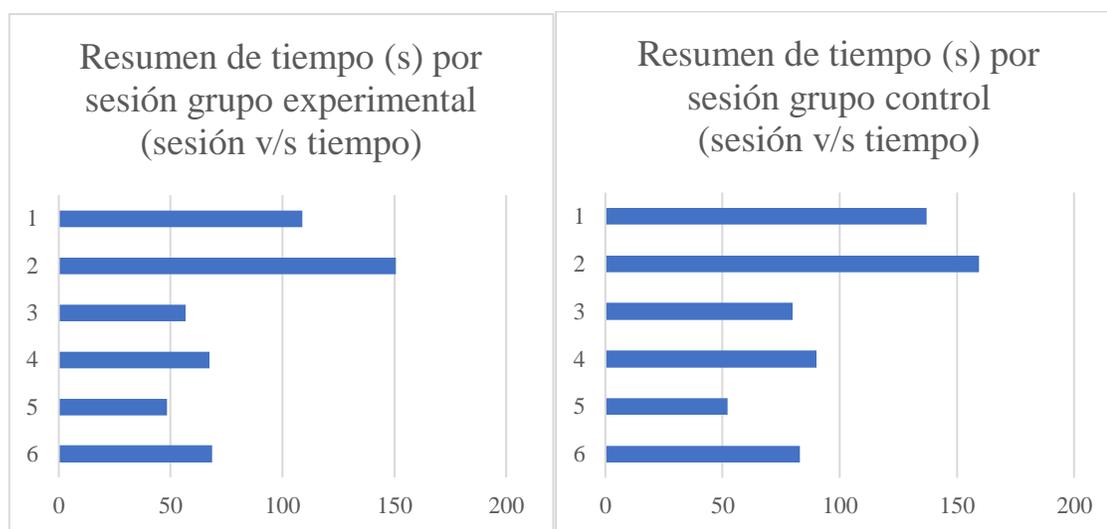


Figura 9: Tablas de resumen de tiempo por grupo en segundos

Para contrarrestar estas medidas con la satisfacción que presentaron los usuarios, se llevó a cabo un recuento de los registros en las pautas de observación conjuntas. Se puso énfasis en registrar el número de observaciones específicas que tuvieron los usuarios durante el uso de la interfaz ligadas a heurísticas de satisfacción que determinaron cambios o problemas en el uso de la aplicación (prioridad 3 o mayor) según las observaciones hechas durante la intervención (Hornbæk, 2006). Estas se compararon con el total de observaciones de satisfacción asociada a cada una de las dimensiones para obtener un porcentaje de cada muestra.

El resumen de esta observación se puede ver en la figura 10 donde el porcentaje azul es el resultado de observaciones que llevaron a un cambio en cada dimensión. Se puede ver que en su mayoría los cambios fueron relacionados a la estética y diseño de la aplicación

seguido del reconocimiento de errores, lo que está derechamente ligado con la retroalimentación. Estas fueron consideradas según la prioridad asignada y solo se tomaron en cuenta para los cambios las observaciones de mayor prioridad (mayor a 3). Así, por ejemplo, para la dimensión de diagnóstico, se tuvo que solo un 30% de las observaciones ligadas a esa categoría desembocaron en un cambio efectivo en la interfaz.

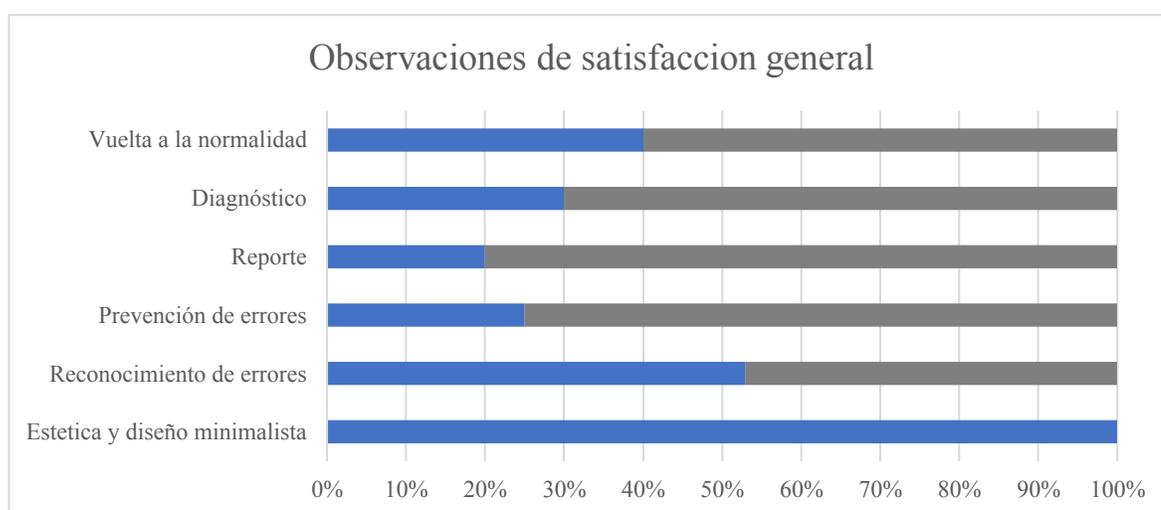


Figura 10: Resumen preguntas de satisfacción

Un punto importante al momento de evaluar la satisfacción y desempeño que presentaron los usuarios con la aplicación fue el uso que le dieron a esta. El registro que la aplicación guardó de cada usuario permitió hacer un mapeo de las acciones que los sujetos cometieron particularmente en cada sesión. Específicamente se llevó a cabo un análisis de las acciones registradas por el *back-end* de la aplicación para determinar qué usuarios le dieron un correcto uso a la herramienta.

Para llevar a cabo este análisis se procedió primero a generar los *workflows* o grupo de acciones que el usuario debería realizar para completar la tarea asignada en la etapa

asignada. Estos fueron particulares para cada etapa y se validaron durante la intervención –se comprobó que los sujetos que seguían estos procedimientos realizaban la tarea de manera correcta. Esto se ve directamente relacionado con la componente de eficacia ya que un correcto uso de estas herramientas lleva a al alcance del objetivo del uso del software en cada sesión.

Por otro lado, aplicación guardó las acciones cometidas por los usuarios en una base de datos particular para cada Tablet. Este registro no solo almacenó los tiempos y cantidad de respuestas correctas / incorrectas, sino que también las acciones realizadas en la interfaz. Estas acciones fueron el movimiento de ficha y reglas (para la sesión de conteo), agrupación de cardúmenes de peces, acciones cometidas con el cronómetro u otras herramientas, pausas y tiempos de pausa por cierre de la aplicación, entre otras propias para cada sesión.

Los registros particulares de cada Tablet se unificaron en una gran base de datos donde se le dio seguimiento a cada niño durante las 6 sesiones. El registro o *log* sirvió para categorizar las acciones realizadas por los alumnos en la aplicación cualitativamente. En la figura 12 se puede ver una captura de pantalla de un log categorizado. En este se puede apreciar el recuadro con el detalle de las acciones, y a la derecha un resumen de estas.

Posteriormente se procedió a observar y resumir el número de acciones promedio de cada etapa y en base a eso hacer un conteo de los sujetos que cometieron acciones que no contribuían al workflow y aumentaban su tiempo de uso de la aplicación de manera redundante. En la figura 11 se puede ver el caso de un log con movimientos innecesarios de la regla. Esta caracterización comparó el número y los tiempos de acciones promedio

para el grupo y buscó identificar los números y acciones *outlayers* en base al promedio de la muestra; la Figura 12 muestra un resumen porcentual de los casos encontrados.

| | A | B | C | D | E | F | G |
|----|-----------------|-----------------|--------------------|-------|-------|-----------------------------|------------------------|
| 1 | Sesion de Juego | Numero de Juego | Accion | Dato1 | Dato2 | Tiempo actual del ejercicio | Tiempo actual completo |
| 2 | *** | *** | *** | *** | *** | *** | *** |
| 3 | 6 | 1 | ComenzarEjercicio | | | 0:00 | 0:06 |
| 4 | 6 | 1 | PonerFicha | 10 | 8 | 0:01 | 0:07 |
| 5 | 6 | 1 | BotonEliminarFicha | | | 0:04 | 0:10 |
| 6 | 6 | 1 | EliminarFicha | 10 | 8 | 0:04 | 0:10 |
| 7 | 6 | 1 | BotonPonerFicha | | | 0:05 | 0:11 |
| 8 | 6 | 1 | PonerFicha | 8 | 8 | 0:52 | 0:58 |
| 9 | 6 | 1 | PonerFicha | 10 | 7 | 0:55 | 1:01 |
| 10 | 6 | 1 | PonerFicha | 10 | 6 | 0:56 | 1:02 |
| 11 | 6 | 1 | PonerFicha | 10 | 8 | 1:05 | 1:11 |
| 12 | 6 | 1 | BotonEliminarFicha | | | 1:12 | 1:18 |
| 13 | 6 | 1 | EliminarFicha | 10 | 8 | 1:13 | 1:19 |
| 14 | 6 | 1 | EliminarFicha | 10 | 7 | 1:13 | 1:19 |
| 15 | 6 | 1 | EliminarFicha | 10 | 6 | 1:14 | 1:20 |
| 16 | 6 | 1 | EliminarFicha | 8 | 8 | 1:14 | 1:20 |
| 17 | 6 | 1 | BotonPonerFicha | | | 1:15 | 1:21 |
| 18 | 6 | 1 | PonerFicha | 10 | 1 | 1:19 | 1:25 |
| 19 | 6 | 1 | PonerFicha | 10 | 2 | 1:20 | 1:26 |
| 20 | 6 | 1 | PonerFicha | 10 | 3 | 1:21 | 1:27 |
| 21 | 6 | 1 | PonerFicha | 9 | 3 | 1:23 | 1:29 |
| 22 | 6 | 1 | PonerFicha | 9 | 1 | 1:24 | 1:30 |
| 23 | 6 | 1 | PonerFicha | 3 | 1 | 1:39 | 1:45 |
| 24 | 6 | 1 | PonerFicha | 2 | 1 | 1:40 | 1:46 |

Figura 11: Captura del registro generado por la aplicación

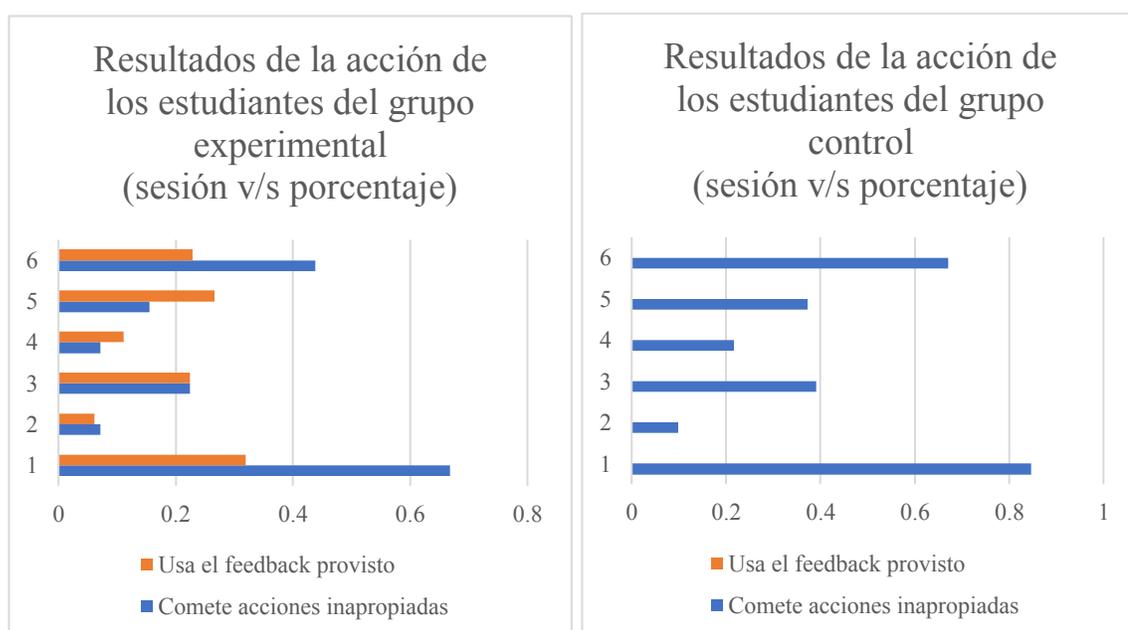


Figura 12: Resumen acciones grupo control y experimental

De la Figura 12 se puede ver que el porcentaje de acciones inapropiadas para el grupo experimental fue menor a las del grupo control en todas las sesiones, resultado que se discute en la sección de análisis.

Según las observaciones hechas, el material y los contenidos del curso fueron bien recibidos por los grupos de testeo. El material entregado estuvo bien estructurado y no presentó problemas de diseño y estética para el usuario; se presentó bien y con cuidado, la aplicación fue estable y en su mayoría representa, diagnosticó y corrigió los errores presentados.

El número de ejercicios realizados en cada sesión se consideró suficiente y guarda relación con los tiempos de realización en cada sesión. También se consideró que el material y los contenidos son eficientes, porque los sujetos tuvieron tiempo de terminar la obra en el tiempo fijado sin dificultad, y este mejoró en base a los cambios propuestos de usabilidad; salvo la última sesión que presentó problemas de dificultad en términos de contenidos para los alumnos, que se vio reflejado en los resultados del pre y post test que se analiza en la segunda etapa de esta investigación.

4.3 Conclusiones

En Chile existe poca evidencia sobre el impacto que tienen las herramientas tecnológicas de aprendizaje sobre el logro que alcanzan los usuarios mediante la utilización de estas. El objetivo del presente estudio fue determinar un método de implementación que disminuya el impacto que tiene la adaptación de los estudiantes a la tecnología, en mediciones hechas sobre el logro. Esto se hizo mediante el desarrollo e implementación de una aplicación con dos variantes: Retroalimentación elaborada (Grupo Experimental) y Conocimiento de Resultados (Grupo Control). La implementación de esta aplicación se llevó a cabo mediante un método elaborado de medición de usabilidad, recopilando datos y observaciones en todas las dimensiones de esta.

Se diseñó una intervención, la que incluyó un pre-test y pautas de observación de usabilidad. Después de seis sesiones en el aula, repartidas en un período de seis semanas, se pudieron recoger resultados tanto cuantitativos como cualitativos de uso de la aplicación. Estos instrumentos permitieron adaptar la aplicación a las necesidades de los alumnos de manera dinámica durante el desarrollo e implementación del software, y recoger datos que sirvieron para respaldar las hipótesis planteadas.

Específicamente se desarrollaron pautas de diseño de usabilidad que ayudaron en la metodología de implementación de la aplicación. Estas pautas sirvieron para determinar los cambios de diseño en base a las heurísticas y workflows planteados. Estos resultados se compararon con los tiempos de ejecución, observaciones de satisfacción y con la caracterización de las acciones que los alumnos realizaron en la intervención. Esto sirvió

para responder al objetivo de esta tesis que fue cuantificar el impacto que tienen los cambios de diseño sobre el logro; mediante análisis del User Experience.

Bajo este modelo se observó una disminución del tiempo que los usuarios estuvieron en la aplicación—Figura 9; manteniendo un nivel de dificultad constante. Según la metodología de desarrollo planteada una de las posibles causas fueron las observaciones y cambios de diseño planteados por el equipo en base a las pautas y su análisis.

Más aún, se puede ver que porcentualmente los usuarios del grupo experimental obtuvieron mejores resultados en cuanto a tiempo. Gracias a la iteración en el diseño y la mejora de la usabilidad, el porcentaje de acciones inapropiadas que los alumnos realizaron disminuyó en el tiempo, un resumen de esto se puede ver en la Figura 13.

Porcentualmente los sujetos sobre los que se aplicó la versión con retroalimentación cometieron menos acciones inapropiadas; es decir, le dieron un mejor uso a la aplicación en cuanto al objetivo que fue llevar a cabo la tarea (relacionadas al workflow de realización del objetivo). Esto pudo deberse a que la versión del grupo experimental contó con feedback, el cual al momento de detectar el error o respuesta incorrecta del alumno lo diagnosticó y le dio soporte para orientarlo en cuanto al objetivo a realizar en la tarea.

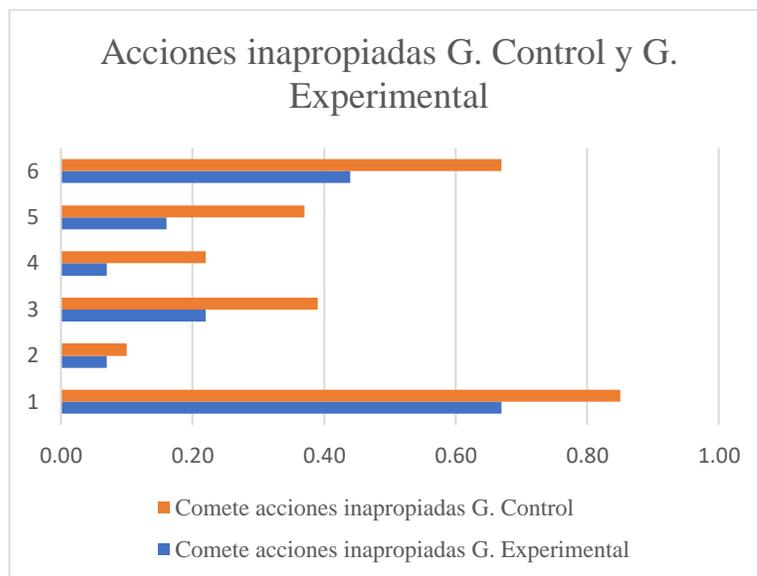


Figura 13: Porcentajes de acciones inapropiadas por grupo

En términos de usabilidad, se puede señalar que en las dimensiones de eficiencia, eficacia y satisfacción el grupo que obtuvo mejores resultados fue el grupo experimental. Porcentualmente estos demostraron desarrollar las tareas asignadas de manera más eficiente y rápida (en cuanto a acciones inapropiadas y tiempos de ejecución). Esto respalda la hipótesis de que la usabilidad mejora los resultados obtenidos, ya que estos alumnos contaron con más recursos ligados a las componentes de satisfacción, específicamente a las heurísticas ligadas al diagnóstico de respuestas incorrectas. Esto último respalda la componente de logro de la hipótesis planteada pero aún no se puede señalar una relación específica con el aprendizaje. Específicamente la componente de aprendizaje de esta hipótesis se abarca en la segunda etapa de esta investigación.

En base a las observaciones y datos obtenidos, se puede señalar que, en las aplicaciones educativas, es de suma importancia centrar la atención de los estudiantes en la tarea real

y reducir la carga cognitiva necesaria para usar la aplicación. Esto es parte de la motivación para diseñar un sistema amigable para el usuario.

El software implementado demostró ser eficiente en el desarrollo de habilidades para la resolución de problemas matemáticos de las unidades planteadas y así lo demuestran los tiempos de respuesta obtenidos por los grupos mostrados en los recuadros de la figura 9.

Para evaluar la componente de satisfacción se usaron las pautas de observación propuestas. Estas se aplicaron para categorizar las observaciones en base a las dimensiones de satisfacción e interacción de que tuvieron los usuarios con la aplicación. Tanto el material como la tecnología utilizada demostró ser aceptada y romper las limitantes en cuanto al logro que pudieran generar en los alumnos. El software demostró ser robusto, ya que no se detectaron errores significativos durante su uso. Como se detalla en la figura 11, los cambios más relevantes se llevaron a cabo en el ajuste del diseño y la entrega de retroalimentación al usuario. Las otras dimensiones de satisfacción presentaron observaciones poco significativas en cuanto a la prioridad asignada, por lo que se puede inferir que tuvo una buena recepción en los grupos aplicados.

Este resultado se complementa con la evidencia encontrada por Martin-Gutiérrez (2010) quien comprobó que los procedimientos de aprendizaje y enseñanza tienen que evolucionar para tener en cuenta el alto perfil tecnológico y lógico que la mayoría de los estudiantes muestran (Martin-Gutierrez, 2010). En algunos casos, los métodos de enseñanza anticuados crean una barrera para algunos estudiantes que están acostumbrados a interactuar con los gadgets y computadoras lógicas modernos. Es por esto por lo que, en

las aplicaciones educativas, es de suma importancia enfocar la atención de los estudiantes en la tarea real y reducir la sobre-cabeza cognitiva necesaria para usar la aplicación.

Una limitante no menor del análisis propuesto fue la no diferenciación en el análisis de satisfacción según grupo de aplicación. Para futuras versiones de estudios de usabilidad en este contexto sería interesante poder ver como la satisfacción influye los resultados de logro y eficacia. Específicamente se podrían generar clústeres y caracterizar el logro que tienen los distintos grupos en base a sus propios paradigmas y contrarrestar esto con el análisis propuesto.

Es posible concluir que existe un impacto en el logro que obtuvieron los alumnos en la aplicación y el grado de usabilidad que esta presenta en los contextos señalados. Esto se puede aseverar en base a las dimensiones y caracterizaciones hechas: los análisis cuantitativos de logro (específicamente en los tiempos de ejecución) y los análisis cualitativos de usabilidad (porcentaje de acciones inapropiadas y caracterización de las componentes de satisfacción).

Un análisis que no se consideró y pudo ser interesante de ver en esta sección, fue el impacto en el porcentaje de respuestas correctas versus incorrectas logradas por los alumnos en el software. Cabe destacar que, pese a que se tuvieron los datos de los números de respuestas para cada alumno en cada sesión, estos no son suficientes para determinar si la componente de eficacia está relacionada con estos. Esto debido a que el scaffolding si influencia en el logro que presentan los alumnos en el software y a que este debe contrarrestarse en algún punto con el mismo rendimiento en el dominio sin el software.

Se puede concluir que uno de los principales factores que influyen en la inclusión de tecnologías en el aula es el sesgo de adaptación que tienen tanto los profesores como los alumnos. Entonces, una componente crucial para la correcta integración de tecnología en el aula son las metodologías de integración y medición de usabilidad. Estas permiten disminuir el impacto que la adaptación genera en la eficiencia y satisfacción del software, y es una manera objetiva de caracterizar la integración de la tecnología en cuanto a los modelos de enseñanza-aprendizaje. Esto último puede influir positivamente los resultados.

BIBLIOGRAFIA

- Anderson, J. (2010). *ICT transforming education: A regional guide*. Bangkok, TA: UNESCO.
- Benito, M. Desafíos pedagógicos de la escuela virtual. Las TIC y los nuevos paradigmas educativos. *TELOS cuadernos de Comunicación en Innovación*. 78. 2009.
- Bergen, A., French, L., & Hawkins, L. (2015). *Teaching and learning in a digital world: A developmental evaluation of virtual learning environments in the Upper Grand and York Region district school boards*.
- Bevan, N. (2006). Practical issues in usability measurement, *interactions*, v. 13 n. 6. November+ December.
- Biswas, G., Leelawong, K., Schwartz, D., Vye, N., & The Teachable Agents Group at Vanderbilt. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, 19(3-4), 363-392.
- Cáceres, M. (Diciembre, 2017). ¿Es mejor una retroalimentación más detallada para la resolución de problemas?. *Computers & Education*.
- Chase, C. C., Chin, D. B., Opezzo, M. A., & Schwartz, D. L. (2009). Teachable agents and the protégé effect: Increasing the effort towards learning. *Journal of Science Education and Technology*, 18(4), 334-352.
- Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3), 297-334.
- Dix, A. (2009). Human-computer interaction. In *Encyclopedia of database systems* (pp. 1327-1331). Springer US.
- Dwyer, C. P., Hogan, M. J., & Stewart, I. (2014). An integrated critical thinking framework for the 21st century. *Thinking Skills and Creativity*, 12, 43-52.
- Franzke, M. (1995, May). Turning research into practice: Characteristics of display-based interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 421-428). ACM Press/Addison-Wesley Publishing Co..
- George, D., & Mallery, P. (2003). *SPSS for Windows step by step: a simple guide and reference*. Boston, MA: Allyn and Bacon.

- Gonzalez, F. (2017, Marzo). Diseño e implementación de videojuegos matemáticos para desarrollar el pensamiento crítico en clases de quinto año básico. Santiago, Chile: Pontificia Universidad Católica de Chile.
- Grünbaum, L., Pederson, M., & Nielsen, S. (2004). Study on innovative learning environments in school education. Final report for the European Commission DG Education and Culture: Rambold Management.
- Hargreaves, A., & Fink, D. (2012). Sustainable leadership (Vol. 6). John Wiley & Sons.
- HOLLINGSWED, Tasha; NOVICK, David G. Usability inspection methods after 15 years of research and practice. En Proceedings of the 25th annual ACM international conference on Design of communication. ACM, 2007. p. 249-255.
- Hornbæk, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *International journal of human-computer studies*, 64(2), 79-102.
- Ilabaca, J. S. (2003). Integración curricular de TIC concepto y modelos. *Revista enfoques educacionales*, 5(1), 01-15.
- ISO, I. (1998). TC 159/SC4/WG3 N147: Ergonomic requirements for office work with visual display terminals (VDTs)-Part 9-Requirements for non-keyboard input devices. International Organisation for Standardisation.
- Janssen, F., Westbroek, H., Doyle, W., & Van Driel, J. (2013). How to make innovations practical. *Teachers College Record*, 115(7).
- Ledesma, L. (2006). Learning in a digital world-Learning through ICT in New Zealand. Retrieved January 1, 2009.
- Levy, F., & Murnane, R. J. (2012). The new division of labor: How computers are creating the next job market. Princeton University Press.
- Martín-Gutiérrez, J., Contero, M., & Alcañiz, M. (2010). Evaluating the usability of an augmented reality based educational application. In *Intelligent tutoring systems* (pp. 296-306). Springer Berlin/Heidelberg.
- Moreno, L. B., & Martín, R. F. P. (2016). Análisis de la implementación de Flipped Classroom en las asignaturas instrumentales de 4º Educación Secundaria Obligatoria. *EduTec. Revista Electrónica de Tecnología Educativa*, (55).

- Moss, P. (2006). The New Division of Labor: How Computers are Creating the Next Job Market. *Contemporary Sociology: A Journal of Reviews*, 35(4), 383-384.
- Oblinger, D., Oblinger, J. L., & Lippincott, J. K. (2005). Educating the net generation. Boulder, Colo.: EDUCAUSE, c2005. 1 v.(various pagings): illustrations..
- Pedró, F. (2009, September). New Millennium learners in higher education: Evidence and policy implications. In *International Conference on 21st Century Competencies*, Brussels: OECD/CERI.
- Polson, P. G., Lewis, C., Rieman, J., & Wharton, C. (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies*, 36(5), 741-773.
- Prensky, M. (2001). Digital natives, digital immigrants part 1. *On the horizon*, 9(5), 1-6.
- Reimers, F. M., & Chung, C. K. (2016). *Teaching and Learning for the Twenty-First Century: Educational Goals, Policies, and Curricula from Six Nations*. Harvard Education Press. 8 Story Street First Floor, Cambridge, MA 02138.
- Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4), 591-611.
- Sasse, M. A., & Johnson, C. (1999). *Human-computer Interaction, INTERACT'99: IFIP TC. 13 International Conference on Human-Computer Interaction, 30th August-3rd September 1999, Edinburgh, UK (Vol. 1)*. IOS Press.
- Sawyer, R. K. (Ed.). (2005). *The Cambridge handbook of the learning sciences*. Cambridge University Press.
- Shirk, H. N. (1996). Usability Inspection Methods, edited by Jakob Nielsen and Robert L. Mack. *JOURNAL OF TECHNICAL WRITING AND COMMUNICATION*, 26, 97-97.
- TIMSS. (2017). "About TIMSS 2015". In: [Timssandpirls.bc.edu](http://timssandpirls.bc.edu). Retrieved 5 May 2017, from <http://timssandpirls.bc.edu/timss2015/international-results/timss-2015/about-timss-2015/>
- Trilling, B., & Fadel, C. (2009). *21st century skills: Learning for life in our times*. John Wiley & Sons.

- Vacek, J. E. (2009). Using a conceptual approach with a concept map of psychosis as an exemplar to promote critical thinking. *Journal of Nursing Education*, 48(1), 49-53.
- Veletsianos, G. (Ed.). (2016). *Emergence and innovation in digital learning: Foundations and applications*. Athabasca University Press.
- Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994, June). The cognitive walkthrough method: A practitioner's guide. In *Usability inspection methods* (pp. 105-140). John Wiley & Sons, Inc..
- Wood, D. F. (2003). ABC of learning and teaching in medicine: Problem based learning. *BMJ: British Medical Journal*, 326(7384), 328.
- Zabalza, M. Á., & Beraza, M. Á. Z. (1987). *Diseño y desarrollo curricular* (Vol. 45). Narcea Ediciones.

ANEXOS

ANEXO 1 : ESTRUCTURA Y MOTOR DE LA APLICACIÓN (GONZALEZ, 2017)

1.1.1 Arquitectura.

La arquitectura de software de excelencia en los desarrollos con elementos visuales, tal como es el desarrollo de videojuegos, es la arquitectura MVC (modelo, vista, controlador) (Flynt, Salem, & Zerbst, 2004). En nuestro juego estará representada de la siguiente manera:

1. Vista: las escenas de juego son creadas por medio del editor visual, pudiendo manipular en un comienzo cada elemento visual por separado, principalmente: la posición, rotación, escala y el polígono de colisión.
2. Controlador: por medio de scripts se crean funciones que se ejecutan al cumplirse un evento en el juego. Cada elemento de la vista puede estar asociado a uno o más scripts.
3. Modelo: los datos están almacenado en cada elemento del juego y pueden ser manipulados en un comienzo mediante el *Inspector* de Unity.

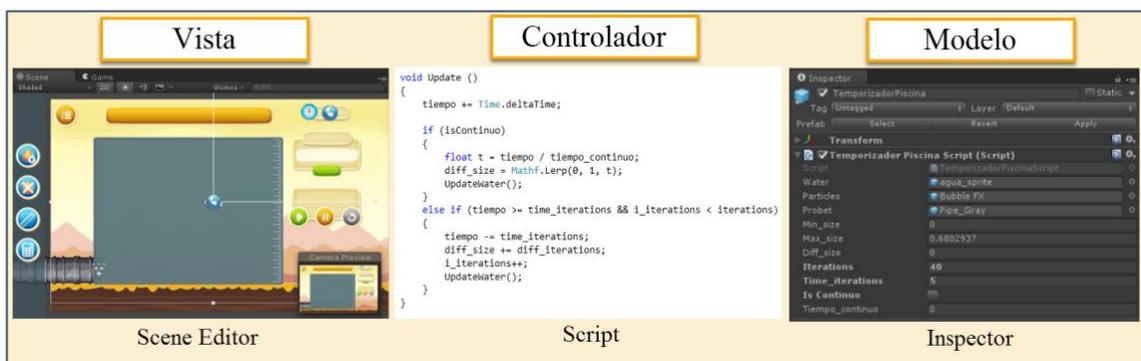


Figura 0.1 Arquitectura MVC utilizando Unity.

En la figura 0.2 se muestra el diagrama típico de interacciones entre los componentes de la arquitectura MVC y el usuario, que tiene el siguiente flujo de control en el proyecto desarrollado:

1. El usuario interactúa con los elementos del juego o con la interfaz de usuario utilizando el touch.
2. El controlador recibe la notificación de la acción realizada por el usuario, gestiona el evento que llega y ejecuta la función de código asociada.
3. El controlador accede a los datos necesarios del modelo, actualizándolo.
4. El controlador delega a la vista la tarea de actualizarse, utilizando los datos del modelo.
5. La vista espera nuevas interacciones de usuario.

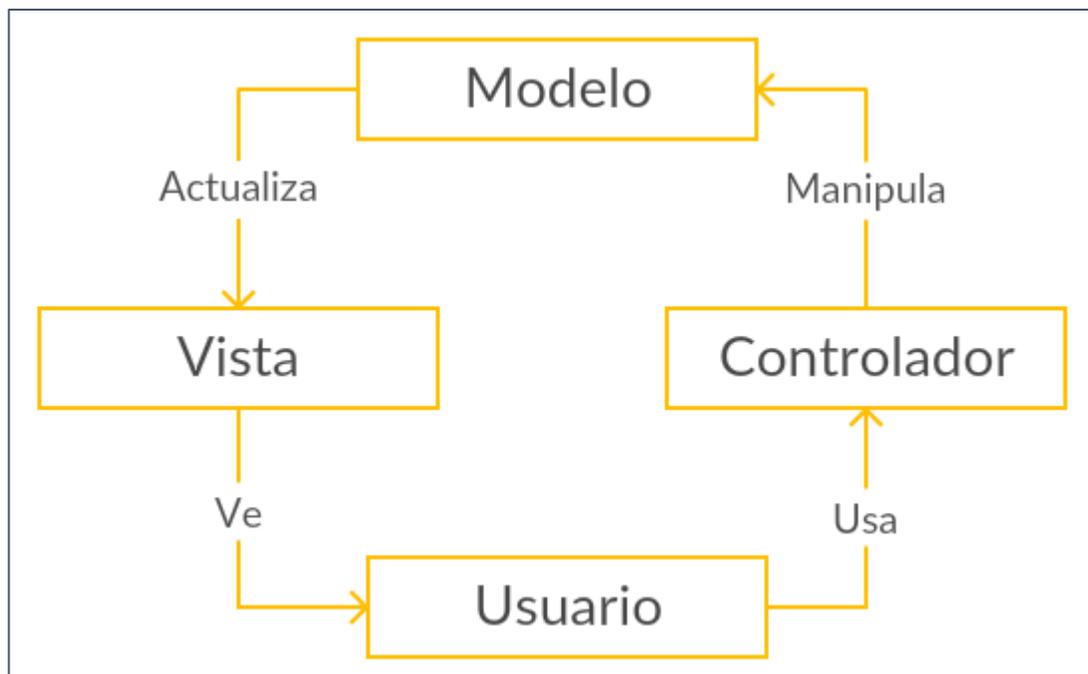


Figura 0.2 Diagrama de colaboración de componentes.

1.1.2 Diagrama de clases UML.

En ingeniería de software, las clases de código por lo general se modelan utilizando un diagrama de clases UML (Lenguaje Unificado de Modelado), en donde se describen las relaciones, atributos y métodos (Flynt, Salem, & Zerbst, 2004).

En la figura 0.3 se describe un diagrama de clases UML simplificado, en donde se muestran sólo los controladores que están presentes en todos los juegos del proyecto, y en donde se muestra la característica de modularidad, muy importante en la programación orientada a objetos, en donde el software se divide en pequeñas partes (módulos) y que es representada por una clase específica. Cada uno de estos controladores son únicos en el juego, por lo que se usa el patrón *Singleton* para su implementación (Nystrom, 2014).

Además, en el diagrama no se indican los métodos de los controladores, ya que en su gran mayoría se utilizan los métodos de eventos predefinidos de parte de Unity. A continuación, se describen cada uno de los controladores:

1. *GameController*: es el controlador principal del juego, quien se encarga de manejar los eventos principales y delegar funciones a otros controladores.
2. *AdaptativeController*: se encarga de manejar el comportamiento adaptativo del juego descrito en el modelo del capítulo 3.
3. *LogController*: se encarga de hacer un registro de cada una de las acciones realizadas por el usuario.
4. *TimerController*: se encarga de manejar el tiempo de juego, el cual varía entre 2 a 3 minutos.
5. *MenuController*: se encarga de manejar los botones del menú del juego, cuya función es realizar acciones en el juego mismo o bien en mostrar u ocultar una herramienta, como las reglas o la calculadora.
6. *AnswerController*: se encarga de manejar el panel de respuesta del juego, en donde el usuario ingresa la respuesta.
7. *FinalResultController*: se encarga de mostrar el resultado, dependiendo de la respuesta ingresada por el usuario.

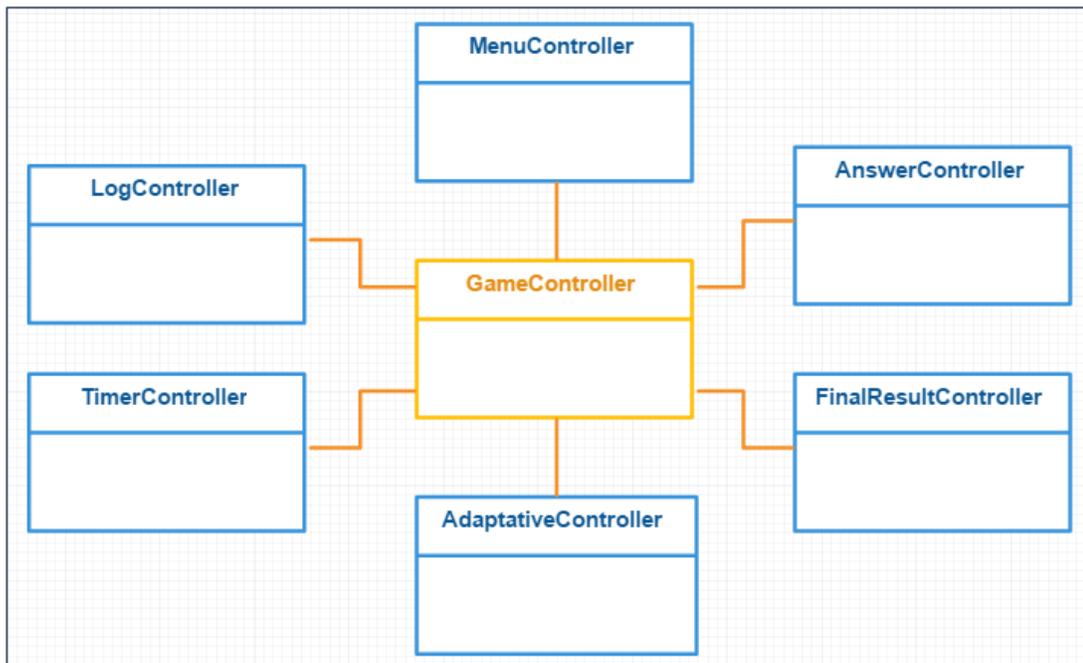


Figura 0.3 Diagrama de clases UML.

De forma adicional, cada sesión de juego cuenta con controladores propios para su funcionamiento, los que se describen a continuación, teniendo en cuenta que las mecánicas de juego asociadas se describirán a detalle en el capítulo 5:

Sesión 1:

- *BoardController*: es el controlador del tablero del juego, se encarga de manejar las fichas que se insertan e indicar si los espacios están vacíos u ocupados.
- *RulerController*: se encarga de manejar las reglas de medir y el comportamiento de *drag* sobre ellas.

Sesión 2:

- *FishController*: Se encarga de mover a los peces, dependiendo de la agrupación seleccionada de los juegos con escenarios en el fondo del mar.

Sesión 3:

- *TimerEventsController*: Se encarga de controlar los eventos periódicos del juego para poder reproducirlos, pausarlos o reiniciarlos.

Sesión 4:

- *CubeController*: Se encarga de manejar el comportamiento de touch sobre cada uno de los cuadrados de la figura, de manera tal que todos los cuadros se comporten como si fuesen una sola figura al momento de rotarla.

Sesión 5:

- *BoardController*: Se utiliza el mismo controlador que el de la sesión 1.

Sesión 6:

- *BoardController*: Se utiliza el mismo controlador que el de la sesión 1.
- *FigureActionsController*: El controlador se encarga de ejecutar las acciones de traslación y rotación a la figura creada, y al terminar el set de acciones verifica que quedó en la misma posición que la figura original.

Además, cabe destacar, que en un videojuego el código no solamente está estructurado en base a la programación orientada a objetos, existen varias clases en donde se programan “comportamientos” y éstos están estructurados en base a la programación basada en eventos. Esto se aplica para cada uno de los botones que se muestran en la aplicación y para los eventos del *touch*.

1.1.3 Implementación de la calculadora.

Se implementó la calculadora utilizando un sistema de “evaluación de expresiones”, similar a cómo funciona el método *eval* en los lenguajes de programación interpretados como JavaScript; de esta manera la expresión matemática es un *string*. Un ejemplo de esta expresión es: “3 * 5 + 2”, y al aplicar el método *eval* se tendrá como resultado el valor 17.

La estructura de una expresión matemática sigue la siguiente regla escrita en expresiones regulares: $(0 - 9)^+ [(+, -, *, /) (0 - 9)^+]^*$. Lo anterior quiere decir que una expresión

matemática es un número seguido de una combinación de operación más número. Esto último puede aparecer muchas veces como también no aparecer, además se debe observar que el número es una combinación de dígitos.

Al presionar los botones de la interfaz se va aplicando la regla anteriormente descrita y se va guardando la expresión como *string*, teniendo cuidado con algunas otras restricciones:

- En caso de que el primer botón presionado es de tipo operación, entonces el primer número es un cero.
- En caso de que se escriba un número con uno o más ceros al inicio, se eliminan todos los ceros iniciales.
- En caso de que la expresión actual termine con una operación se evalúa la expresión sin la última operación.
- En caso de que lo último presionado sea un botón de tipo operación, y se vuelva a presionar otro botón de tipo operación, se conserva sólo la operación presionada.
- Si se presiona el botón “volver”, se quita el último número u operación de la expresión.
- Si se presiona el botón “limpiar”, se eliminan todos los números y operaciones de la expresión.

Finalmente, dado que se está programando con el lenguaje C#, éste es un lenguaje compilado y no interpretado, es por ello que el método *eval* no existe dentro de este lenguaje, y por lo tanto se buscaron librerías de código escritas en C# que implementarían el sistema de evaluación de expresiones, y la mejor opción fue la librería *dotMath* (<https://dotmath.codeplex.com>) que cumple con todos los requerimientos necesarios, y está especializada en evaluar expresiones matemáticas.

1.1.4 Implementación de registro de datos.

Se implementó un sistema de registro de datos para poder recopilar todas las acciones realizadas por los alumnos para un posterior estudio del equipo de investigación, quienes **analizarán los comportamientos de los niños en cada una de las jugadas.**

Para cada una de las acciones realizadas por los usuarios, se debe complementar con algunos datos adicionales, de manera tal que al analizar estos datos se pueda inferir cómo se hizo la acción y en qué momento, es por ello que los datos guardados para cada acción son: número de la sesión de juego, número de juego y variante, nombre acción, dato 1, dato 2, tiempo de juego actual y tiempo de juego completo. Los datos 1 y 2 son datos que sirven como complemento a la acción, por ejemplo, en los juegos de la sección 1 una acción es “poner ficha en el tablero”, y los datos son la “posición x” y “posición y” en donde se puso la ficha. Los tiempos son el tiempo total de juego desde que se comenzó a jugar en la Tablet y el tiempo que le ha tomado al usuario resolver el juego actual.

Las acciones necesarias de registro para todos los juegos son: comenzar juego, presionar botón, mostrar calculadora, operación calculadora, responder, finalizar juego. Además, cada sesión de juego tiene acciones propias, las que se describen en la

tabla 0.1, y **cuyas acciones** se explicarán en detalle en el capítulo 5.

Tabla 0.1 Acciones registradas de las 6 sesiones de juegos.

| Sesión | Tema | Forma de interacción |
|--------|--|--|
| 1 | Conteo en arreglos bidimensionales. | Poner ficha, quitar ficha, mostrar reglas, ocultar reglas, mover regla horizontal, mover regla vertical. |
| 2 | Conteo por agrupaciones. | Agrupar. |
| 3 | Estimación de tiempo en eventos periódicos. | Reproducir evento, pausar evento, reiniciar evento. |
| 4 | Conteo en figuras tridimensionales. | Desfragmentar, fragmentar. |
| 5 | Cálculo de área y perímetro de figuras compuestas por rectángulos. | Agregar ficha, eliminar ficha. |
| 6 | Transformaciones isométricas. | Agregar ficha, eliminar ficha, agregar acción de traslación, agregar acción de rotación. |

Los sistemas de analítica de datos o tracking de datos, tales como Google Analytics o Flurry, se dedican a guardar un registro de acciones determinadas por el programador usando una simple línea de código, pero este sistema es online, y esto agregaría una complejidad extra a la hora de usar los juegos, ya que se tendría que añadir a los requisitos que los colegios deben contar con Wi-fi para poder ejecutarlos, y que éste funcione con buena velocidad y sin intermitencia.

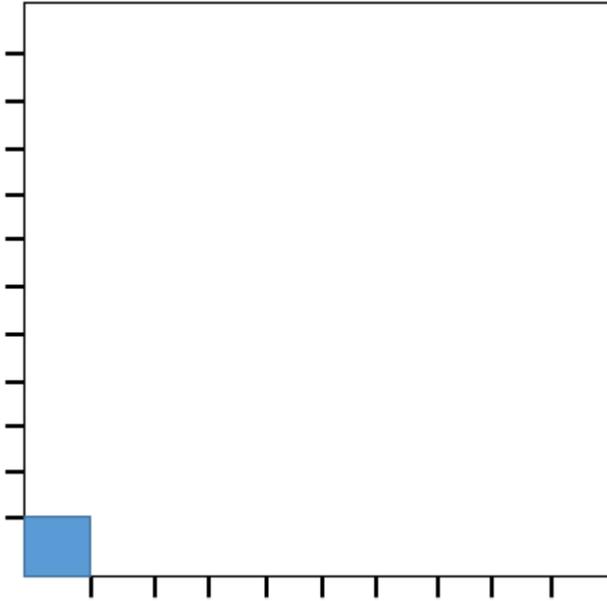
Finalmente, para implementar este sistema de registro se tomó el concepto de los sistemas actuales tracking de datos para la programación y se guardó el registro en un archivo en la memoria interna de la aplicación. El archivo está en formato CSV (*Comma Separated Value*), el cual es un documento de texto plano en donde cada línea de información cuenta con datos separados por coma, y estos archivos pueden abrirse con programas lectores de hojas de cálculo como lo es Excel. El archivo se ubica dentro de una carpeta reservada por Android a la aplicación, y la ubicación por lo general es: `Android/data/identificador`”, donde el identificador sigue la sintaxis de las aplicaciones móviles (`com.NombreCreador.NombreApp`), en nuestro caso es `com.UC.JuegoEducativoMatematicas`. En la figura 0.4 se visualiza una muestra de este archivo con datos guardados.

| | A | B | C | D | E | F | G |
|----|-----------------|-----------------|--------------------|-------|-------|-----------------------------|------------------------|
| 1 | Sesion de Juego | Numero de Juego | Accion | Dato1 | Dato2 | Tiempo actual del ejercicio | Tiempo actual completo |
| 2 | *** | *** | *** | *** | *** | *** | *** |
| 3 | 6 | 1 | ComenzarEjercicio | | | 0:00 | 0:06 |
| 4 | 6 | 1 | PonerFicha | 10 | 8 | 0:01 | 0:07 |
| 5 | 6 | 1 | BotonEliminarFicha | | | 0:04 | 0:10 |
| 6 | 6 | 1 | EliminarFicha | 10 | 8 | 0:04 | 0:10 |
| 7 | 6 | 1 | BotonPonerFicha | | | 0:05 | 0:11 |
| 8 | 6 | 1 | PonerFicha | 8 | 8 | 0:52 | 0:58 |
| 9 | 6 | 1 | PonerFicha | 10 | 7 | 0:55 | 1:01 |
| 10 | 6 | 1 | PonerFicha | 10 | 6 | 0:56 | 1:02 |
| 11 | 6 | 1 | PonerFicha | 10 | 8 | 1:05 | 1:11 |
| 12 | 6 | 1 | BotonEliminarFicha | | | 1:12 | 1:18 |
| 13 | 6 | 1 | EliminarFicha | 10 | 8 | 1:13 | 1:19 |
| 14 | 6 | 1 | EliminarFicha | 10 | 7 | 1:13 | 1:19 |
| 15 | 6 | 1 | EliminarFicha | 10 | 6 | 1:14 | 1:20 |
| 16 | 6 | 1 | EliminarFicha | 8 | 8 | 1:14 | 1:20 |
| 17 | 6 | 1 | BotonPonerFicha | | | 1:15 | 1:21 |
| 18 | 6 | 1 | PonerFicha | 10 | 1 | 1:19 | 1:25 |
| 19 | 6 | 1 | PonerFicha | 10 | 2 | 1:20 | 1:26 |
| 20 | 6 | 1 | PonerFicha | 10 | 3 | 1:21 | 1:27 |
| 21 | 6 | 1 | PonerFicha | 9 | 3 | 1:23 | 1:29 |
| 22 | 6 | 1 | PonerFicha | 9 | 1 | 1:24 | 1:30 |
| 23 | 6 | 1 | PonerFicha | 3 | 1 | 1:39 | 1:45 |
| 24 | 6 | 1 | PonerFicha | 2 | 1 | 1:40 | 1:46 |

Figura 0.4 Muestra del archivo de registro de datos.

ANEXO 2 : EJERCICIOS DEL PRE TEST (CÁCERES, 2017)**Problema 1**

¿Cuántas baldosas cubren el piso completo?

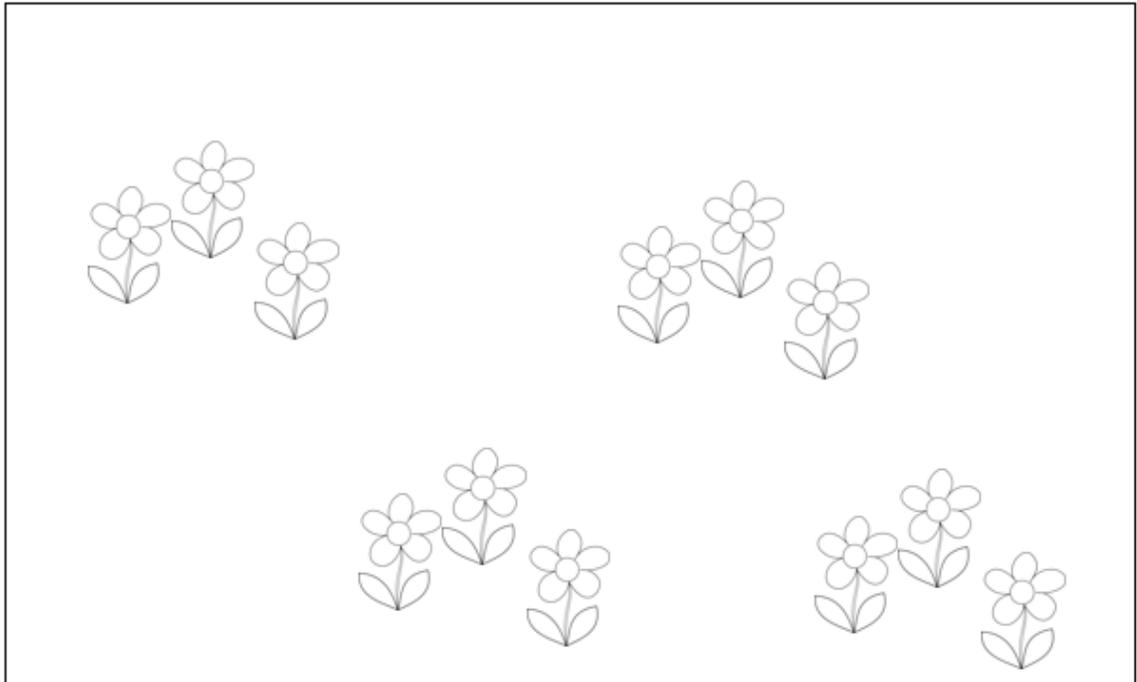


Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución

Problema 2

¿Cuántos pétalos de flores e muestran en la figura a continuación?

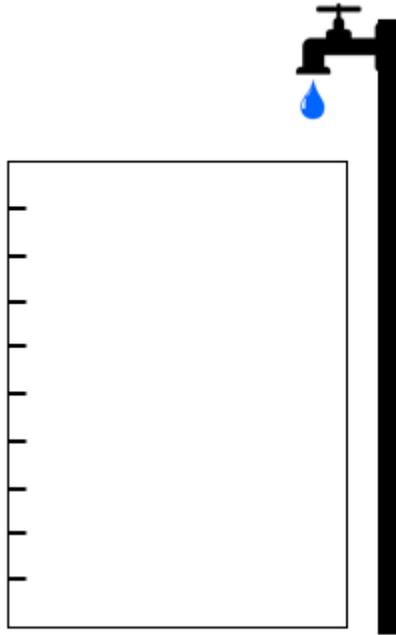


Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución

Problema 3

Cada 5 segundos el agua sube 1 metro. ¿Cuánto tiempo tomará para llenar el estanque?
(Las líneas del estanque están a 1 metro de distancia entre ellas)



Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución

Problema 4

¿Cuántos dedos hay en la sala de clases?

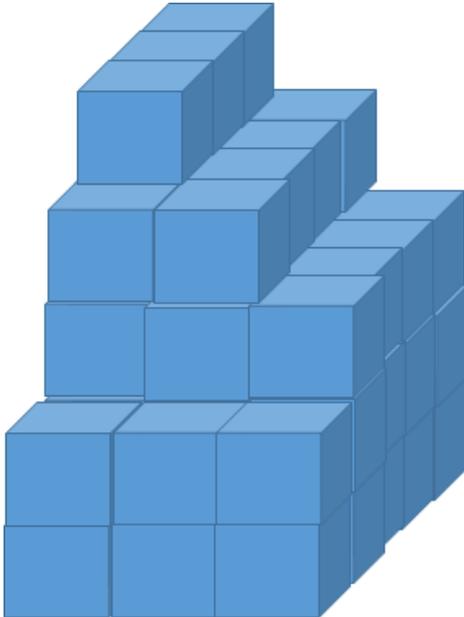
Si quieres, puedes usar el espacio abajo para ayudarte con un dibujo:

Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución

Problema 5

¿Cuántas cajas hay apiladas en la figura? Nota: No hay espacios vacíos entre las cajas.

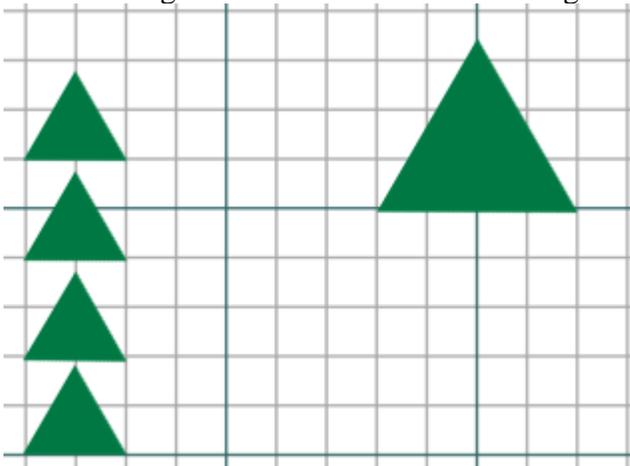


Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución

Problema 6

Usando solo transformaciones isométricas (reflexión, rotación y traslación), ¿Cómo crearías la figura de la derecha usando las figuras de la izquierda?

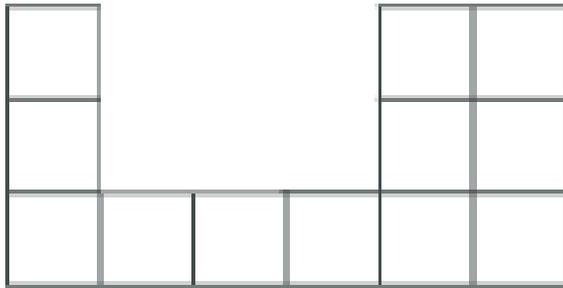


Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución

Problem 7

¿Cuál es el área y el perímetro de la figura?

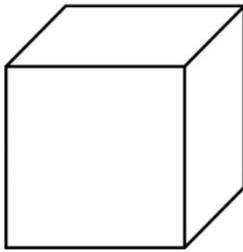


Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución

Problema 8

¿Cuántos cubos como los que se muestran en la figura son necesarios para construir una silla como en la que estás sentado?



Respuesta:

¿Qué estrategia usaste para resolver el problema? Explica paso a paso como llegaste a la solución