

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación



TESIS

Uso de una herramienta interactiva de matemáticas para
combatir el desinterés en estudiantes de secundaria ante
la crisis sanitaria

Presenta: Marco Antonio Pérez Bonilla

Para obtener el grado de: Licenciatura en Ingeniería en Ciencias
de la Computación

Director: M. C. Juan Carlos Conde Ramírez

Codirector: Dr. Héctor David Ramírez Hernández

Puebla, Pue, Febrero 2026

“Todo lo que vale la pena hacer, vale la pena hacerlo mal” - G. K. Chesterton

Dedicatoria

A mi tío y mis abuelos que, aunque ya no están, estoy seguro de que se sentirían orgullosos de ver hasta donde he llegado.

Agradecimientos

A mis asesores, por todo el apoyo (y paciencia) que me han tenido a lo largo del proyecto de tesis, a mi familia por siempre impulsarme a ser mejor, y a Dios que no me ha abandonado, muchas gracias.

Contenido

Capítulo 1: Introducción.....	7
1.1 Motivación y propósito.....	7
1.2 Problema	7
1.3 Justificación.....	8
1.4 Hipótesis	13
1.5 Objetivos.....	13
Capítulo 2: Marco teórico	14
2.1 Rezago matemático y hábitos digitales en adolescentes de Puebla.....	14
2.2 Metodologías de aprendizaje	15
2.3 Gamificación	16
2.4 <i>Serious games</i> (Juegos serios) e Interacción Humano Computadora (IHC)	17
2.4.1. Relación entre los juegos serios e IHC.....	18
2.5 Software educativo y sus categorías	19
2.6 Frameworks de JavaScript.....	20
2.6.1 Ventajas de las aplicaciones web sobre las aplicaciones de escritorio	21
2.7 Motor de juegos.....	22
2.7.1 Phaser3	23
2.8 Píxel Art	23
2.9 Sprites, spritesheets y fotogramas por segundo.....	25
2.10 LaTeX.....	26
Capítulo 3: Desarrollo de la herramienta interactiva.....	28
3.1 Infraestructura.....	28
3.2 Mapa del sitio.....	28
3.3 Mecánicas, bosquejos y guion	29
3.4 Diseño de sprites y spritesheets.....	30
3.4.1 Pixilart y Lagrida	31
3.4.2 GammaFP, Atlas Packer y Animator Tool.....	35
3.5 Código.....	37
3.5.1 Carga de elementos gráficos	39
3.5.2 Text e Image	40
3.5.3 Alpha, setAlpha, Visible y setVisible	41
3.5.4 Tweens y Timelines	42

3.5.5	Manejo de sprites	42
3.5.6	Lógica de las mecánicas	43
Capítulo 4: Pruebas y resultados		55
4.1	Pruebas de funcionamiento	55
4.2	Resultados obtenidos	57
Capítulo 5: Conclusiones y trabajo a futuro		62
Anexos		64
Bibliografía		66

Capítulo 1: Introducción

El presente proyecto busca ser una herramienta interactiva potenciada por un motor de juegos basado en JavaScript, cuyo nombre es Phaser3; enfocada en el aprendizaje de temas matemáticos referentes a los grados de primero y segundo año de secundaria, debido a que, con base en la investigación realizada, hay un bajo nivel de aprovechamiento en esta disciplina, derivado de la pandemia. De manera más específica, el proyecto engloba los temas que han sido de mayor dificultad para la muestra seleccionada, incluyendo tutoriales con ejemplos y, además, ejercicios interactivos que motiven al alumno a aprender.

1.1 Motivación y propósito

El proyecto nace de una inspiración, un recuerdo de la infancia, el software Microsoft Encarta fue prometedor en su momento, ya que tenía la intención de ayudar a los usuarios a aprender o a reforzar conocimientos preexistentes en diversas áreas como lo son la química, las artes, la música, historia, etc. Sin embargo, en un punto de la historia, la empresa previamente mencionada decidió finar a este software educativo (Merino, 2023).

Ya que la herramienta va orientada a la ludificación se pretende que, al tratarse de un programa interactivo, se despierte en los alumnos un instinto de competitividad contra sí mismos, es decir, que busquen incrementar la cantidad de aciertos que obtienen en cada ejercicio y, de manera paralela, mejoren su nivel de aprovechamiento en los temas seleccionados para la materia antes mencionada.

Adicionalmente, este proyecto también busca ayudar al alumno a detectar y corregir errores mediante mensajes de apoyo; mostrando el procedimiento adecuado para resolver los problemas matemáticos en caso de que se seleccione una respuesta incorrecta.

Se incluirán tutoriales que sean llamativos, motivo por el cual se resaltan ciertas secciones dependiendo del contenido analizado y también para que el alumno entienda correctamente el procedimiento que se tiene que realizar.

Son cuatro los temas matemáticos abordados, que son: 1) operaciones con fracciones, 2) cálculo de áreas y perímetros, 3) resolución de ecuaciones de primer grado, 4) expresiones algebraicas para sucesiones y 5) medidas de tendencia central.

1.2 Problema

Hoy en día hay pocas herramientas tecnológicas enfocadas al área de matemáticas de forma que no sea necesario que el usuario tenga una preconcepción de los temas de dicha materia. Por ejemplo, Geogebra “necesita” que sepas qué es lo que

quieres hacer, si quieres graficar debes tener una noción de lo que son las ecuaciones para estimar una “salida” probable. Siguiendo este hilo, una de las ventajas de Encarta era que no necesitabas saber nada del área para aprender, objetivo en común que tiene el presente proyecto.

Todo esto es basado en la información recolectada y a continuación presentada acerca del rezago educativo que existe en Puebla, siendo las disciplinas de español, formación cívica y ética, así como matemáticas, las más afectadas en este aspecto (Milenio, 2023), siendo de poco o nulo interés para el alumno de educación básica, inclusive llegando ellos a formular preguntas y comentarios como “¿Quién inventó las matemáticas?, ¿Para qué las inventaron si son aburridas?, No me gustan las matemáticas, ¿Para qué me va a servir esto en la vida diaria?, etc.”.

Otro de los efectos negativos generados por la pandemia es la reducción de la matrícula escolar en México, según datos de la Secretaría de Educación Pública (Secretaría de Educación Pública, 2024) y, aunque se contemplan los ciclos del 2019 al 2023 dentro de dicho reporte, noticieros de renombre, como Animal Político o la Organización Editorial Mexicana, atribuyen a que la pérdida de alumnos es multifactorial («La Pérdida de Matrícula Escolar En México: Entre el Abandono Escolar, la Privatización y los “Otros Datos”», 2025), se estima que esto ha afectado a aproximadamente 1 millón de estudiantes en toda la nación en el ciclo 2024-2025, donde la mayor tasa de incidencia se presentó en los niveles de secundaria y bachillerato (García, 2025).

Si tomamos ahora como base al Estado de Puebla y la tendencia bajista nacional, el panorama no cambia ya que, con base en los datos de la SEP referentes al ciclo escolar del 2021-2022, hay un abandono escolar del 2.8%, donde los hombres son los que desertan más que las mujeres, con un 3.5% y 2.1% respectivamente (Secretaría de Educación Pública, 2023), propiciando, todo esto en conjunto, el aumento en la brecha existente en el aprendizaje de todas las asignaturas esenciales, entre ellas, las matemáticas.

Partiendo de lo anterior y del hecho de que, también mostrado aquí, la estadística menciona que el uso de videojuegos y dispositivos electrónicos ha aumentado durante la crisis sanitaria por el virus SARS-COV-2 («Encuesta Nacional de Consumo de Contenidos Audiovisuales», 2023), por extensión, reduciendo el nivel académico en los aprendientes de nivel básico, por lo que, al motivarlos con una enciclopedia interactiva estilo videojuego, se pretende reducir esa brecha en el desarrollo de habilidades matemáticas.

1.3 Justificación

Primero cabe aclarar que el razonamiento lógico-matemático es una habilidad cognitiva fundamental cuyo fin es resolver, de la manera óptima posible, gran variedad de problemas y situaciones que se puedan llegar a presentar en la vida

diaria (Ceupe, 2023). Por ello, tal como mencionan los autores Janeth Torres Pérez, Rhonda Starr y Ruben Esquivel del sitio www.formandoformadores.org.mx, “el desarrollo de este pensamiento es clave para el desarrollo de la inteligencia matemática y es fundamental para el bienestar de los niños y niñas y su desarrollo, ya que [...] aporta importantes beneficios como la capacidad de entender conceptos y establecer relaciones basadas en la lógica de forma esquemática y técnica” (LA IMPORTANCIA DEL PENSAMIENTO MATEMATICO el. . . | Formando Formadores, s. f.). En otras palabras, con el uso de este es requerido para poder entender adecuadamente conceptos abstractos, razonar y comprender relaciones entre cada uno de estos, de forma que contribuyen al éxito y desarrollo personal.

En un mundo en el cual la tecnología es cada vez más presente, este tipo de razonamiento se vuelve extremadamente relevante cuando se trata de encontrar soluciones digitales, tal como es el desarrollo de algoritmos o, inclusive, crear software estimando cuál sería el posible comportamiento de este, tomando como base el campo de la computación. Si se sigue por este rumbo, se exigen resolver operaciones complejas y/o diseñar estructuras funcionales hablando de las diversas ingenierías o arquitectura. Hablando de economía, este tipo de pensamiento es ampliamente ocupado en proyecciones, estimaciones y análisis de mercado. (Derivando, 2022)

Entonces, este tipo de pensamiento no es de estricto uso para quienes únicamente se enfocan al área de ciencia, ingeniería y tecnología, sino también para diversas materias, como lo es la psicología, educación y sociología para poder medir efectos o, inclusive, realizar estadísticas entorno a un tema específico; la literatura y la lingüística inclusive pueden llegar a apoyarse de las matemáticas en momentos puntuales; etc. Básicamente, cualquier persona puede hacer uso del razonamiento lógico-matemático en cualquier momento (Derivando, 2022). La gran mayoría de las actividades cotidianas terminan haciendo uso de conceptos matemáticos, desde básicos hasta avanzados dependiendo de lo que se tenga que hacer, pero, ni todas son evidentes, así como tampoco todas las personas las perciben de la misma manera (Uees & Uees, 2022).

Mediante el uso del razonamiento lógico-matemático desde primaria o secundaria, se propicia que los estudiantes sean capaces de dividir un problema en sus partes esenciales, ya que este tipo de pensamiento crea nuevas conexiones neuronales a partir de situaciones complejas, es decir, “entrenamos a nuestro cerebro con tareas matemáticas complicadas porque así funciona mejor en situaciones normales complicadas” según palabras de Edu Sáenz de Cabezón del canal de YouTube “Derivando” (Derivando, 2022).

De igual forma, mediante el uso de las matemáticas, también se generan emociones que afectan a las personas, esto dependiendo de los resultados obtenidos, es decir, tanto pueden ser positivas como negativas. En esta situación, llegar a la resolución de situaciones que involucren a las matemáticas mejora las emociones por el logro

de objetivos, es decir, motiva a las personas a ser felices y a tratar de cumplir más objetivos de este estilo (Uees & Uees, 2022).

Esto, aunque sin relación aparente, se puede relacionar con el “por qué procrastinamos”, ya que se utilizan dos áreas del cerebro, el sistema límbico y la corteza prefrontal. De manera más específica, el sistema límbico es un grupo de estructuras encargadas de dirigir las emociones y el comportamiento (Sistema Límbico: MedlinePlus Enciclopedia Médica Ilustración, s. f.), además de segregar dopamina, neurotransmisor que se encuentra asociado al aprendizaje y al sistema de recompensas del cerebro (Dopamine, 2022); mientras que la corteza prefrontal es “nuestra parte más racional” ya que se encarga de los procesos cognitivos y las funciones ejecutivas, por ello tarda en desarrollarse, aproximadamente entre los veinte y veinticinco años de edad (Castillero Mimenza, 2025).

La procrastinación entonces no es otra cosa más que la postergación voluntaria por parte de un sujeto a una acción o actividad planificada (insightspsychology, 2024), provocada principalmente por la lucha de poder entre el sistema límbico y la corteza prefrontal. Esto es especialmente importante y vinculatorio con el desarrollo del razonamiento lógico-matemático porque el sistema límbico se encargará de “evitar molestias” (insightspsychology, 2024) a partir de la producción de la dopamina, que, como ya se ha hablado, es la hormona de la felicidad. “Cuando una tarea desencadena sentimientos abrumadores, la amígdala (parte del sistema límbico) se activa, dirigiéndolo hacia un alivio inmediato, como desplazarse por las redes sociales o ver una serie” según el artículo de Insights Psychology en “La neurociencia de la procrastinación: ¿Qué sucede en tu cerebro?” (insightspsychology, 2024).

Para poder evitar a la procrastinación es necesario “vencer” a la dopamina, para ello entra en juego la corteza prefrontal, que se encarga de luchar para anular los impulsos emocionales generados por el sistema límbico, este “estira y afloja” no dura más de cinco minutos, ya que es lo que tarda este último en resistirse, y con ello, se pierde cualquier tipo de oposición a realizar cualquier trabajo.

Sin embargo, ¿qué pasaría si “engañamos” al cerebro para que lo que considera “una molestia” ya no lo vea así?, es aquí donde entra un nuevo actor, el sistema de recompensas del cerebro. Este último es un “centro en el sistema nervioso central que obedece a estímulos específicos y naturales. Regulado por neurotransmisores, permiten que el individuo desarrolle conductas aprendidas que responden a hechos placenteros o de desagrado” según palabras del Dr. Belsasso, Dr. Estañol y el Dr. Juárez, autores del libro “Nuevas estrategias farmacológicas en el tratamiento de las adicciones” (Belsasso et al., 2002).

Por lo tanto, si se estimula adecuadamente el sistema de recompensas, se puede lograr que la corteza prefrontal obtenga el control sin tanta resistencia por parte del sistema límbico, esto a partir de un elemento que fue mencionado en la problemática, el consumo de videojuegos, sólo que, en este caso, con un fin lúdico para promover el aprendizaje, respondiendo así la pregunta previamente planteada,

es decir, “engañamos” al sistema límbico con el uso de videojuegos lúdicos y, a la vez, logramos cederle el control a la corteza prefrontal para así poder desarrollar el razonamiento lógico-matemático. En esencia, buscando reducir la molestia percibida por los alumnos hacia las matemáticas.

La UNICEF considera que el juego es provechoso, divertido, iterativo, socialmente interactivo y, además, invita a la participación activa; todo esto ya que los niños hacen este tipo de actividades para darle sentido y explicación a algo que ya conocían previamente; además de disfrutar, motivarlos y despertar emoción y/o placer. El juego no es algo estático, ya que se puede ocupar para practicar, crear y probar estrategias a través del planteamiento de hipótesis y descubrir nuevos retos para, de esta forma, no solamente reforzar lo que ya se sabía, sino generar nuevo conocimiento (Aprendizaje A Través del Juego: Reforzar el Aprendizaje A Través del Juego En los Programas de Educación En la Primera Infancia, 2018). A esto último, desde un enfoque pedagógico, se le conoce como metacognición, visto desde la taxonomía de Marzano/Bloom, mismas que se aprecian en las imágenes 1 y 2 respectivamente. Entonces, el juego tiene por propósito llenar esa necesidad fundamental de la humanidad de poder manifestar lo que ocurre en la imaginación, a través de la curiosidad y la creatividad. “Estos son recursos clave en un mundo basado en el conocimiento, y nos ayudan a afrontar las cosas, a ser capaces de disfrutar y a utilizar nuestra capacidad imaginativa e innovadora” (UNICEF, 2018).

TAXONOMÍA DE MARZANO

Nueva Taxonomía de los Objetivos Educativos



Imagen 1 Taxonomía de Marzano



Imagen 2 Taxonomía de Bloom

Teniendo en cuenta que vivimos en un mundo tecnológico y conectado, es necesario voltear a ver, por un momento, a la industria de los videojuegos, que hoy por hoy, tiene una enorme importancia dentro de la sociedad, al grado de que al año 2022 hubo un ingreso global estimado en 180,000 millones de dólares (Iberdrola, s. f.). Esto, en conjunto con datos que se abordarán más adelante, sólo demuestra lo obvio, los videojuegos ya son una parte intrínseca de la sociedad sin importar la edad.

La ludificación, concepto que también será abordado más adelante por su enorme importancia, busca dinamizar la educación, para lograr que los alumnos no recurran en su totalidad a la teoría, que de por sí no es de su total agrado, y hagan cosas que sean divertidas y emocionantes para ellos, haciendo así que ellos asimilen, comprendan y retengan contenidos de manera casi desapercibida; incrementa la motivación, ya que con cada avance se les premia con coleccionables, como lo son medallas, desbloqueables, bonus, etc.; y también facilita la práctica, pudiendo llevar lo aprendido a su contexto diario (Iberdrola, s. f.). Esto se relaciona directamente con el estímulo al sistema de recompensas del cerebro, logrando así vencer al sistema límbico y facilitar el trabajo de la corteza prefrontal del cerebro.

Algunas ventajas detrás de la ludificación son la mejoría en la capacidad de generar respuestas ya que agilizan al cerebro en la resolución de problemas, más cuando se tiene un tiempo limitado para resolverlos; promueven el trabajo en equipo, ya que refuerzan la capacidad de idear estrategias para llegar a un objetivo en común, mejorando así la toma de decisiones; estimulan la creatividad, atención y memoria visual ya que existen retos que obligan a la concentración, pensar “fuera de la caja” y ser observador en los detalles; enseñan idiomas, a través de la prueba y error, con consejos en pantalla para hacer una retroalimentación, o inclusive para comunicarse con otros jugadores, así como también no perderse parte de la historia

si es que el videojuego se encuentra en otra lengua; y, por último, favorecen el pensamiento crítico, ayudando a reflexionar a los jugadores en todas las acciones que realizaron, tanto buenas como malas, con el fin de poder mejorar (Iberdrola, s. f.).

Es tanta la importancia de la ludificación que, según las consultoras Deloitte y Gartner, el 25% de los procesos empresariales englobarán también dinámicas de este estilo en los años futuros, además de que esta cantidad podría llegar al 50% en aquellos relacionados a la innovación, sin mencionar que el 70% de empresas que conforman la *Global 2000 list, ranking* de la revista Forbes que clasifica a las 2000 empresas públicas más grandes del mundo, ya emplea la gamificación dentro de alguno de sus procesos (Playmotiv & Playmotiv, 2022). Además, según la universidad de Stanford, la ludificación, implementada adecuadamente, puede aumentar hasta en un 40% la absorción de contenidos en comparación con los métodos tradicionales de enseñanza, además de hacer que el alumnado muestre una mayor motivación y compromiso con el aprendizaje, impactando de manera positiva en el rendimiento académico de ellos (blogs-es.vorecol.com & Vorecol Editorial Team, s. f.).

1.4 Hipótesis

Contribuir a la disminución del rezago educativo y aumentar el interés de alumnos de secundaria en el área de matemáticas mediante el uso de la ludificación, a través de la implementación de una herramienta interactiva estilo videojuego, evaluando principalmente el cambio en el interés y desempeño inmediato.

1.5 Objetivos

Objetivo general: Crear una herramienta de estudio estilo videojuego 2D relacionada a las matemáticas con el fin de despertar el interés de los estudiantes de secundaria en esta área, obteniendo como mínimo, un prototipo funcional.

Objetivos específicos:

- Diseñar la estructura y la interfaz de la herramienta interactiva.
- Desarrollar contenido educativo atractivo.
- Integrar características de videojuegos.
- Evaluar el uso de la herramienta en estudiantes de secundaria resultado de la aplicación de una evaluación diagnóstica para evidenciar los cambios presentes en los participantes.
- Documentar el proceso de desarrollo.
- Difundir los resultados.

Capítulo 2: Marco teórico

Para poder respaldar el por qué de la creación de la herramienta, en el presente capítulo se abordarán temas importantes, como lo son: la cantidad de rezago educativo existente en el Estado de Puebla, más específicamente, en niños y jóvenes de dicha Entidad Federativa; cantidad de jóvenes que ocupan su tiempo en los videojuegos; lo que son las metodologías de aprendizaje, con énfasis en las activas, y la importancia de estas en la educación, así como la relación de estas con la gamificación; el por qué esta es especial para mejorar la absorción de conocimientos por parte del alumnado y un caso exitoso de su implementación. De igual manera surgen otros temas como: los juegos serios y su relación con la interacción humano computadora; las ventajas de las aplicaciones web sobre las de escritorio, con frameworks de JavaScript; Phaser3 como un derivado de este último y como motor de juegos y la teoría relacionada a las animaciones y píxel art.

2.1 Rezago matemático y hábitos digitales en adolescentes de Puebla

A partir de la investigación realizada, se ha encontrado que el 54% de niñas y niños en México juegan videojuegos, ocupando aproximadamente 2.1 horas al día («Encuesta Nacional de Consumo de Contenidos Audiovisuales», 2023), además de que, en cuestión a Puebla, se estima que los jóvenes dedican un 75% de su tiempo libre para esta actividad, donde los varones juegan más en comparación con las mujeres. (Guerrero, 2022)

Tomando lo anterior en cuenta, también se analizó la cantidad de rezago que hay en el estado de Puebla en el área de matemáticas, encontrando que uno de cada cuatro habitantes se encuentra en rezago educativo según datos de la CONEVAL (Consejo Nacional de Evaluación de la Política de Desarrollo Social), (Milenio, 2023) lo que implica que no terminaron su educación secundaria, media superior o superior; además de que, según datos de la SEP (Secretaría de Educación Pública), Puebla tiene un rezago educativo del 35.5 por ciento aproximadamente. (Ruiz, 2023)

Para obtener información más precisa, se tomaron los resultados de la evaluación diagnóstica estandarizada de una escuela aleatoria, siendo esta aplicada a nivel nacional por la entidad federal MEJOREDU, tomando como muestra a los alumnos de primero y segundo de secundaria, obteniendo que ellos tienen rezago en las áreas de Sentido numérico, álgebra y variación: extensión del significado de las operaciones y sus relaciones inversas; Forma, espacio y medida: medición y cálculo en diferentes contextos; Forma, espacio y medida: regularidades y patrones; y, por último, Manejo de la información: interpretación de la información a través de medidas de tendencia central y dispersión; apegándose a los nuevos planes de estudio pertenecientes a la NEM (Nueva Escuela Mexicana).

2.2 Metodologías de aprendizaje

Para poder entender cómo es que los alumnos analizan y absorben la información, es necesario recurrir a las metodologías de aprendizaje y sus diversos tipos. La Universidad Europea define a estas como “conjunto de estrategias, procedimientos y acciones organizadas y planificadas por el profesorado, de manera consciente y reflexiva, con la finalidad de posibilitar el aprendizaje del alumnado y el logro de los objetivos planteados” (Universidad Europea, Universidad presencial (Madrid, Valencia, Alicante, Canarias, Málaga) y Online, 2023) que, básicamente significa apoyarse de múltiples estrategias, herramientas y recursos para que el estudiantado alcance la mayor cantidad posible de conceptos clave o fundamentales referentes a alguna materia. Simplificándolo, las metodologías de aprendizaje tienen por fin motivar al alumno y ayudarlo a aprender.

Las metodologías de aprendizaje tienen dos clasificaciones, las tradicionales, o pasivas, y las innovadoras, o activas. En la primera, son los docentes quienes proporcionan la información a los alumnos, generalmente mediante la exposición, mientras que los últimos únicamente la reciben, el problema radica en que se puede caer en que el rol del estudiante sea exclusivamente de receptor, aludiendo al aprendizaje basado en la memorización. En la segunda se busca que el estudiante se vea involucrado, estimulando así el pensamiento crítico y la aplicación, de manera práctica, de los conocimientos adquiridos (Metodologías de Enseñanza Innovadoras: ¿Cuáles Son las Más Efectivas?, 2025).

Dentro de los distintos tipos de metodologías de aprendizaje activo, se pueden listar algunas como las siguientes.

- **Aprendizaje Basado en Proyectos (ABP):** En este tipo de metodología se busca que el alumno ponga a prueba sus conocimientos a partir de proyectos proporcionados por el docente o, incluso, propuestos por el primero, de forma que, preferentemente, trabaje en grupos con sus compañeros.
- **Aprendizaje Basado en Problemas (ABPr):** El docente, a partir de una situación hipotética y contextualizada, plantea una problemática al alumnado, de forma que ellos busquen la solución óptima a esto.
- **STEM/STEAM (Ciencia, Tecnología, Ingeniería, Artes y Matemáticas):** La SEP menciona que la metodología STEM/STEAM tiene por propósito “diseñar proyectos interdisciplinarios y atender problemas en su contexto que demanden de conocimientos relacionados con Ciencia, Tecnología, Ingeniería, Artes y Matemáticas para su solución” (Apoyo A la Apropiación del Plan y Programas de Estudio 2022: Enfoque STEAM, 2024).
- **Repetición espaciada:** Duolingo menciona a través de su blog que esta metodología consiste en repasar los contenidos y/o ponerlos en práctica de manera constante para ayudar a recordarlas mejor, con el fin de fortalecer la memoria a largo plazo, ya que es fácil olvidar la información si no se ha empleado durante un tiempo (MS, 2025).

- **Gamificación**, abordada completamente en la siguiente sección.

2.3 Gamificación

El proyecto tiene como base apoyarse de la ludificación o gamificación que, como señala el banco Santander en un artículo, “Se trata de un método de enseñanza con el que se pretende que los alumnos aprendan a través de juegos” (Santander Open Academy, 2025), de forma que se tengan en cuenta ciertos factores para que esta sea exitosa, como lo son el crear hábitos, esto implementado retos para que los usuarios tengan la necesidad de superarlos; progresión, para que ellos sientan que progresan en todo momento; y el ganar recompensas para que sientan que todo el tiempo que han invertido valga la pena (Mecánicas de Juego Más Habituales En los Videojuegos | Tokio School, s. f.).

Además, en palabras de Bernardo Jareño Manclús, catedrático del Colegio Alberto Sols, “Intentamos potenciar al máximo las habilidades de los niños a través de la experimentación y el juego, puesto que mediante la experiencia y la diversión creemos que es más fácil alcanzar aprendizajes más significativos y funcionales”. (EDUCACIÓN 3.0, 2019)

Ahora bien, la ludificación o gamificación proviene de la palabra anglosajona “game”, que literalmente se traduce como “juego”, que consiste en aprovechar los recursos y herramientas que estos proveen para mejorar el rendimiento de los usuarios y, por extensión, sus respectivos resultados académicos (Santander Open Academy, 2025).

En este contexto, algunas técnicas dinámicas que se pueden emplear para lograr el éxito de la ludificación son las siguientes:

1. Recompensa: Obtener un beneficio merecido.
2. Logro: Como superación o satisfacción personal.
3. Competencia: Por el simple afán de competir e intentar ser mejor que los demás.

Entre las aplicaciones que incorporan esta metodología activa para optimizar el aprendizaje de sus usuarios, se pueden mencionar las siguientes:

- **Duolingo**: Aplicación utilizada para el aprendizaje de idiomas mediante el uso de elementos interactivos.
- **Brainscape**: Implementa el uso de tarjetas de estudio para poder aprender nuevos temas o, incluso, repasar conceptos del interés del usuario.
- **Minecraft Education Edition**: Basado en el juego con el mismo nombre, el propósito es promover que los usuarios, en este caso estudiantes y profesores, colaboren entre sí, aplicando la imaginación para resolver problemas en un entorno que les sea entretenido.

- **Kahoot!:** Es una herramienta tipo encuesta que permite a los educadores generar preguntas y respuestas personalizadas implementando un sistema de puntaje que le puede resultar interesante a los estudiantes.
- **CodeCombat:** Esta herramienta es más un videojuego en sí, sin embargo, se enfoca en que sus usuarios aprendan a programar, es por esto por lo que ellos deben resolver las distintas problemáticas que surjan a través de la implementación de código.

Sin olvidar mencionar, por lo menos, uno de los casos de éxito de la ludificación, siendo este el de Jann Mardenborough, piloto británico de carreras que, antes de jugar el videojuego “Gran Turismo” de PlayStation, era un estudiante; fue que, a través del simulador, incluyendo su participación y victoria en la *GT Academy*, logró llegar hasta donde está ahora (Jann Mardenborough - GT Academy, s. f.).

2.4 *Serious games* (Juegos serios) e Interacción Humano Computadora (IHC)

Los juegos serios, conocido como *serious games*, pretenden ser una herramienta que implementa características de la gamificación sobre elementos que, esencialmente, no tienen nada que ver, de forma que se busca que el usuario tenga una experiencia positiva y didáctica, con un propósito más formativo que de entretenimiento. Si bien se les llama “serios”, esto es porque este tipo de software no pretende ser un videojuego en toda regla, sin embargo, sí busca ser empleado en áreas como la educativa, científica, médica, etc. (Fuerte, 2022).

Entonces, los *serious games* tienen por finalidad enseñar y divertir a los usuarios para el desarrollo de habilidades concretas, desde el aprendizaje de idiomas o matemáticas, lo cual se convierte en una experiencia doblemente positiva, ya que se aprende jugando.

Si bien existe cierto debate sobre el uso de los videojuegos como herramienta educativa, principalmente por su naturaleza, existe evidencia que respalda su efectividad. Las estrategias lúdicas propias de los juegos serios crean una metodología activa conocida como Aprendizaje Basado en Juegos. Esta provoca que los alumnos sean enganchados con mayor facilidad tanto por docentes como por los contenidos educativos de los programas de estudio, permitiendo aprender de manera divertida.

Sin embargo, es indispensable tener en cuenta la curva de dificultad, apreciable en la imagen 3. Si un juego es demasiado complicado y las habilidades del usuario son bajas, puede generar frustración. En cambio, si el juego es demasiado fácil y el usuario tiene habilidades altas, este terminará por aburrirse.

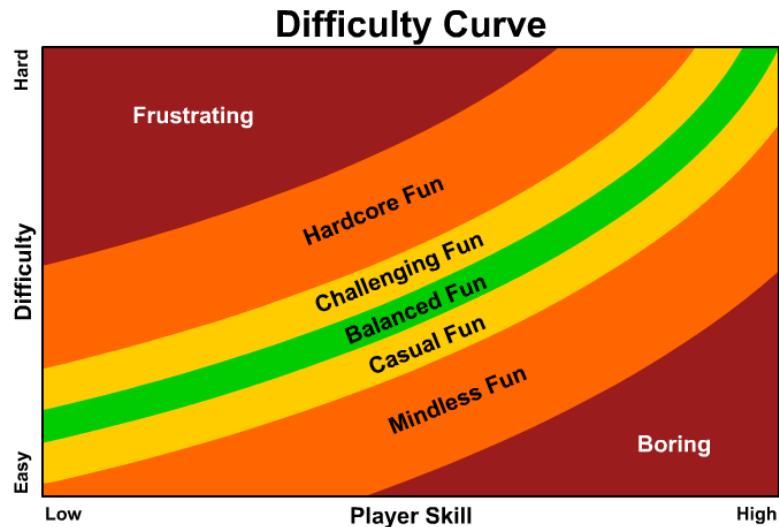


Imagen 3 La curva de dificultad de un videojuego (Castaño, 2023)

Por su parte, la IHC “trata sobre el diseño, implementación, y evaluación de sistemas interactivos en el contexto de ciertas actividades del usuario” (Castro & Rodríguez, 2020), es decir, esta no se centra en un único usuario, ni en una sola computadora, al contrario, toma a un conjunto de humanos, así como diversos sistemas (como las propias PC, laptops, teléfonos celulares, etc.) y analiza como los primeros utilizan los últimos, es por esto por lo que, realmente, se habla de sistemas interactivos.

La interacción humano computadora, además, prioriza la usabilidad, que significa que el usuario pueda realizar sin problemas las tareas que él desee o, en otras palabras, las máquinas puedan facilitar el trabajo de los humanos y no complicarlo y, de ser así, entonces se considera que el sistema no es usable. Es decir, “para que un sistema tenga éxito, éste debe ser útil (hacer lo que se desea hacer: escuchar música, cocinar), usable (fácil de usar, fácil de aprender, sin errores), y usado de manera frecuente (aceptado, de manera que la gente desee usarlo porque es útil, divertido o atractivo)” (Castro & Rodríguez, 2020).

2.4.1. Relación entre los juegos serios e IHC

Como ya se mencionó, existe una relación entre los juegos serios y la Interacción Humano Computadora, basada en los siguientes aspectos:

- Proporcionan y propician un diseño centrado en el usuario: ambos se enfocan en entender a este, específicamente sus necesidades, motivaciones y limitaciones para que, a partir de esto, se desarrolle una interfaz o una experiencia personalizada.
- Promueven la interactividad: en los juegos serios y la IHC, la interactividad es un elemento clave, tanto que estos buscan facilitar la interacción con los

sistemas y, en el caso del primero, también generar aprendizaje o impacto a través del juego.

- Brindan retroalimentación activa: en ambos, se busca proporcionar una retroalimentación al jugador/usuario con el fin de guiarlo y así mantenerlo inmerso en el sistema.
- Se enfocan en la usabilidad y experiencia del usuario (UX): en ambos casos, se requiere que exista una experiencia intuitiva, clara y fluida ya que, si esto no ocurre, entonces el sistema no podrá cumplir con el propósito para el cual fue diseñado.
- Ambos pueden aplicar narrativas para motivar a los usuarios: si bien esto es más común en los juegos, IHC también puede hacer uso de este recurso para despertar el interés del usuario, sobre todo en aplicaciones educativas o de salud.

Además de compartir todo eso en común, hay elementos aún más específicos que hacen de ambos estar aún más conectados, tal como los siguientes:

- Accesibilidad: Sistemas/juegos adaptados para todo tipo de público.
- Feedback inmediato: Mensajes, sonidos, animaciones, indicadores visuales, etc., que notifiquen sobre alguna acción realizada, progreso y/o errores.
- Curva de aprendizaje: Interfaz intuitiva junto con tutoriales y/o niveles progresivos.
- Metáforas visuales: Iconos, botones, mecánicas, etc., que recuerden a algo conocido para que, de esta forma, se sepa qué acción se está realizando o qué se puede desencadenar.
- Motivación y compromiso: Experiencia fluida y mecánicas lúdicas junto con recompensas.
- Interacción y pruebas con usuarios: Pruebas de usabilidad, jugabilidad y eficiencia.
- Diseño emocional: Colores, narrativa, estética, música, etc.
- Contexto de uso: Diseño según el dispositivo y el tipo de entorno (educativo, personal, científico, médico, etc.)

2.5 Software educativo y sus categorías

El software a desarrollar, si bien se apoya de la gamificación, no es un juego en toda regla, sino más bien, un software educativo, que se define como “programas de ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y aprendizaje” (García Peñalvo, 2009), además de que permiten poder dividirse en múltiples categorías como LTS (*Learning Technology Systems*), CBL (*Computer Based Learning*), CAI (*Computer-Assisted Instruction*), CBI (*Computer-Based Instruction*), CBT (*Computer-Based Training*), CAL (*Computer-Aided Learning*), CME (*Computer-Mediated Education*), CMI (*Computer-Managed Instruction*), ECB (*Electronic*

Course Book), ITS (*Intelligent Tutoring System*), WbIS (*Web-based Instructional Systems*), o LMS(*Learning Management Systems*), entre otros, sin embargo, por la cantidad de estos, se puede generar bastante confusión (García Peñalvo, 2009).

De estas categorías, la más resaltante sería la ITS, esto porque se dividen en tres tipos de enseñanza:

- Tutoría, con un estilo orientado a dirigir y a presentar, semejante a los sistemas CAI, pero la selección y la secuencia de los materiales docentes no ha sido predefinido, sino que se tiene en cuenta el modelo de estudiante para generar una ruta individualizada a través de los materiales para cada estudiante.
- Diálogo socrático, de forma que el sistema envuelve al alumno en un diálogo e intenta que éste descubra sus ideas equivocadas haciéndole preguntas que creen controversia en sus creencias.
- Entrenamiento, caracterizado por el papel pasivo del sistema, que está en un segundo plano observando y evaluando el rendimiento del estudiante, y está siempre disponible para ofrecer ayuda, criticar una solución o interferir si fuera necesario.

Siguiendo con esta rama, la herramienta digital correspondiente al presente proyecto será del tipo ITS, enfocada al entrenamiento, ya que sólo ofrecerá mensajes de apoyo y guía cuando así se considere necesario, como lo es cuando el usuario seleccione una respuesta incorrecta a determinado ejercicio.

2.6 Frameworks de JavaScript

Para poder hablar de lo que son los frameworks, es necesario hablar inicialmente de términos como Back-end y Front-end que, de manera resumida, el primero se encarga del lado del cliente, es decir, bases de datos, autenticación, etc., mientras que el último del lado del cliente, o más bien, de lo que ve e interactúa a través del navegador web. Para el Front-end se utilizan diversos elementos, como lo es HTML, CSS y/o JavaScript, así como derivaciones del último, conocidas como Frameworks, cuyo objetivo es simplificar y agilizar el desarrollo de aplicaciones Web.

En esencia, los frameworks de JavaScript se ocupan como parte fundamental del desarrollo web Front-end para mejorar la experiencia del usuario final, proveyendo a los desarrolladores herramientas probadas para la creación de aplicaciones web que puedan ser de fácil escalabilidad y, a la vez, interactivas. El desarrollador web, además, tiene múltiples opciones de las cuales puede disponer, dependiendo del propósito para el que se requiera, ya que cada una de estas puede llegar a tener un enfoque distinto.

Como se mencionó anteriormente, existen múltiples tipos de frameworks de JavaScript en el mercado, todo depende de la finalidad con la que se pretenda utilizar. Algunos de estos se listan a continuación.

- **Angular:** Respaldado por Google, Angular es uno de los frameworks más potentes que existe, útil cuando se pretende crear un sitio web interactivo y dinámico, que contenga una sola página. Es amigable con los desarrolladores que no han utilizado este tipo de herramientas antes, ya que, una vez dominado, facilita poder aprender otros frameworks («Los 6 Principales Frameworks de JavaScript Que Debes Conocer | UNIR», 2023).
- **React:** Respaldado por Meta, permite desarrollar aplicaciones web intuitivas, de forma que permite un trabajo versátil, robusto y fiable. Facebook no es el único sitio que implementa React, también Instagram o Airbnb se apoyan de este («Los 6 Principales Frameworks de JavaScript Que Debes Conocer | UNIR», 2023).
- **Vue.js:** Ofrece una interfaz de programación de aplicaciones amigable con el desarrollador, además de permitirle modularizar, es decir, el programador puede escoger de este framework los módulos que le interesen y descartar los demás. Nintendo implementa esta tecnología en sus sitios web oficiales («Los 6 Principales Frameworks de JavaScript Que Debes Conocer | UNIR», 2023).
- **Ember:** Uno de los frameworks más populares desde su entrada al mercado en 2015, ofrece enlace de datos bidireccional, que permite una actualización en tiempo real cuando múltiples dispositivos se conectan o acceden al sitio. Empresas y aplicaciones como LinkedIn o Apple Music la implementan («Los 6 Principales Frameworks de JavaScript Que Debes Conocer | UNIR», 2023).
- **Node JS:** A diferencia de los demás frameworks que se ejecutan del lado del cliente, Node JS tiene la particularidad de hacerlo del lado del servidor, esto para poder desarrollar aplicaciones que faciliten la escalabilidad de estas, lo cual es útil para aquellas que vayan a tener un tráfico de datos muy elevado en tiempo real. Netflix, PayPal o Uber son ejemplos de sitios que la han implementado («Los 6 Principales Frameworks de JavaScript Que Debes Conocer | UNIR», 2023).
- **Phaser3:** Este es un framework especial, ya que se centra en el desarrollo de videojuegos 2D, siendo de código abierto y, además, amigable con el desarrollador.

2.6.1 Ventajas de las aplicaciones web sobre las aplicaciones de escritorio

Una aplicación web “es un programa de software que se ejecuta en un servidor remoto y al que se accede a través de un navegador web a través de Internet o una intranet” (Mika, 2025). Además, por su naturaleza, no requieren ser descargadas ni instaladas en ningún equipo de cómputo. Las aplicaciones de escritorio, por otro

lado, “son software que se descarga e instala directamente en la computadora de un usuario en lugar de ejecutarse de forma remota a través de la web” (Mika, 2025).

Entonces, las ventajas de las aplicaciones web se pueden listar de la siguiente forma:

1. Son accesibles desde cualquier lugar, siempre y cuando se tenga acceso a internet, esto las vuelve bastante accesibles y convenientes para los usuarios.
2. Son capaces de funcionar en distintas plataformas y dispositivos, tal que pueden acceder a estas a través de teléfonos, tabletas, PC, etc., sin la necesidad de tener que crear aplicaciones específicas para cada uno de estos y sin tener en cuenta los distintos tipos de sistemas operativos existentes.
3. No consumen almacenamiento de los dispositivos ya que estas se encuentran alojadas en la nube, permitiendo adaptarse rápidamente a muchos usuarios, a diferencia de las aplicaciones de escritorio.

2.7 Motor de juegos

También conocido como *game engine*, no es otra cosa más que un conjunto de herramientas creadas para el desarrollo y optimización de un videojuego, implementando, regularmente, un bucle de juego y también un motor de renderizado 2D o 3D según se requiera, ya que, con este último, se maneja la gestión de memoria y el *buffering* de los elementos gráficos que van a aparecer en pantalla (Universidad Europea, 2024).

Los motores de juego del mercado, por lo regular, implementan algunas herramientas, como lo son las siguientes.

- **Bucle de juego:** Uno de los elementos más importantes del motor de juegos, ya que se encarga de la ejecución en tiempo real del videojuego (Universidad Europea, 2024).
- **Motores de animación:** Su funcionalidad es la carga y almacenamiento de los recursos gráficos y, además, de darle movimiento a estos, en caso de ser requerido, a través de la interpretación de un atlas u hoja de sprites, también conocida como spritesheet (Universidad Europea, 2024).
- **Motores de audio:** Responsable de la carga y almacenamiento de archivos de audio, además de la reproducción total o parcial de estos según se requiera (Universidad Europea, 2024).
- **Multijugador y herramientas de red:** Se encarga de establecer conexión con otros dispositivos, además del intercambio de datos en tiempo real (Universidad Europea, 2024).

- **Inteligencia Artificial:** Esta herramienta es multipropósito, desde darle “inteligencia” a los enemigos, hasta ajustar la dificultad de manera dinámica para adecuarse a las habilidades del usuario (Universidad Europea, 2024).

2.7.1 Phaser3

Phaser3, tal como se mencionó con anterioridad, es un framework de JavaScript orientado al desarrollo de videojuegos. Esta herramienta fue creada por Richard Davey con el fin de crear juegos de forma rápida para navegadores web, teniendo como ventaja que se pueda utilizar tanto en dispositivos móviles como PC, de forma que este se carga en un Canvas de HTML o en WebGL, permitiendo hacer composiciones de fotos o inclusive realizar animaciones (Navarro, 2021).

Phaser3 implementa algunos de los elementos mencionados en el apartado anterior y unos extras, siendo las siguientes:

- **Game Object:** Esta clase se encarga de extender a todos los objetos del juego (Navarro, 2021).
- **Game Loop:** Siendo este el bucle de juego, se encarga de detectar todos los eventos y cambios que se puedan producir mientras este está en ejecución (Navarro, 2021).
- **Escena:** Un videojuego es comparable a una obra de teatro, en el aspecto de que está compuesta por escenas, en la primera hay distinta utilería y fondos, mientras que en la segunda hay distintas lógicas, dependiendo de lo que se requiera, tanto así que estas se pueden ejecutar en paralelo o de forma simultánea, dependiendo del evento o condicionante que la desencadene (Navarro, 2021)
- **Sprites:** De manera resumida, es una imagen o conjunto de imágenes agrupadas en un archivo único, por lo regular representan a los personajes del juego (Navarro, 2021).
- **Tile Set:** Es un conjunto de imágenes, separadas en cuadrículas, que sirven para crear los objetos del juego, por ejemplo, los mapas o mundos (Navarro, 2021).
- **Máscara de colisión:** También conocida como hitbox, sirve para detectar la colisión del Sprite con otras máscaras de colisión (Navarro, 2021).

2.8 Píxel Art

El píxel art es un tipo de estilo artístico que consiste en la creación de gráficos, por lo regular imágenes, basadas en un conjunto de píxeles, de forma que se genere una matriz con estas o, en otras palabras, un mosaico. Este tipo de estilo se ha ocupado desde la década de los 80 y actualmente es uno de los más populares, precisamente porque, a diferencia de los gráficos de alta resolución de los

videojuegos más modernos, el píxel art despierta un sentimiento de nostalgia por su estilo retro.

El píxel art se puede clasificar a partir de los diferentes tipos y características que existen, habiendo 5 principales, siendo estos los siguientes.

- **Píxel art de alta resolución:** Emplea una matriz de 64x64 píxeles para crear imágenes detalladas, precisas y complejas con el fin de hacerlas detalladas, de calidad y con un aspecto retro, un ejemplo de este está en la imagen 4.



Imagen 4 Píxel art de alta resolución (Octo-rl @Octo-rl, s. f.)

- **Píxel art de baja resolución:** Apreciable en la imagen 5, utiliza una matriz de 16x16 o de 32x32 píxeles. A diferencia de la de alta resolución, su paleta de colores es limitada y funciona únicamente para la creación de imágenes con detalles simples pero icónicos.

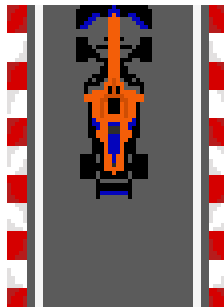


Imagen 5 Píxel art de baja resolución (BrookyLnG @BrookyLnG, s. f.)

- **Píxel art animado:** Funciona para la simulación de movimiento que, por lo regular, se ocupa para hacer GIFs o sprites.

- **Píxel art isométrico:** Ejemplificado en la imagen 6, busca utilizar el 2D y emplearlo de manera que se parezca a un entorno 3D, siendo utilizado por lo regular en animaciones y videojuegos.

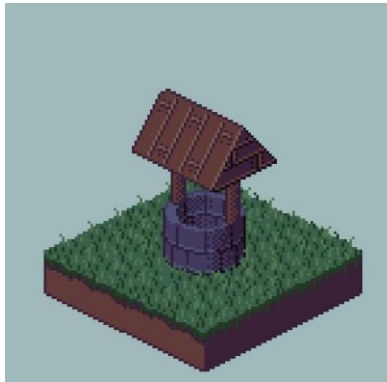


Imagen 6 Píxel art isométrico crabCrabCrab @ crabCrabCrab, s. f.)

- **Píxel art dithering:** Es utilizado para crear sombras y gradientes que generen una percepción de profundidad y de mayor detalle, sin embargo, es por esto por lo que también tiene una paleta de colores limitada, una muestra de este se encuentra en la imagen 7.



Imagen 7 Píxel art dithering (MasterBeany @MasterBeany, s. f.)

2.9 Sprites, spritesheets y fotogramas por segundo

Los sprites, spritesheets y fotogramas por segundo conllevan una fuerte correlación, ya que se necesita de estos tres elementos para poder generar una animación o, más específicamente, el movimiento de un personaje ya sea que se ocupe en una animación o en un videojuego que haga uso del píxel art.

Un Sprite, como se mencionaba anteriormente, no es otra cosa más que una imagen 2D, creada usando píxel art, vector art o raster art, y que se puede ocupar en la escena de un videojuego, de forma que este represente personajes, objetos o incluso efectos. En la imagen 8 se tiene un sprite correspondiente a un personaje popular de los videojuegos, Sonic el erizo.



Imagen 8 Sprite de Sonic el erizo (SauceK, s. f.)

A su vez, un spritesheet u hoja de sprites contiene en su interior un conjunto de sprites referentes a algún objeto específico, de forma que sirven para poder animarlo. Dicha hoja puede llegar a contener distintos “comportamientos” de ese objeto. Por ejemplo, si se habla de un personaje, el spritesheet vinculado a este puede tener sprites para cuando está detenido, para cuando está en movimiento, está agachado, saltando, etc. Un ejemplo de lo anterior se aprecia en la imagen 9, misma que le da animación a Sonic.



Imagen 9 Spritesheet de Sonic el erizo (SauceK, s. f.)

Por último, los fotogramas por segundo, también conocidos como FPS, son la cantidad de imágenes consecutivas que son proyectadas en la pantalla por segundo, es decir, si se reproducen 30 imágenes en un segundo, entonces se tienen 30 FPS, si son 60 imágenes en la misma cantidad de tiempo, entonces son 60 FPS de forma que se crea la ilusión de movimiento. Dicha percepción ocurre a partir de los 12 FPS y, entre mayor sea la cantidad de FPS, más rápido o más fluida ira una animación.

2.10 LaTeX

La Universidad Complutense de Madrid define a esta herramienta como “un sistema de composición tipográfica de alta calidad. Incluye características especialmente diseñadas para la producción de documentación científica y técnica. LaTeX es el estándar para publicación de documentos científicos, pero puede usarse para cualquier tipo de publicación” (UCM-Proyecto de Innovación Software Libre Para Ciencias E Ingenierías, s. f.). En esencia, permite no sólo redactar, sino también poder crear expresiones matemáticas diversas gracias a su conjunto de herramientas, tal como se ve en la imagen 10. Además, hay plataformas que,

basándose en LaTeX, permiten poder exportar esas expresiones a imágenes para poder utilizarlas en otro tipo de trabajos. Una de estas plataformas es Lagrida, que se abordará más adelante.

$$x = 2$$

Imagen 10 Expresión generada con LaTeX y Lagrida

Capítulo 3: Desarrollo de la herramienta interactiva

Para poder llevar a cabo el proyecto, es conveniente saber con cuáles dispositivos se dispone, sobre todo con la potencia que tienen estos para facilitar el diseño y el desarrollo de la herramienta; así como también conocer qué recursos digitales están disponibles para dicho propósito, contemplando ventajas y desventajas. Por esto, en el presente capítulo se consideran aspectos como la infraestructura ocupable, así como también los primeros vestigios en el diseño gráfico del proyecto, como lo son el mapa del sitio, las mecánicas de juego, los bosquejos y el guion detrás de este, así como también el diseño de personajes y sus hojas de movimiento, sin olvidar al código que le da vida a la herramienta interactiva.

3.1 Infraestructura

Tal como se mencionaba al inicio del capítulo, es menester saber qué recursos son los que se encuentran al alcance para poder llevar a cabo el desarrollo de la herramienta, habiendo así identificado 5 recursos digitales y 2 dispositivos, de los cuales, se tratan de una laptop y una tableta electrónica. Por esto, la infraestructura utilizada, de manera detallada, para el desarrollo del proyecto es la siguiente.

- Pixilart
- Visual Studio Code
- Phaser3 (JavaScript)
- Extensión Live Server para Visual Studio Code
- Lagrida (LaTeX)
- ASUS TUF FA506NF con 8 GB de RAM, procesador AMD Ryzen 5 (Ryzen 5 7535HS) serie 7000 de 6 núcleos y 12 hilos de procesamiento, con velocidad base de 3.3 GHz y máxima de 4.55 GHz, SSD M.2 NVME PCIe de cuarta generación de 500 GB, gráficos integrados AMD Radeon 660M y gráficos dedicados NVIDIA RTX 2050.
- Samsung Galaxy Tab S9 FE+ con 12 GB de RAM, procesador Exynos 1380 con velocidad de 2.4 GHz, almacenamiento de 256 GB y gráficos integrados Mali G-68

3.2 Mapa del sitio

Para la creación del mapa del sitio, visible en la imagen 11, se toman en cuenta los temas mencionados en la introducción del presente documento, siendo estos las operaciones con fracciones, cálculo de áreas y perímetros, ecuaciones de primer grado, expresiones algebraicas para sucesiones y las medidas de tendencia central. A su vez, también se incluyen módulos que sirvan de orientación para el usuario, denominados “explicaciones” y también apartados que incluyen los ejercicios relacionados a los contenidos mencionados, dependiendo de donde se esté.



Imagen 11 Mapa del sitio

3.3 Mecánicas, bosquejos y guion

La herramienta interactiva, al pretender incluir elementos de videojuegos para intentar despertar el interés de los usuarios, se apoya de las siguientes mecánicas.

- **Sistema de puntaje:** Si el usuario contesta correctamente un ejercicio, el contador de puntos aumentará en uno, si no, el contador se reiniciará en cero.
- **Sistema de recompensa:** Cuando el jugador consigue determinada cantidad de puntos, aparecerá un mensaje indicando que ha obtenido una medalla. Dependiendo de la cantidad de aciertos será el tipo de medalla obtenida, por ejemplo, cuando se responden correctamente 10 ejercicios, se obtendrá una medalla de bronce, cuando sean 15 ejercicios será una de plata, y a las 20 una de oro.
- **Sistema de puntos de vida:** Mientras el usuario conteste correctamente, no se le restarán puntos de vida, sin embargo, por cada error que este cometa, perderá un punto de vida, hasta que se quede con cero de estos, teniendo la oportunidad de volver a empezar la partida.
- **Sistema de navegación en el sitio:** El usuario podrá desplazarse entre distintas secciones del sitio para poder acceder a distintos temas o ejercicios dentro de la herramienta.
- **Generación de ejercicios aleatorios:** El jugador no siempre se enfrentará a los mismos ejercicios, gracias a esta mecánica. Los problemas aparecerán de manera aleatoria, de esta forma se intenta que el usuario se enfrente a cosas distintas cada vez.
- **Interacción con los personajes:** Para evitar que el usuario se sienta aburrido, este puede interactuar con el personaje de corazón que, cada vez, dará un mensaje diferente, dependiendo de múltiples causas, como, por ejemplo, sus puntos de vida.
- **Sistema de explicaciones por tema:** Diseñado para que el jugador, si lo necesita, pueda orientarse en la resolución de distintos problemas, de modo que la explicación le resulte atractiva.

Para la creación de los personajes se consideran a dos, incluyendo las interacciones que podrían llegar a tener con el usuario, si es que aplica, y en qué contextos. Por ejemplo, el comportamiento y los diálogos del personaje de corazón irán en función de los puntos de vida que el jugador posea en ese momento, mientras que las medallas “guiñarán sus ojos” y “sonreirán” si es que ya fueron desbloqueadas, caso contrario, sólo se mostrará una medalla negra.

De manera adicional, y con el fin de cumplir con el objetivo de que la herramienta sea interactiva y, sobre todo, educativa, se diseña un guion para la adecuada explicación de cada uno de los temas seleccionados, que aparecen en la introducción y en el mapa del sitio, realizado previo desarrollo del software, permitiendo así realizar correcciones y asegurar su adecuada implementación.

Además, similarmente a los bosquejos, el diseño de un guion también permite marcar cuáles elementos son más relevantes, y con eso, que deben ser más llamativos para el usuario. Ambos elementos diseñados se pueden apreciar mejor en la imagen 12.

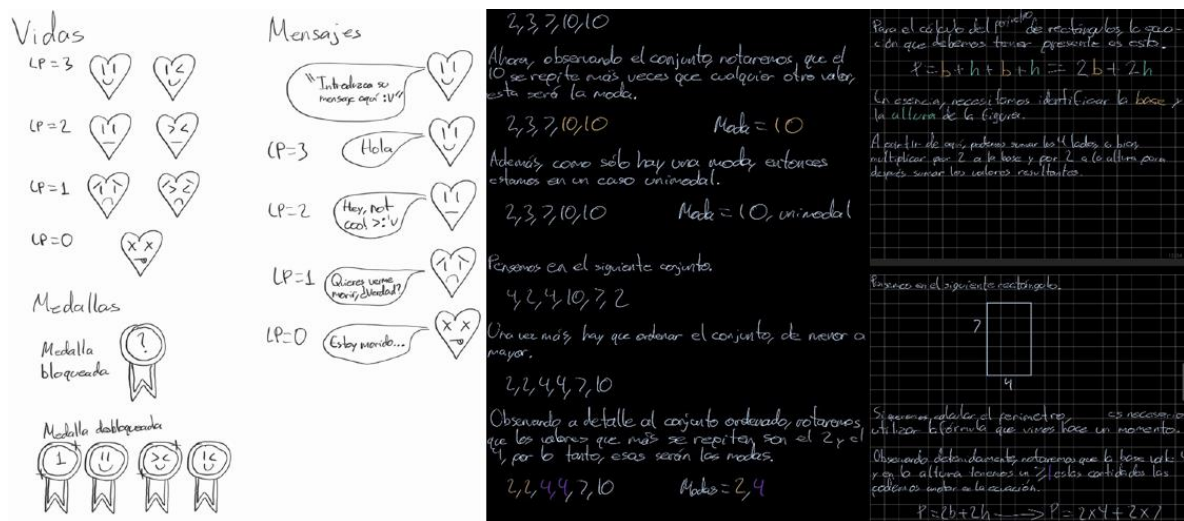


Imagen 12 Guion y bosquejos

3.4 Diseño de sprites y spritesheets

En el proyecto, además de los personajes planteados en el bosquejo, también se contempla el desarrollo de los ejercicios y explicaciones mediante el uso de píxel art o de LaTeX según se requiera. Por ejemplo, para las operaciones con fracciones se ocupó exclusivamente el primero, mientras que para el resto de los módulos se ocupó una mezcla de ambos, siendo esto explicado más a profundidad a continuación.

3.4.1 Pixilart y Lagrida

Pixilart es una herramienta en línea que permite la creación de personajes en píxel art y/o arte de este estilo, cuya pantalla de diseño se aprecia en la imagen 13. Dicha herramienta fue creada para el desarrollo de los personajes de la herramienta, así como de algunas imágenes que se ocupan en los distintos ejercicios y elementos de todos los módulos del proyecto.

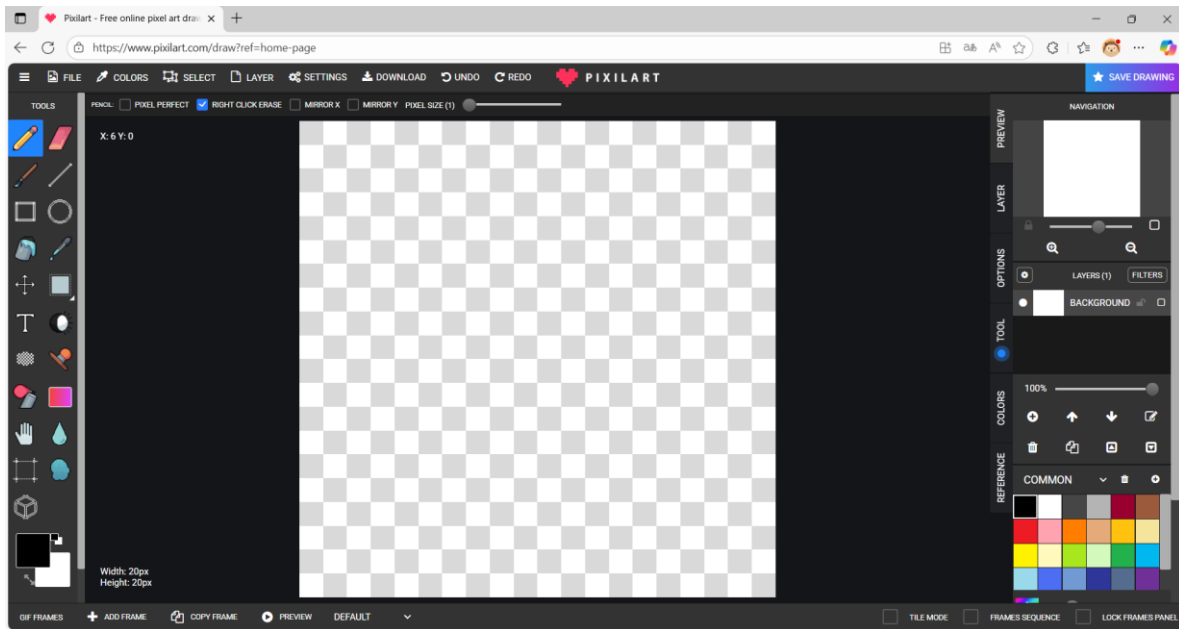


Imagen 13 Pantalla de diseño de Pixilart

Uno de los personajes creados con la herramienta mencionada es precisamente el corazón, que ya se había mencionado en el bosquejo, este siendo visible en la imagen 14. Este representa los puntos de vida del jugador, pudiendo interactuar además con él.



Imagen 14 Personaje de corazón ligado a los puntos de vida del usuario

Como la intención es darle “vida” al personaje, también se creó un spritesheet específico para este, que se aprecia en la imagen 15, de forma que represente todos los estados o “comportamientos” que puede llegar a tener, siendo cuatro, cuando tiene todos sus puntos de vida, siendo tres, este estará feliz; cuando tiene dos puntos mostrará disgusto, un punto hará que se preocupe o tenga miedo y, por último, para cuando “muere” al quedarse sin puntos de vida.



Imagen 15 Spritesheet del personaje de corazón

Además, dependiendo de su estado de ánimo, este puede mostrar distintos mensajes al usuario, por esto, se crea también una burbuja de diálogo, visible en la imagen 16, que contenga lo que el personaje quiera “decir”.



Imagen 16 Burbuja de diálogo del personaje de corazón

Otros personajes diseñados con pixilart son las medallas para generar interés del usuario, siendo estas de bronce, plata y oro. Sin embargo, como sólo deben mostrarse cuando una condición se haya cumplido para cada una de estas, siendo alcanzar determinado puntaje mencionado en las mecánicas, es necesaria la creación de un Sprite que refleje que estás todavía no se han obtenido, creando así el “personaje” de medalla desconocida, mostrada en la imagen 17.



Imagen 17 Sprite de medalla desconocida

De igual forma, a medida que el usuario vaya respondiendo correctamente, aparecerán los sprites de los personajes referentes a las medallas de bronce, plata y oro, que se aprecian en la imagen 18.

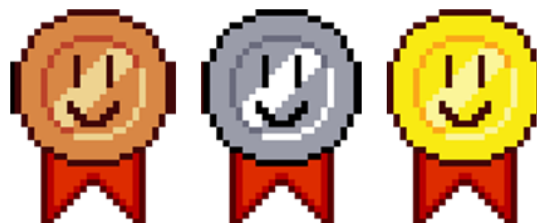


Imagen 18 Sprites de las medallas de bronce, plata y oro

Para despertar el interés del usuario se pretende que las medallas no sean un elemento “estático”, es decir, que las medallas se queden inmóviles, es por esto por lo que se crea un spritesheet para estos personajes con el fin de animarlos y así volverlos llamativos.

En dicha hoja de sprites las medallas muestran un brillo y también sonríen como parpadean, tal como se muestra en la imagen 19.



Imagen 19 Spritesheet de las medallas

Algo que se implementa dentro de la herramienta es la navegación entre pantallas, tal como se mencionaba en el mapa del sitio, es por esto por lo que, manteniendo el estilo píxel art, se crean botones que le permitan regresar a la pantalla anterior o incluso retornar al menú principal, mostrados en la imagen 20.



Imagen 20 botones de regresar a la pantalla anterior y retornar al menú principal

Algo más que se implementa son los mensajes de explicación, que orientan al usuario en el correcto proceso de resolución de problemas, de forma que este pueda navegar libremente a través de dicha pantalla, pudiendo avanzar al siguiente mensaje, retroceder al anterior, omitirlos o, inclusive, volver al inicio de estos. Nuevamente, es por lo anterior por lo que se generan botones estilo píxel art que indiquen cuál es la funcionalidad que van a tener, mismos que son mostrados en la imagen 21.

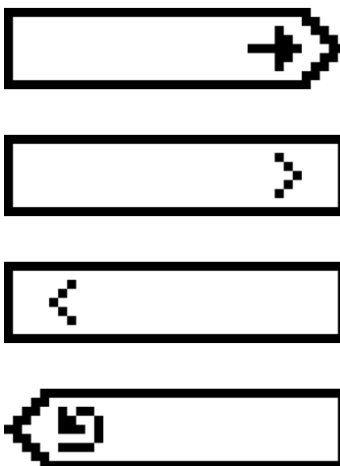


Imagen 21 Botones de navegación en la pantalla de explicación de ejercicios

De igual forma, se utiliza la herramienta de pixilart para lo que son algunos de los elementos que conforman a los ejercicios y a las explicaciones, tal como se muestra en las imágenes 22, 23, 24 y 25.



$$\begin{array}{r} 14 + 15 \\ \hline 29 \end{array}$$

Imagen 22 Sprite de una de las explicaciones del módulo de operaciones con fracciones

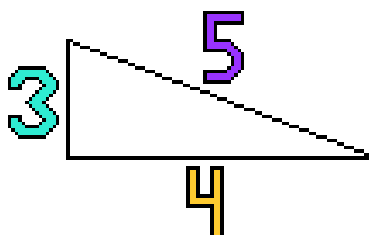


Imagen 23 Sprite de una de las explicaciones del módulo de cálculo de áreas y perímetros



$$3, 6, 9, 12, \dots$$

Imagen 24 Sprite de uno de los ejercicios del módulo de expresiones algebraicas para sucesiones



$$\{2, 3, 7, 10, 10\}$$

Imagen 25 Sprite de una de las explicaciones del módulo de medidas de tendencia central

Sin embargo, no todo se puede diseñar con píxel art, o puede que incluso no sea conveniente, sobre todo si representa una alta cantidad de tiempo en su creación o es algo complejo de realizar, es por esto por lo que se recurre a otro tipo de herramientas, como es el caso de Lagrida, cuya pantalla de diseño se aprecia en la imagen 26, que es un editor de ecuaciones en línea basado en LaTeX y que tiene por bondad poder exportar expresiones matemáticas a formato PNG, como las que se mostrarán a continuación, permitiendo así crear contenido más preciso y, además, acelerar los tiempos de diseño. Otra de las ventajas de Lagrida es que permite que las imágenes generadas no tengan fondo, lo cual no interfiere de manera estética con el resto de los elementos gráficos de la herramienta.

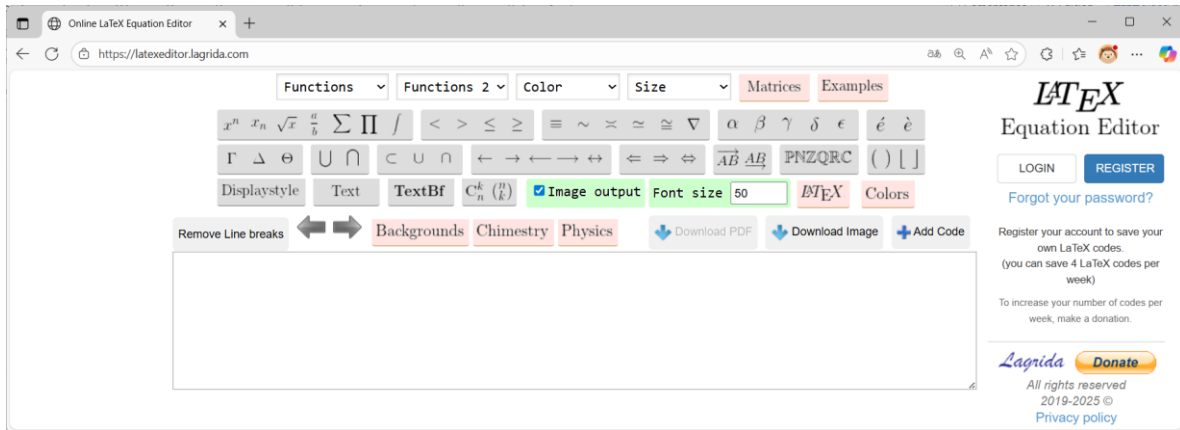


Imagen 26 Pantalla de diseño de Lagrida

Lagrida, como se explicó con anterioridad, se ocupó para el diseño de expresiones matemáticas que le abonen a la creación de la herramienta, tal como las que se muestran en las imágenes 27 y 28.

$$A = L \times L$$

Imagen 27 Sprite de una de las explicaciones del módulo de cálculo de áreas y perímetros

$$a_n = dn + k$$

Imagen 28 Sprite de una de las explicaciones del módulo de expresiones algebraicas para sucesiones

3.4.2 GammaFP, Atlas Packer y Animator Tool

GammaFP, creada por el desarrollador web Francisco Pereira Alvarado, es un sitio web con herramientas gratuitas que permiten el diseño de múltiples elementos que se pueden importar con Phaser3 para el diseño de videojuegos o similares. La interfaz de inicio de dicha web se aprecia en la imagen 29.

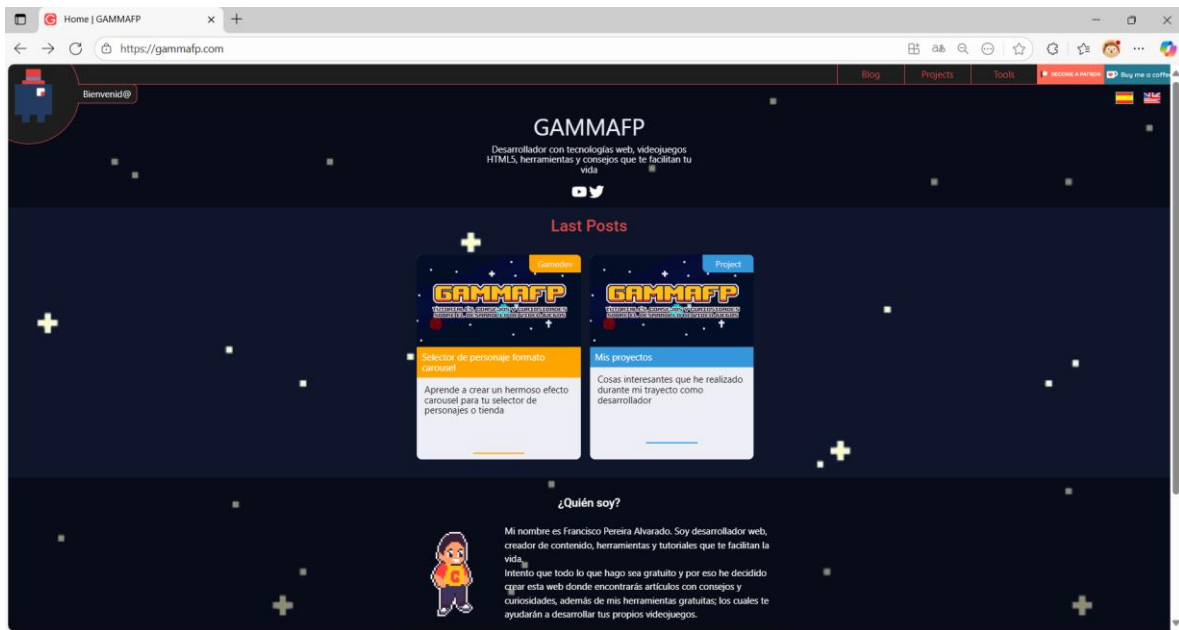


Imagen 29 Pantalla de inicio de GammaFP

Dentro de estas podemos notar una serie de herramientas, visibles en la imagen 30, como Easy Resizer, que permite reescalar a una imagen individual o a un conjunto de estas; Atlas Packer que, a partir, de un conjunto de sprites, genera un spritesheet; Animator Tool, cuya función es generar animaciones a partir de una hoja de sprites; Audio Sprite, que también crea un atlas de audios; etc.

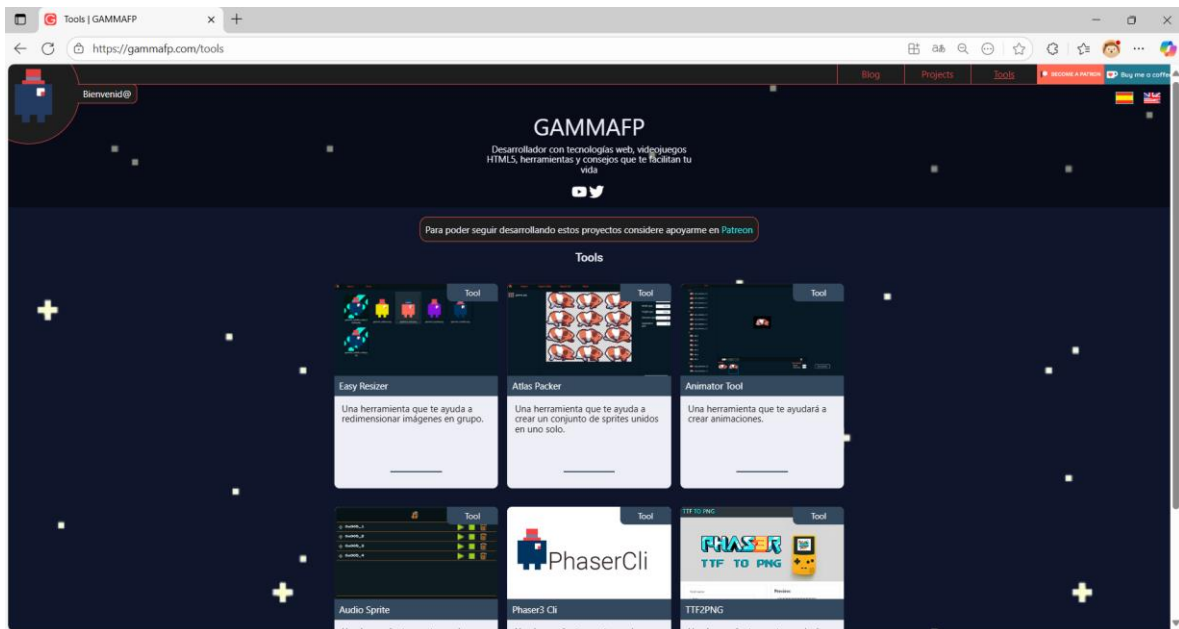


Imagen 30 Herramientas de diseño y animación de GammaFP

Para la animación de los distintos personajes, se ocupa Animator Tool, partiendo de sus respectivos spritesheets. En la imagen 31 se puede apreciar al personaje de corazón junto con los distintos sprites que lo conforman, y adicionalmente, cuatro animaciones que representan el comportamiento de este, tal como se describió en el bosquejo.

Otra cosa que también aparece en la herramienta es el control de FPS y repetición de la animación, de forma que, con el primero se puede especificar la velocidad de esta, y con el segundo, indicar cuántas veces se repetirá la animación, representando -1 un ciclo infinito.

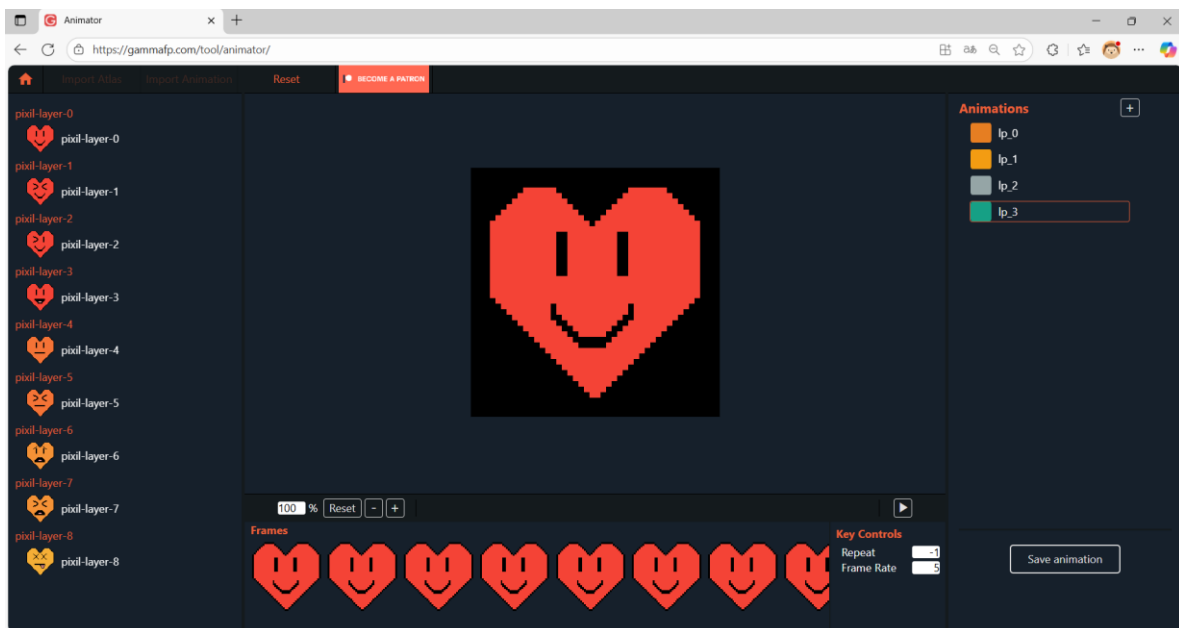


Imagen 31 Animación del personaje de corazón

3.5 Código

Un proyecto de Phaser3 consta de dos archivos fundamentales para su funcionamiento, sin contar los correspondientes a las escenas del juego, siendo estos "index.html" y "main.js". En el primero se define la estructura de la página web y, sobre todo, un contenedor div en la que se mostrará todo lo generado por Phaser3, es decir, todo el juego, lo anterior mostrado en la ilustración 1.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Módulo 1 (Tesis)</title>
  <script src="/node_modules/phaser/dist/phaser.min.js"></script>
```

```

    <script src="./src/main.js" type="module"></script>
</head>
<body>
    <div id="contenedor"></div>
</body>
</html>

```

Ilustración 1 Estructura de "index.html"

En el segundo, visible en la ilustración 2, se definen múltiples parámetros del proyecto, desde el nombre del juego hasta las escenas que la conforman en orden de prioridad, pasando por el ancho y alto de la pantalla de juego.

```

const config = {
  title: "", //Nombre del juego (opcional)
  url: "", //Dirección de la página del juego (opcional)
  version: "0.0.1", //Versión alfanumérica (opcional)
  type: Phaser.AUTO, //Tipo de renderizado (WEBGL, CANVAS,
  // AUTO: busca primero WEBGL y si no está
  // disponible eligirá CANVAS
  width: 1500, //Ancho de pantalla del juego
  height: 730, //Alto de pantalla del juego
  parent: "contenedor", //Nombre del id del elemento <div> en el
  index.html // se refiere a dónde se pondrá el canvas
  o lienzo
  pixelArt: true, //Diseño con pixeles definidos (no borrosos)
  backgroundColor: "#34495e", //Color de fondo del canvas ()
  scene: [
    //Escenas del juego
  ]
}

```

Ilustración 2 Estructura de "main.html"

Tal como se muestra en la ilustración 3, cada escena de Phaser3 debe heredar de `Phaser.Scene` para poder implementar las siguientes cinco funciones principales, siendo su constructor, en el que se define el nombre de esta; `init()`, encargada del intercambio de datos entre escenas; `preload()`, cuyo fin es la carga de recursos multimedia; `create()`, para la generación de los recursos que se ocuparán a lo largo del juego; y, por último, `update()` que actualiza los cambios que se produzcan a lo largo del juego dependiendo de determinadas acciones.

```

class Menu_principal extends Phaser.Scene{
  constructor(){
    super({

```

```

        });
    }

    init(){

    }

    preload(){

    }

    create(){

    }

    update(time, delta){

    }
}

export default Menu_principal;

```

Ilustración 3 Estructura de una escena heredando de Phaser.Scene

3.5.1 Carga de elementos gráficos

Tal como se mencionaba, para poder realizar la carga de los recursos que se usarán en el proyecto, es necesario ocupar la función `preload()` de Phaser3, apreciable en la ilustración 4, de forma que, primero, hay que definir dónde se encuentran estos elementos, para ello, se ocupa la línea `this.load.path` y, seguidamente, la ruta de ubicación de los archivos.

De manera similar, para poder cargar los sprites generados con anterioridad, o los assets, es necesario del uso de `this.load.image` que permite la carga de una imagen con un identificador único, o también la carga de un arreglo de estas, ahorrando tiempo y líneas de código.

```

preload(){
    this.load.path = './assets/';
    this.load.image(["siete_quintos", "tres_medios", "mas_positivo",
"menos_negativo", "multiplicacion", "division", "igual", "Op_suma",
"Res_suma", "cinco_sexto", "diez_sexto", "Op_suma2", "Res_suma2",
"Op_resta", "Res_resta", "Op_resta2", "Res_resta2", "Op_multi", "Res_multi",
//Resto de elementos gráficos...

```

```
]);
```

Ilustración 4 Carga de elementos gráficos con la función preLoad()

Otras funciones fundamentales para indicarle a Phaser3 cómo es que debe interpretar la hoja de sprites son `this.load.atlas` y `this.load.animation`, mostradas en la ilustración 5, la primera para que este identifique la posición de cada Sprite dentro de la spritesheet, y la segunda para que sepa cómo realizar correctamente las animaciones tomando como base a los propios sprites.

```
this.load.atlas("life_point","life_point.png","life_point_atlas.json");
    this.load.animation("life_point_anim","life_point_anim.json");
    this.load.atlas("medallas_ejercicios","medallas_ejercicios.png","medallas_ejercicios_atlas.json");
    this.load.animation("medallas_ejercicios_anim","medallas_ejercicios_anim.json");
}
```

Ilustración 5 Estructura de this.load.atlas() y this.load.animation()

3.5.2 Text e Image

Una vez cargados los recursos multimedia, es necesario crearlos para que Phaser3 pueda hacer uso de ellos, es por esto por lo que ahora se ocupa la función `create()`.

De entrada, como el usuario interactuará con el software a través del mouse, es necesario ocupar la línea `Phaser.Input.Events`, mostrada en la ilustración 6, que sirve para interpretar las acciones realizadas con este, desde que el puntero este sobre un objeto de juego, fuera de este, el usuario haga clic en uno, lo arrastre, etc. Esto es especialmente útil ya que se requiere que el usuario sea capaz de hacer clic en texto para acceder a los distintos módulos del programa, así como para, cuando interactúa con imágenes, también se desencadene cierta acción.

```
const eventos = Phaser.Input.Events;
```

Ilustración 6 Phaser.Input.Events

Para añadir texto a un juego, así como al proyecto en cuestión, se utiliza la instrucción `this.add.text`, preferentemente ligándola a una variable para su posterior manipulación, de forma que esta requiere, como mínimo, tres parámetros, siendo estos la coordenada "x", la coordenada "y" y el texto que se mostrará en pantalla. Un parámetro extra, que es opcional, es el estilo de letra, que contiene la familia a la que pertenece y también el tamaño que tendrá al mostrarse en pantalla. Como extra, para que Phaser3 sepa que el usuario puede interactuar con el texto,

es indispensable ocupar la línea `.setInteractive()` una vez escrita la función anterior.

Por ejemplo, en la ilustración 7 podemos ver que el usuario, aunque podrá ver el texto contenido en la variable `this.menuPrincipal`, no podrá hacerle clic, ni seleccionarlo o siquiera poder interactuar con este, caso contrario con `this.opFracciones`, todo gracias a la función `.setInteractive()`.

```
this.menuPrincipal = this.add.text(0,10,"¡Selecciona uno de los siguientes temas para estudiar dando clic!",  
    {fontFamily: "sans-serif", fontSize: 30});  
  
    this.opFracciones = this.add.text(0,50,"1.- Operaciones con fracciones",  
    {fontFamily: "sans-serif", fontSize: 30}).setInteractive();
```

Ilustración 7 Declaración de elementos interactivos y no interactivos en Phaser3

Cuando se trata de una imagen, pero no de un personaje, cosa que se verá más adelante, se ocupa `this.add.image`, que exige un mínimo de tres parámetros, siendo la coordenada en “x”, la coordenada en “y” y el identificador de la imagen, mismo que se empleó en la carga de elementos gráficos dentro de la función `preload()`.

De manera similar con el uso del texto, para que el usuario pueda interactuar con las imágenes de cualquier forma, es necesario utilizar la función `.setInteractive` al final, con ello el objeto se vuelve interactuable, tal como se aprecia en la ilustración 8.

```
this.ejercicio1 = this.add.image(750,200,"AreaPerimetroEj1");  
this.ejercicio1a = this.add.image(180,560,"AreaEj1RA").setInteractive();  
this.ejercicio1b = this.add.image(750,560,"AreaEj1RB").setInteractive();  
this.ejercicio1c = this.add.image(1320,560,"AreaEj1RC").setInteractive();
```

Ilustración 8 Interacción con el texto de la herramienta

3.5.3 Alpha, setAlpha, Visible y setVisible

Parte de los atributos de las imágenes son la visibilidad y transparencia, siendo estos el “visible” y el “alpha” respectivamente. Estos atributos, visibles en la ilustración 9, son manipulados a través de Phaser con las funciones `.setVisible()` y `setAlpha()` permitiendo que los elementos de juego sean utilizados a

conveniencia, es decir, que se muestren dentro de la escena cuando sea necesario y, cuando ya no sean requeridos, simplemente no aparezcan, además, con el alpha, se puede generar un desvanecimiento progresivo, yendo desde cero, completamente invisible, hasta uno, sin transparencia, mientras que, con la visibilidad, sólo se muestra o no.

```
this.contenedorEx2.setVisible(false);  
this.contenedorEx2.setAlpha(1);
```

Ilustración 9 Visibilidad y Alpha de un elemento gráfico

3.5.4 Tweens y Timelines

Para poder producir animaciones más allá de los spritesheets, se recurre regularmente a los tweens y timelines de forma que permiten manipular múltiples atributos de los objetos de juego, desde el tamaño, hasta la transparencia pasando por el movimiento. En cuestión a este proyecto, para darle un poco de “movilidad”, se aplican en los ejercicios, de forma que, al responder correcta o incorrectamente, se genera un desvanecimiento para dar paso a la siguiente pregunta. Esto, a través de código, se aprecia mejor en la ilustración 10.

```
this.animCont = this.tweens.timeline({  
  tweens: [  
    {  
      delay: 1000,  
      targets: [this.contenedorEx4, this.mensajeAcierto],  
      alpha: 0,  
    },  
  ],  
  onComplete: () => {  
    this.limpiaTinta();  
    this.contenedorEx4.setVisible(false);  
    this.contenedorEx4.setAlpha(1);  
    this.arrayContenedores.at(this.selector).setVisible(true);  
    this.bandEjercicios = true;  
  }  
});
```

Ilustración 10 Animaciones con tweens

3.5.5 Manejo de sprites

El manejo de sprites se realiza de una manera similar al de las imágenes, con una ligera variación, en vez de utilizar `this.add.image()` se emplea `this.add.sprite()` para hacerle saber a Phaser3 que vamos a trabajar con un

personaje y no con un objeto “estático”. De igual manera, requiere de tres parámetros, siendo estos las componentes “x” e “y” de las coordenadas en las cuales se ubicará el sprite, así como también el identificador que tenga asignado, mismo que obtuvo en la carga de elementos gráficos de la función `preload()`.

Adicionalmente, se le pueden añadir funciones extra al sprite, como lo son `.setScale()` que permite ajustar el tamaño de este según se necesite, y también `.setInteractive()` para hacer del sprite un objeto interactuable.

Una diferencia sustancial de los sprites con respecto a las imágenes está en que los primeros permiten seleccionar desde qué fotograma empezar, así como cuál animación ejecutar desde un inicio, así como, dependiendo de los eventos, desencadenar una u otra animación, para ello, es necesario añadirle a la variable que contenga al sprite la función `.animations.play()` y, como único parámetro, el identificador de la animación a mostrar, mismo que se asignó al momento de crearlas con la herramienta “animator tool” de GammaFP. Además de reproducir, un sprite también cuenta con otras funciones para controlar la reproducción de la animación, como lo es pausarlo, detenerlo, repetirlo o retrasarlo.

En la ilustración 11 se muestra la adición del personaje de corazón a la escena, en la coordenada (1330,120) ejecutando una animación identificada como `lp_3` que representa cuando este tiene su salud completa.

```
this.life_point =  
this.add.sprite(1330,120,"life_point").setScale(0.2).setInteractive();  
this.life_point.animations.play("lp_3");
```

Ilustración 11 Adición del personaje de corazón a través de código

3.5.6 Lógica de las mecánicas

Para las mecánicas, que le dan vida a la herramienta, se toman en cuenta los siguientes aspectos:

- Ejecución de nuevas escenas
- Desplazamiento en las explicaciones
- Generación aleatoria de los ejercicios
- Puntos por ejercicio
- Puntaje máximo
- Puntos de vida
- Interacción con el personaje de corazón
- Desbloqueo de las medallas de bronce, plata y oro
- Reinicio de los datos de la aplicación

De entrada, es necesario recordar la estructura de la aplicación, mostrada en el mapa del sitio, de forma que se entiende que esta está compuesta por 42 escenas interconectadas.

Para poder acceder a nuevas escenas se aprovecha que se puede interactuar con el texto, de forma que, al tener el cursor sobre este, el primero cambia de color, y al hacer clic sobre el texto, se inicia la escena asociada a este, tal como se observa en la imagen 32.

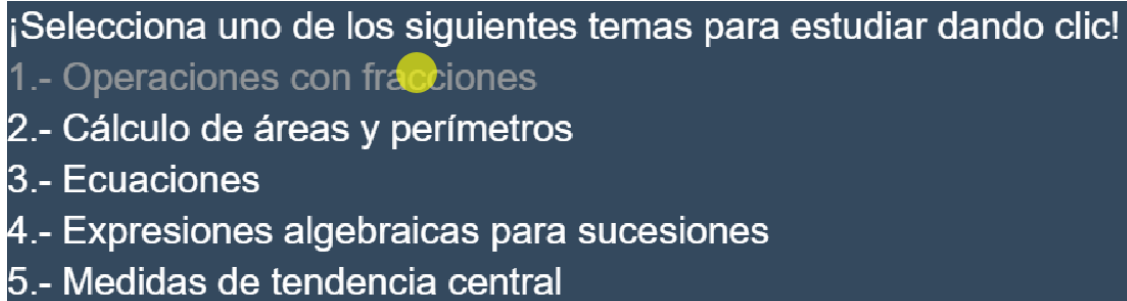


Imagen 32 Interacción con los elementos de texto de la herramienta

Esto es gracias a que Phaser3 registra eventos del mouse, recurriendo a la línea `Phaser.Input.Events`, explicada en el apartado “Text e Image” del presente documento, siendo estos cuando el cursor se encuentra encima de, o que se pulsó un objeto de juego, siendo este el primero.

En la ilustración 12, la lógica para hacer que el texto cambie de color, en función del comportamiento del mouse, depende de las líneas eventos.`GAMEOBJECT_OVER` y eventos.`GAMEOBJECT_OUT` que, en resumen, detectan cuando el cursor está encima de un objeto interactivo y cuando el puntero sale de este. Como extra, si se está en la primera situación, el texto cambia de color a un tono más grisáceo, tal como en la imagen anterior, caso contrario, el texto vuelve a su tonalidad original.

```
this.input.on(eventos.GAMEOBJECT_OVER, (pointer, gameObject) => {  
    gameObject.setTint(0x909497);  
});  
  
this.input.on(eventos.GAMEOBJECT_OUT, (pointer, gameObject) => {  
    gameObject.clearTint();  
});
```

Ilustración 12 Cambio de color del texto de la herramienta a través de código

Entonces, cuando se pulsa el texto relacionado a un módulo específico, se inicia la siguiente escena, esto con las líneas eventos.`POINTER_DOWN` que registra cuando se ha hecho clic a un objeto de juego específico y `this.scene.start()` que ejecuta la escena seleccionada como parámetro.

```
this.opFracciones.on(eventos.POINTER_DOWN, () => {
```

```
this.scene.start('Menu_op_fracciones');
});
```

Ilustración 13 Arranque de escenas desde código

En caso de que, por error, se seleccione una escena incorrecta o que por voluntad del propio usuario se desee regresar a otra, se disponen de los botones de retorno a la pantalla anterior y retorno al menú principal, mostrados en la imagen 20. Dichos botones aparecen en todas las escenas de la herramienta, excepto en la pantalla principal.

Para las escenas que contienen las explicaciones se implementan, una vez más, imágenes y textos para que el usuario tenga una mayor noción acerca de lo que se está mostrando, dándole la posibilidad de poder controlarlas mediante botones que aparecen en la pantalla, tal como se muestra en la imagen 33.

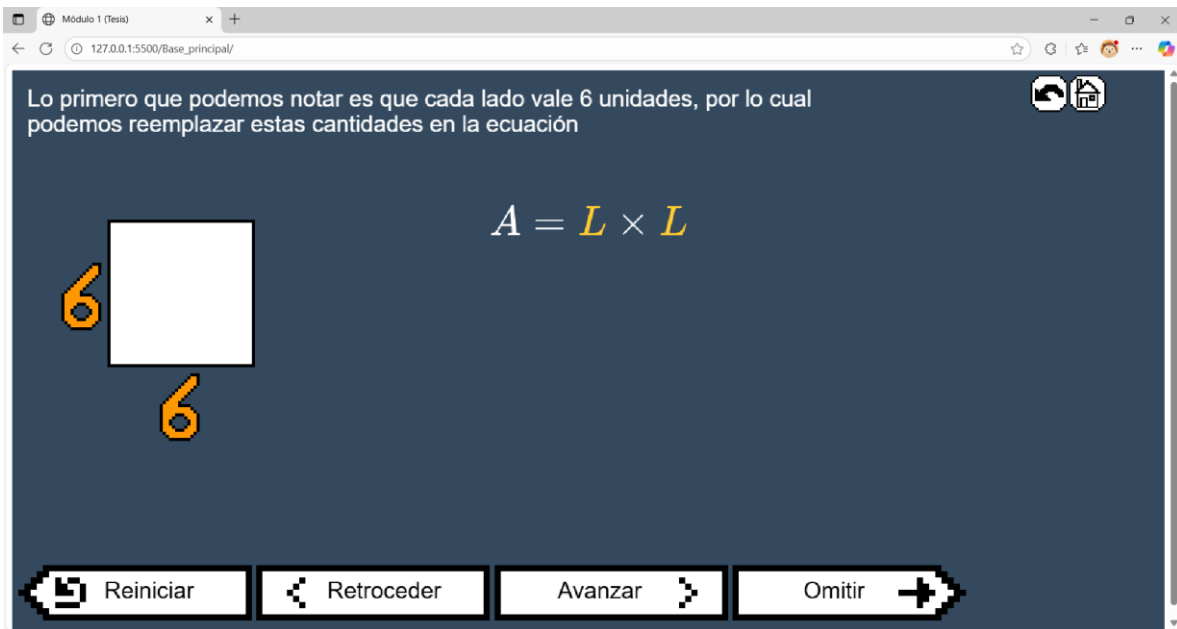


Imagen 33 Interfaz de la herramienta, con los botones de control para la explicación de ejercicios

En cuestión al código, para lograr que la pantalla de explicaciones funcione tal y como se espera, se trabaja con una serie de textos, denominados mensajes, que contienen el paso a paso en la resolución de ejercicios, todos dentro de contenedores para su fácil manipulación vía código, esto siendo apreciable en las ilustraciones 14, 15 y 16.

```
//Mensajes para la explicación
this.mensaje1 = this.add.text(20,20,"Hola, se bienvenido o bienvenida
al módulo de cálculo de áreas y perímetros,\n"
```

```

        +"en el presente aprenderás a calcular áreas de diferentes figuras
geométricas básicas.\n"
        +";Comencemos!",
        {fontFamily: "sans-serif", fontSize: 30});

    this.mensaje2 = this.add.text(20,20,"Primero, debemos tener en cuenta
de que cada figura tiene su propia\n"+
        "fórmula o ecuación que permite calcular el área de esta",
        {fontFamily: "sans-serif", fontSize: 30});

    this.mensaje3 = this.add.text(20,20,"Empezando con el cuadrado, la
fórmula que permite calcular\n"+
        "su área es esta",
        {fontFamily: "sans-serif", fontSize: 30});

```

Ilustración 14 Mensajes para la explicación de ejercicios vía código

```

this.contenedorExpP1.add([
    this.mensaje1
]);
this.contenedorExpP2.add([
    this.mensaje2
]);
this.contenedorExpP3.add([
    this.mensaje3
]);
this.contenedorExpP4.add([
    this.mensaje4
]);

```

Ilustración 15 Contenedores de mensajes para la explicación de ejercicios vía código

```

this.arrayExp = [this.contenedorExpP1, this.contenedorExpP2,
this.contenedorExpP3, this.contenedorExpP4, this.contenedorExpP5,
this.contenedorExpP6, this.contenedorExpP7, this.contenedorExpP8,
this.contenedorExpP9, this.contenedorExpP10, this.contenedorExpP11,
this.contenedorExpP12, //Resto de contenedores...];

this.indexExp = 0;

```

Ilustración 16 Arreglo de contenedores de mensajes para la explicación de ejercicios vía código

Además, cada uno de los últimos constituyen un arreglo único para que, dependiendo de las acciones del usuario con los botones, se muestre el mensaje que se encuentre en la posición “i”, ocultando todo lo demás, es decir, se trabaja con el índice del array, tal que el botón “reiniciar” muestra el contenido de la posición 0, “retroceder” la posición “i-1”, “avanzar” la posición “i+1” y “omitir” muestra la

pantalla para acceder a los ejercicios relacionados al tema, ejemplificado en la ilustración 17.

```
this.botonAvanzar.on(eventos.POINTER_DOWN, () => {
    this.botonRetroceder.setVisible(true);
    this.mensajeRetroceder.setVisible(true);
    this.botonReiniciar.setVisible(true);
    this.mensajeReiniciar.setVisible(true);
    this.arrayExp[this.indexExp].setAlpha(0);
    this.indexExp=this.indexExp+1;
    this.arrayExp[this.indexExp].setAlpha(1);
    if(this.indexExp==this.arrayExp.length-1){
        this.botonAvanzar.setVisible(false);
        this.mensajeAvanzar.setVisible(false);
        this.botonOmitir.setVisible(false);
        this.mensajeOmitir.setVisible(false);
    }
    this.condicionalesExp();
});
```

Ilustración 17 Manejo de los contenedores de mensajes desde código

Sin embargo, para la implementación con imágenes que apoyen las explicaciones, el funcionamiento es distinto, esto es porque puede que se requiera que se siga mostrando una específica de estas, pero también cambiar de texto, por ello, las primeras no se pueden incluir en el arreglo de contenedores de las últimas. Es por esto por lo que, empleando un dato que sí se conoce, siendo este el valor del índice actual del array, se utilizan condicionales, esto permite que, dependiendo de la posición del arreglo, se mantengan, o no, ciertas imágenes, pudiendo así cambiar libremente el texto, según se requiera.

```
condicionalesExp(){

    if(this.indexExp>=2&&this.indexExp<4){
        this.ecuacionAreaCuadrado.setVisible(true);
    }else{
        this.ecuacionAreaCuadrado.setVisible(false);
    }
    if(this.indexExp>=4&&this.indexExp<6){
        this.ejemploArea1.setVisible(true);
    }else{
        this.ejemploArea1.setVisible(false);
    }
}
```

Ilustración 18 Condicionales para la aparición de imágenes con relación a la explicación de ejercicios

En la escena de los ejercicios, para evitar caer en un ciclo repetitivo en el cual aparecen los ejercicios en un determinado orden, nuevamente, se emplea un arreglo en el cual se almacenan estos junto con sus respuestas correctas e incorrectas ligadas, de forma que, mediante que la función `Math.random()` y `Math.floor()` se generan valores aleatorios para el índice, tal que sólo se muestra el contenido de la cantidad obtenida, esto mostrado en la ilustración 19.

```
this.arrayContenedores = [this.contenedorEx1, this.contenedorEx2,
this.contenedorEx3, this.contenedorEx4];

this.selector = Math.floor(Math.random()*this.arrayContenedores.length);
this.arrayContenedores.at(this.selector).setVisible(true);
```

Ilustración 19 Aleatorización de los ejercicios de la herramienta mediante código

Sin embargo, también puede ocurrir un fenómeno en el cual, el próximo ejercicio sea el mismo que el presente, para evitar esto, se utiliza un arreglo más, siendo este dinámico, de forma que contenga todos y cada uno de los valores aleatorios generados, muestra de esto está en la ilustración 20.

```
this.indicesUsados.push(this.selector);
```

Ilustración 20 Guardado de los índices utilizados en la aleatorización de ejercicios a través de código

Así, en la ilustración 21, se produce el cálculo de una cantidad aleatoria hasta que el valor generado no se encuentre en el arreglo previamente mencionado.

```
do {
    this.selector =
Math.floor(Math.random()*this.arrayContenedores.length);
}while(this.indicesUsados.includes(this.selector));

this.indicesUsados.push(this.selector);
```

Ilustración 21 Cálculo de índices utilizados en la aleatorización de ejercicios a través de código

Para evitar que este se llene, también se evalúa cuando el arreglo de índices ya tiene el mismo tamaño que el arreglo de ejercicios y, en caso de ser así, se vacía el primero dejando únicamente el último valor almacenado en este, tal como se muestra en la ilustración 22.

```
if (this.indicesUsados.length==this.arrayContenedores.length){
    this.indicesUsados.splice(0, this.indicesUsados.length-1);
}
```

Ilustración 22 Vaciado del arreglo de índices utilizados a través de código

En la ilustración 23 se observa que, para el cálculo de puntajes se ocupa una variable de clase, llamada `this.puntaje` que, dependiendo de los aciertos del

usuario, esta irá incrementando en una unidad. Además, para evitar que el usuario pueda hacer múltiples clics en la respuesta correcta y el contador aumente, se utiliza una bandera para que, hasta que acabe la animación, sólo se contabilice una vez y ya. El puntaje se muestra, además, en la interfaz de la herramienta, tal como se observa en la imagen 34.

```
console.log("Correcto");
this.mensajeError.setAlpha(0);
this.mensajeAcierto.setAlpha(1);
if(this.bandEjercicios){
    this.puntaje++;
    this.puntajes();
    this.bandEjercicios = false;
}
```

Ilustración 23 Cálculo del puntaje del usuario a través de código

Para poder determinar el puntaje máximo del usuario, se implementa la función `puntajes()` que, en su interior, calcula el puntaje actual con el puntaje máximo registrado y, si es mayor, se almacena de manera local con `localStorage.setItem()` y también se muestra este valor en la pantalla, esto está ejemplificado con código en la ilustración 24 y de manera gráfica en la imagen 34.

```
if(this.puntaje>this.puntajeMax){
    this.puntajeMax=this.puntaje;
    localStorage.setItem("pMaxArea",this.puntajeMax);
    this.mensajePuntajeMax.setText("Máximo: "+this.puntajeMax);
}
```

Ilustración 24 Cálculo del puntaje máximo del usuario a través de código

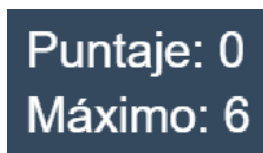
Un recuadro azul oscuro con texto blanco que muestra 'Puntaje: 0' y 'Máximo: 6'.

Imagen 34 Puntaje y puntaje máximo del usuario

En caso de que el usuario escoja una respuesta de manera equivocada, entonces `this.puntaje` regresa a cero, dándole entrada a la función `this.accionErrores()`, esto se muestra con código en la ilustración 25.

```
accionErrores(){
    console.log("Incorrecto...");
    this.mensajeError.setAlpha(1);
    this.puntaje = 0;
    this.puntajes();
    this.LPrestantes();
}
```

Ilustración 25 Reinicio del puntaje del usuario en caso de error a través de código

La función anterior, además, invoca una vez más a la función `puntajes()` exclusivamente para la actualización en pantalla del puntaje del usuario, y también a `LPrestantes()` para la interacción con los puntos de vida.

Entonces, observando la ilustración 26, en caso de un error, se reduce en una unidad los puntos de vida del usuario, almacenados en la variable `this.LP`, posteriormente, se actualizan en la pantalla y, dependiendo de los que le queden, o no, al usuario, se desencadena una animación del personaje de corazón, tal como aparece en la imagen 14 que representa su spritesheet.

```
LPrestantes(){
    this.LP--;
    this.mensajeLP.setText(" = "+this.LP);
    if(this.LP == 2){
        this.life_point.ans.play("lp_2");
        console.log(this.life_point.ans.getName());
        this.timeline3LP.pause();
        this.timeline2LP.play();
    }
    if(this.LP == 1){
        this.life_point.ans.play("lp_1");
        console.log(this.life_point.ans.getName());
        this.timeline2LP.pause();
        this.timeline1LP.play();
    }
    if(this.LP == 0){
        this.life_point.ans.play("lp_0");
        console.log(this.life_point.ans.getName());
        this.timeline1LP.pause();
        this.mensajeLP.setText(" = ...");
        console.log("Muerte...");
        this.arrayContenedores.at(this.selector).setVisible(false);
        this.mensajeMuerte.setVisible(true);
        this.si.setVisible(true);
        this.no.setVisible(true);
        this.mensajeError.setAlpha(0);
    }
}
```

Ilustración 26 Reducción de los puntos de vida del usuario y reproducción de animaciones del personaje de corazón a través de código

Adicionalmente, y como se mencionó con anterioridad en el apartado 3.5.1 del presente documento, el personaje corazón no solamente tiene la función de

representar los puntos de vida del usuario, sino que también se puede interactuar con este. Lo único que tiene que hacer el usuario es poner el cursor encima y este reaccionará. Eso es ejemplificado en la imagen 35.



Imagen 35 Interacción con el personaje de corazón con todos los puntos de vida

Este personaje tiene dos mensajes por punto de vida, excepto cuando ya no tiene, en ese caso sólo se muestra un mensaje. En la imagen 36 se contempla que, además de que el personaje de corazón está preocupado, también cambia el texto el texto que aparece cuando el usuario interactúa con este.



Imagen 36 Interacción con el personaje de corazón con un solo punto de vida

Lo anterior se logra teniendo una bandera de estado que permite cambiar el mensaje visto, además de la evaluación de los puntos de vida con una condicional para desencadenar una u otra animación, esto mostrado en la ilustración 27.

```
this.life_point.on(eventos.POINTER_OVER, () => {  
    this.cuadroTexto.setVisible(true);
```

```

this.mensajeBronce.setVisible(false);
this.mensajePlata.setVisible(false);
this.mensajeOro.setVisible(false);
if(this.bandEstadoLifePoint == 0){
    this.bandEstadoLifePoint = 1;
    if(this.life_point.ans.getNombre()=="lp_3"){
        this.mensajeLifePoint3A.setVisible(true);
    }
    if(this.life_point.ans.getNombre()=="lp_2"){
        this.mensajeLifePoint2A.setVisible(true);
    }
    if(this.life_point.ans.getNombre()=="lp_1"){
        this.mensajeLifePoint1A.setVisible(true);
    }
    if(this.life_point.ans.getNombre()=="lp_0"){
        this.mensajeLifePoint0.setVisible(true);
    }
}
}

```

Ilustración 27 Desencadenantes de las animaciones del personaje de corazón en función de los puntos de vida del usuario

En la imagen 37 se tiene que el personaje de corazón, a su vez, también le notifica al usuario en cuanto su puntaje máximo es lo suficientemente alto como para desbloquear alguna de las tres medallas, mencionadas igualmente en el apartado 3.5.1.

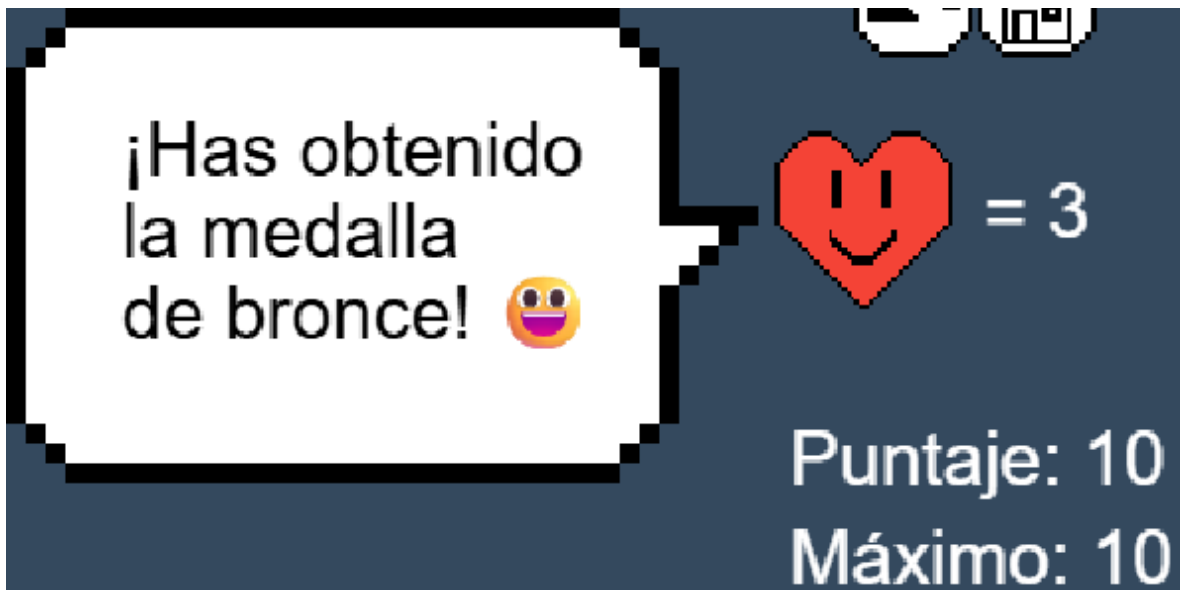


Imagen 37 Notificación de desbloqueo de medalla por parte del personaje de corazón

Una vez más, con la función `puntajes()` también se evalúa el puntaje máximo y, dependiendo de este, se muestra uno u otro mensaje. Esto se puede observar en la ilustración 28.

```
if(this.puntajeMax==10){
    this.cuadroTexto.setVisible(true);
    this.mensajeBronce.setVisible(true);
}
if(this.puntajeMax==15){
    this.cuadroTexto.setVisible(true);
    this.mensajePlata.setVisible(true);
}
if(this.puntajeMax==20){
    this.cuadroTexto.setVisible(true);
    this.mensajeOro.setVisible(true);
}
```

Ilustración 28 Notificación de desbloqueo de medalla en función del puntaje máximo del usuario

Para el desbloqueo de medallas, mostrado con código en la ilustración 29, se recupera el valor del puntaje máximo con la función `localStorage.getItem()` y, dependiendo de su valor, se evalúa si es suficiente para mostrar una medalla o si no.

```
this.puntajeMaxArea = 0;

if(localStorage.getItem("pMaxArea")!=null){
    this.puntajeMaxArea = localStorage.getItem("pMaxArea");
}

//Aquí va el resto del código

if(this.puntajeMaxArea>=10){
    this.medallaBronceArea.anims.play("medalla_bronce");
}
```

Ilustración 29 Desbloqueo de medalla en función del puntaje máximo del usuario

De esta forma, el usuario, al ver el menú, observa estas, teniendo las medallas su propia animación, contemplándose esto en la imagen 38.



Imagen 38 Desbloqueo de medalla en la interfaz de la herramienta

En la imagen 39 se muestra que, en caso de que el usuario, o debugger, quisiera reiniciar los valores por defecto de la aplicación sin recurrir estrictamente al código, se emplea un texto “borrado de datos” que funge como botón para cumplir con la función que lleva en el nombre.

Borrado de datos...

Imagen 39 “Botón” de borrado de datos para depuración

Y lo que ocurre, a nivel de código, es que todo lo que haya sido guardado con `localStorage` es eliminado, borrando consigo el puntaje máximo que haya obtenido y, por ende, las medallas conseguidas.

```
this.borradoDeDatos.on(eventos.POINTER_DOWN, () => {  
  localStorage.clear();  
  this.borradoDeDatos.setTint(0xC0392B);  
  console.log("Borrado completo...");  
});
```

Ilustración 30 Borrado de datos a través de código

Capítulo 4: Pruebas y resultados

Una vez diseñada la herramienta, se vuelve necesario probar y comprobar el rendimiento que esta puede llegar a tener antes de que llegue a manos de los usuarios finales, como los alumnos en este caso, y, en caso de funcionar correctamente, hacer que esta cumpla con el propósito para la cual fue creada. Por esto, a continuación, se mencionan las pruebas de funcionamiento realizadas, incluyendo los dispositivos destinados a dicho propósito; así como también los resultados obtenidos tras la utilización por el alumnado durante determinado tiempo, haciendo énfasis en el análisis previo de resultados y la mejoría posterior a través de la comparación de ambos datos.

4.1 Pruebas de funcionamiento

Para las pruebas de funcionamiento se emplearon un equipo de escritorio y un dispositivo móvil, siendo este una Tablet, además del complemento de Visual Studio Code denominado “live server” que permite convertir el equipo en el cual se esté desarrollando el proyecto web en un servidor de prueba. Para ello, basta con pulsar el botón “Go Live”. Al momento, se abrirá una ventana del navegador web con la raíz del proyecto, indicando también el puerto que se está utilizando para dicha ejecución. En la imagen 40 se observa la raíz del proyecto con todos los elementos que la conforman, resultado de ejecutar la extensión previamente mencionada.

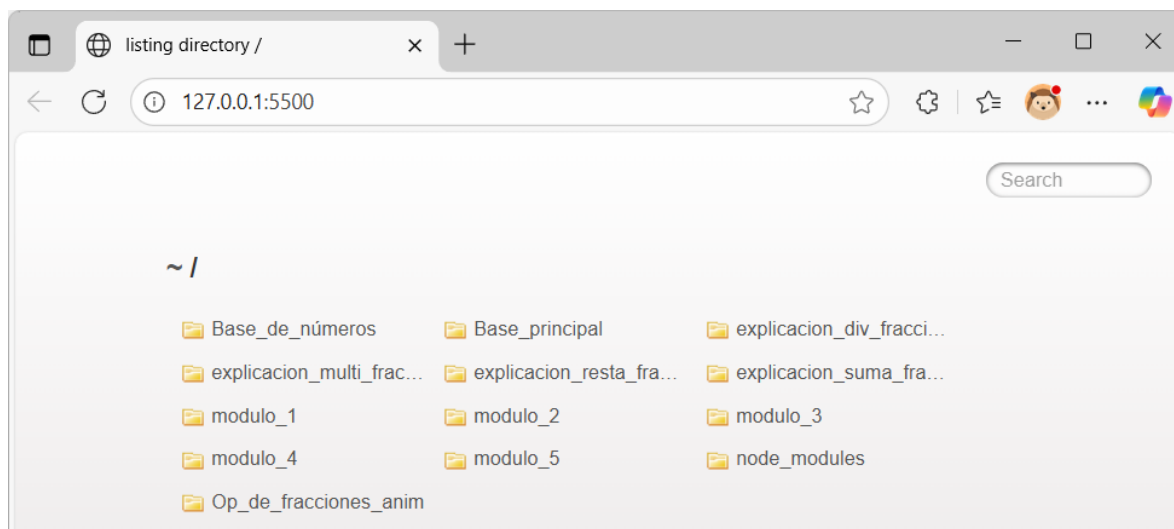


Imagen 40 Raíz del proyecto, mostrada en una PC/Laptop

En el caso de querer probar el proyecto en equipos ajenos, como es el caso, es necesario conocer la dirección IP del equipo huésped, así como también el puerto que se está ocupando, siendo este, en este caso, el 5500. Para la dirección IP es necesario abrir la ventana o consola de comandos y, posteriormente escribir el comando `ipconfig` que muestra dicha información, tal como se contempla en la imagen 41.

```
C:\Windows\system32\cmd.e: x + v
Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . . :
Dirección IPv6 . . . . . : 2806:262:40e:f839::c
Dirección IPv6 . . . . . : 2806:262:40e:f839:529f:782c:
d11d:7058
Dirección IPv6 temporal. . . . . : 2806:262:40e:f839:7c90:86d5:
6dd7:49ba
Vínculo: dirección IPv6 local. . . : fe80::99fd:e9d:58f1:12bc%13
Dirección IPv4. . . . . : 192.168.0.41
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : fe80::4275:c3ff:fe2f:
dcc0%13
192.168.0.1

C:\Users\Marco Antonio PB>
```

Imagen 41 Consola de comandos de Windows y ejecución de *ipconfig*

Con ello, la dirección IP del equipo huésped, en este caso, es 192.168.0.41, misma que tendremos que introducir, junto con el puerto, en el dispositivo ajeno con el que queramos acceder, mostrándose así la raíz del proyecto, tal como se puede contemplar en la imagen 42.

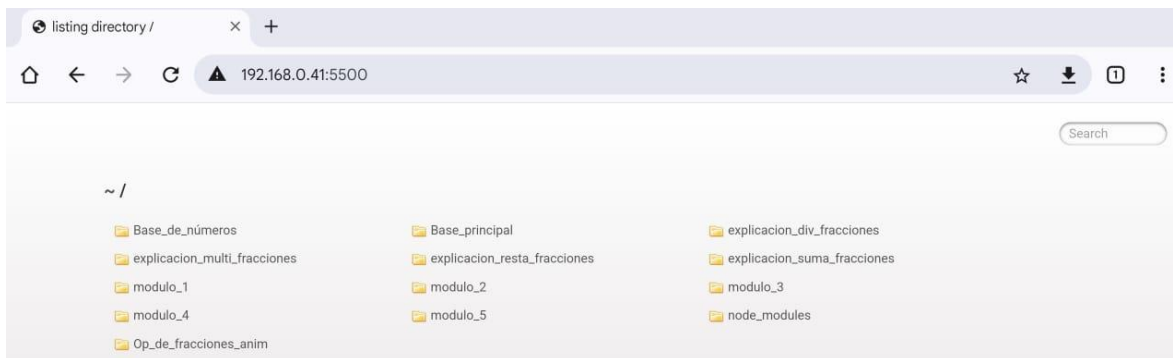


Imagen 42 Raíz del proyecto, mostrada en un dispositivo externo (tablet)

Hecho lo anterior, se puede ver que la herramienta funciona correctamente y con fluidez, sólo teniendo un ligero retraso en la aparición de la pantalla de inicio, esto siendo derivado de la carga de elementos gráficos que son requeridos. De manera adicional, también se comprueba que dicho software es compatible para su uso tanto en computadoras como en dispositivos táctiles de gran pantalla.

4.2 Resultados obtenidos

La muestra considerada está conformada por un total de 30 alumnos de primero y segundo de secundaria, con una edad aproximada de entre 12 y 13 años respectivamente, de una institución privada perteneciente a la CORDE Puebla Sur que, por su condición de menores de edad, se encuentran protegidos por normativas de privacidad y resguardo de datos personales, es por esto por lo que no es posible describir con detalle todos los datos obtenidos, como las características individuales.

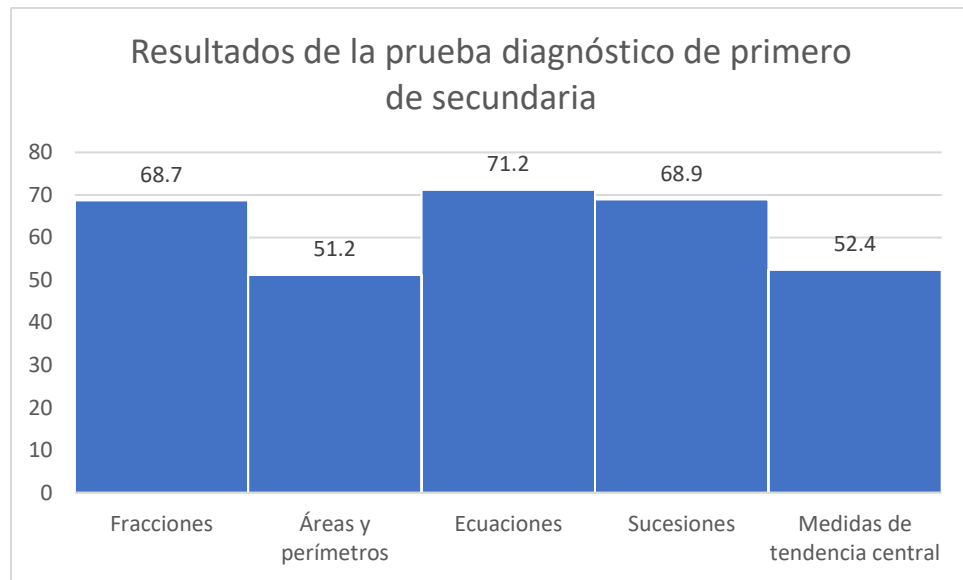
La evaluación realizada corresponde a una evaluación diagnóstica de tipo exploratorio (ver anexo), cuyo propósito fue identificar el desempeño inicial y la respuesta de los estudiantes ante el uso de la herramienta interactiva. Con esta se obtuvieron resultados bajos en 5 temas específicos, siendo estos las operaciones con fracciones, el cálculo de áreas y perímetros, la resolución de ecuaciones de primer grado, las expresiones algebraicas para sucesiones y las medidas de tendencia central.

De forma desglosada, los puntajes son los siguientes.

- Alumnos de primero de secundaria:

Tabla 1 Resultados de la prueba diagnóstico de primero de secundaria

Tema	Puntajes obtenidos
Fracciones	68.7
Áreas y perímetros	51.2
Ecuaciones	71.2
Sucesiones	68.9
Medidas de tendencia central	52.4

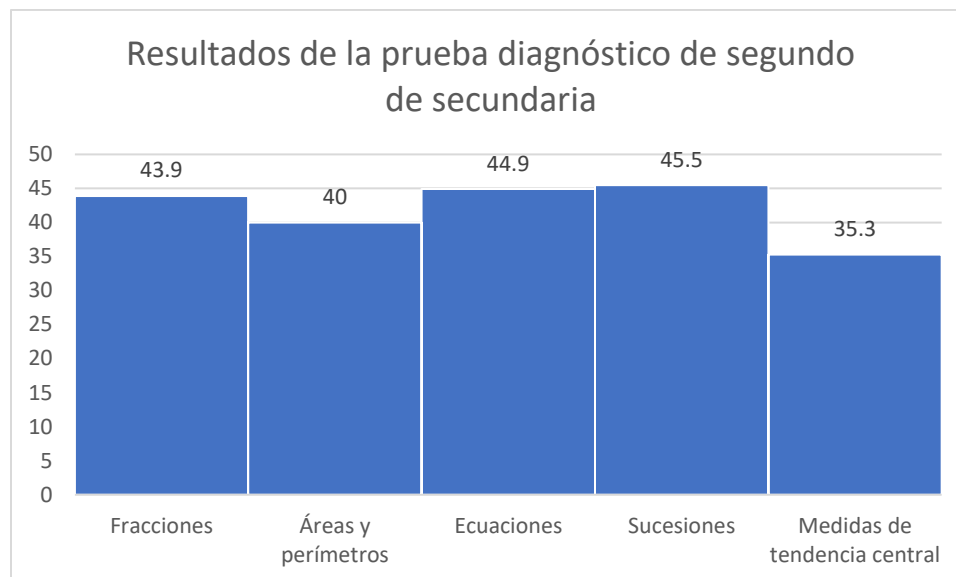


Gráfica 1 Resultados de la prueba diagnóstico de primero de secundaria

- Alumnos de segundo de secundaria:

Tabla 2 Resultados de la prueba diagnóstico de segundo de secundaria

Tema	Puntajes obtenidos
Fracciones	43.9
Áreas y perímetros	40
Ecuaciones	44.9
Sucesiones	45.5
Medidas de tendencia central	35.3



Gráfica 2 Resultados de la prueba diagnóstico de segundo de secundaria

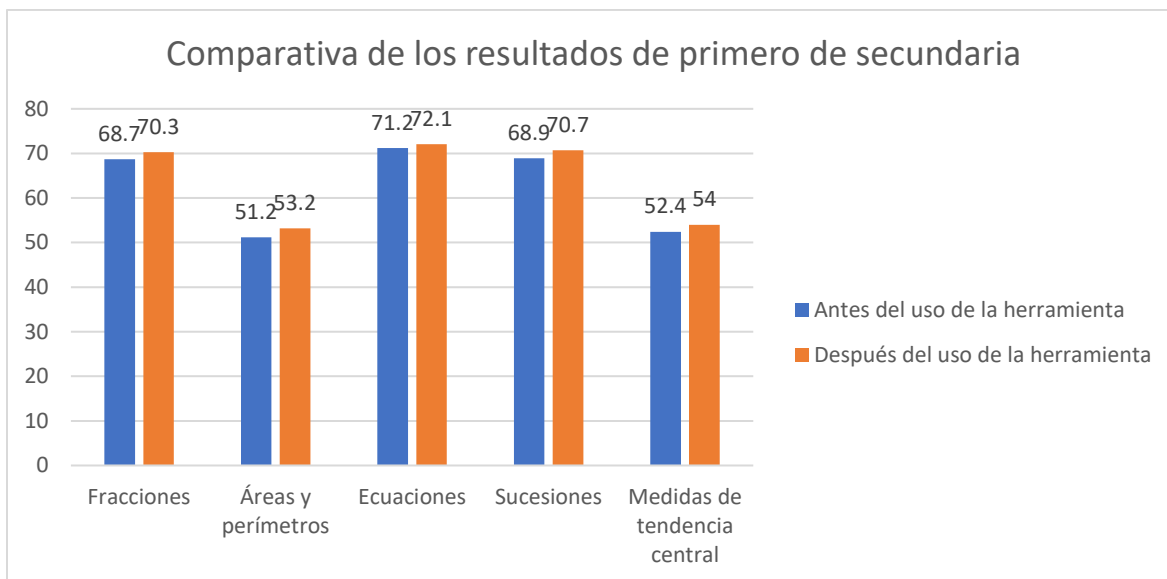
Posterior a esto, se les solicitó a los alumnos que hicieran uso de la herramienta por un tiempo aproximado de un mes, esto porque, principalmente, cada periodo de evaluación tiene esa duración, de forma que, con esto, se pudiera comprobar si hubo una mejoría o si no, tomando como instrumento, una vez más, la prueba diagnóstica mencionada anteriormente (ver anexo).

Tras la utilización de la herramienta, se obtuvieron los siguientes resultados.

- Alumnos de primero de secundaria:

Tabla 3 Comparativa de los resultados de primero de secundaria antes y después del uso de la herramienta

Tema	Diagnóstico previo	Resultados después del uso de la herramienta	Diferencia
Fracciones	68.7	70.3	+1.6
Áreas y perímetros	51.2	53.2	+2
Ecuaciones	71.2	72.1	+0.9
Sucesiones	68.9	70.7	+1.8
Medidas de tendencia central	52.4	54	+1.6

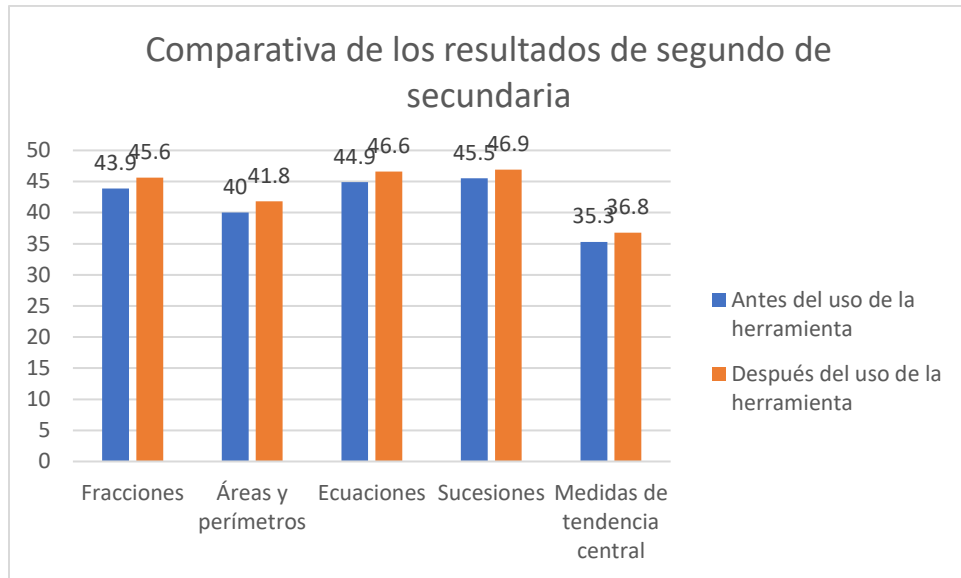


Gráfica 3 Comparativa de los resultados de primero de secundaria antes y después del uso de la herramienta

- **Alumnos de segundo de secundaria:**

Tabla 4 Comparativa de los resultados de segundo de secundaria antes y después del uso de la herramienta

Tema	Diagnóstico previo	Resultados después del uso de la herramienta	Diferencia
Fracciones	43.9	45.6	+1.7
Áreas y perímetros	40	41.8	+1.8
Ecuaciones	44.9	46.6	+1.7
Sucesiones	45.5	46.9	+1.4
Medidas de tendencia central	35.3	36.8	+1.5



Gráfica 4 Comparativa de los resultados de segundo de secundaria antes y después del uso de la herramienta

Realizando la comparación entre los resultados anteriores y los posteriores, se obtiene que, tras el uso de la herramienta por el alumnado de los grupos previamente mencionados, sí existió una mejoría general de, aproximadamente, el 0.9 al 1.8 por ciento, siendo el más bajo el de ecuaciones de primer grado en primero de secundaria, y los más altos los de expresiones algebraicas para sucesiones de primero y cálculo de áreas y perímetros en segundo.

Capítulo 5: Conclusiones y trabajo a futuro

A lo largo de la tesis se ha mencionado la importancia subyacente de este proyecto y su intención de apoyar a alumnos de nivel secundaria en poder comprender temas que, a partir de los datos recolectados y analizados, son complicados para ellos, y más a raíz del rezago educativo, producto de la crisis sanitaria experimentada en el 2020 en adelante. Por esto, el presente se ha realizado íntegramente desde cero, empleando metodologías activas que han demostrado ser más llamativas para personas de ese rango de edad, como lo es el caso de la gamificación.

De igual forma, durante el desarrollo de la herramienta, se pudieron identificar varias problemáticas y, a su vez, formas de resolver, en gran medida, esas limitantes. Una de estas ocurrió en el desarrollo de los elementos gráficos, puesto que trabajar con pixel art requiere de tiempo, además de intentar hacer llamativa la aplicación, para que los usuarios puedan entender, paso a paso, cómo resolver problemas asociados a los temas mencionados con anterioridad. El diseño de las mecánicas también representó un desafío porque, uno de los propósitos es que los usuarios no sientan que se trata de un programa más, sino que también existe un elemento diferenciador que puede hacer que el proyecto resalte por sobre los demás. Otro reto fue definir qué dispositivos y plataformas serían el objetivo, llegando así a la conclusión que lo mejor serían las PC y dispositivos móviles de pantalla grande, para así apreciar cada detalle gráfico de la herramienta, sin tener que instalar nada en estos, volviéndolo así accesible para todo aquel que quisiera hacer uso de esta.

En general, se observa que el proyecto cumple con múltiples objetivos específicos, como lo son el diseño de la estructura y la interfaz, el desarrollo de contenido educativo atractivo, la integración de elementos de videojuegos y la evaluación del cambio en el rendimiento de los usuarios objetivo, siendo estos los alumnos de secundaria, tras el uso de la herramienta.

De los resultados obtenidos, en primero de secundaria se nota una mejoría de 0.9 a 1.8 puntos, en las áreas de ecuaciones y sucesiones; mientras que en segundo de secundaria fue de 1.4 a 1.8 en sucesiones y áreas y perímetros. Entonces, con base en los datos de inicio y fin de la utilización de la herramienta, se ha detectado que la diferencia no es significativa, esto siendo atribuible a la cantidad de tiempo del manejo de esta por parte de los usuarios, ya que fue de, aproximadamente, un mes. Lo anterior, a su vez, sugiere que, de usarse más tiempo, los resultados de los alumnos podrían mejorar un poco más.

Este proyecto, en palabras propias, me ha ayudado a repasar múltiples habilidades aprendidas y adquiridas a lo largo de la carrera, desde la materia de física que, aunque de primer vistazo no lo parezca, se ocupa, ya que Phaser3, por su naturaleza de motor de juego, la implementa. Además de ocupar conceptos como lo que es programación, diseño de algoritmos, pruebas de escritorio y, sobre todo, el uso del pensamiento lógico y crítico para poder predecir y estimar el comportamiento del proyecto, esto siendo específicamente útil en la parte en la que

es necesario saber qué es lo que no está funcionando tal como debería. El contenido de otras materias que me han ayudado han sido las de desarrollo de aplicaciones web y desarrollo de videojuegos, la primera por conocer cómo funciona una página HTML y sus contenidos, así como el uso de JavaScript y el comprender el concepto de framework; y la segunda por el uso de los conceptos previos pero orientados a elementos lúdicos. Ingeniería de software también se hace presente porque tuve que considerar la mayor cantidad de elementos posibles previos al diseño e implementación de la herramienta para asegurar su correcto funcionamiento, sin mencionar las modificaciones y correcciones realizadas sobre la marcha. Interacción humano computadora me ayudó a comprender mejor ciertos conceptos, como lo es el contraste, esto para hacer llamativa la aplicación o, por lo menos, no hacer que, visualmente, fuera disonante para el usuario.

Durante la implementación de la herramienta, se llevó a cabo un seguimiento presencial, de forma que los usuarios pudieron dar sus puntos de vista acerca de su experiencia y recomendaciones para la mejora de esta.

Tomando en cuenta lo anterior, y como parte del trabajo a futuro, se consideran las siguientes áreas de oportunidad:

- **Adición de más temas, incluso más complejos:** la herramienta, con base en los datos recolectados, arrojó una mejoría en los alumnos, con esto, se interpreta que esta se puede beneficiar de más temas que sean de interés, aumentando incluso en nivel de dificultad para aquellos usuarios que deseen un reto mayor.
- **Diseño e integración de más ejercicios:** para que el usuario pueda practicar más e, incluso, estos no se repitan con facilidad.
- **Corrección y mejora de mecánicas:** refinar las mecánicas existentes podría suponer que los usuarios tengan una mejor experiencia con el uso de la herramienta.
- **Configurar nuevas mecánicas:** como lo son la interacción con más personajes o elementos de la herramienta, la posibilidad de corregir errores y recuperar los puntos de vida perdidos, añadir “easter eggs”, o incluso crear una tienda para que los puntos de aciertos se puedan intercambiar por coleccionables, etc.
- **Implementación de más detalles estéticos:** con el fin de hacer la herramienta más inmersiva y, con ello, más interesante para los usuarios.
- **Diseño y adición de más personajes:** para que funcionen como apoyo, a través de consejos; como distractor para que el usuario pierda puntaje o como un añadido extra.

Lo que, al final, me llevo de todo esto es la condensación de los conceptos vistos a lo largo de toda la carrera universitaria y también trabajo autodidacta que, con esto, abre nuevas posibilidades, nuevas oportunidades de mejora y, sobre todo, un sinfín de nuevas experiencias a la espera de ser descubiertas.

Anexos

EXAMEN DIAGNÓSTICO DE MATEMÁTICAS

NOMBRE DEL ALUMNO: _____

GRADO: _____

CALIFICACIÓN: _____

1. ¿Cuál es el resultado correcto de la siguiente suma de fracciones?

$$\frac{1}{9} + \frac{8}{3}$$

a) $\frac{75}{12}$

b) $\frac{75}{27}$

c) $\frac{9}{12}$

2. ¿Cuál es el resultado correcto de la siguiente multiplicación de fracciones?

$$\frac{7}{2} \times \frac{3}{2}$$

a) $\frac{6}{14}$

b) $\frac{14}{6}$

c) $\frac{21}{4}$

3. De la siguiente figura geométrica, ¿Cuál es su área?

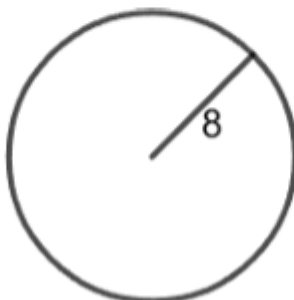


a) $A = 24 u^2$

b) $32 u^2$

c) $128 u^2$

4. De la siguiente figura geométrica, ¿Cuál es su perímetro?



a) $P = 200.96 u$

b) $P = 50.24 u$

c) $P = 25.12 u$

5. A partir de la siguiente ecuación, ¿Cuál es el valor de la variable "x"?

$$\frac{x+4}{2} = 10$$

- a) $x = 16$ b) $x = 24$ c) $x = 9$

6. A partir de la siguiente ecuación, ¿Cuál es el valor de la variable "x"?

$$5y - 7 = y + 1$$

- a) $y = \frac{1}{2}$ b) $y = 2$ c) $y = -1$

7. ¿Cuál de las siguientes funciones representa correctamente el comportamiento de la siguiente sucesión?

$$3, 1, -1, -3, -5, \dots$$

- a) $a_n = -2n + 3$ b) $a_n = -2n + 5$ c) $a_n = 3n - 2$

8. A partir de la sucesión y su función asociada, ¿Cuál es el valor que se encuentra en la posición a_7 ?

$$6, 10, 14, 18, \dots \qquad a_n = 4n + 2$$

- a) $a_7 = 18$ b) $a_7 = 46$ c) $a_7 = 30$

9. A partir del siguiente conjunto de datos, ¿Cuáles son los valores correctos de la media, mediana y moda?

$$\{2, 6, 6, 5, 1\}$$

- a) *Media* = 6,
Mediana = 2, *Moda* = 1 b) *Media* = 1,
Mediana = 6, *Moda* = 2 c) *Media* = 4,
Mediana = 5, *Moda* = 6

10. A partir del siguiente conjunto de datos, ¿Cuáles son los valores correctos de la media, mediana y moda?

$$\{5, 1, 4, 5, 6\}$$

- a) *Media* = 4.2,
Mediana = 5, *Moda* = 5 b) *Media* = 5,
Mediana = 4, *Moda* = 4 c) *Media* = 6,
Mediana = 6, *Moda* = 6

Bibliografía

Encuesta Nacional de Consumo de Contenidos Audiovisuales. (2023). En somosaudiencias.ift.org.mx. Instituto Federal de Telecomunicaciones. https://somosaudiencias.ift.org.mx/archivos/01reportefinalencca2023_vp.pdf

Guerrero, A. A. (2022). Educación y cultura | Los videojuegos en casa. El Sol de Puebla | Noticias Locales, Policiacas, sobre México, Puebla y el Mundo. <https://www.elsoldepuebla.com.mx/analisis/educacion-y-cultura-los-videojuegos-en-casa-8733003.html>

Puebla se ubicó en séptimo lugar a nivel nacional en rezago educativo. (2023). Milenio. <https://www.milenio.com/politica/comunidad/puebla-ubico-septimo-nvel-nacional-rezago-eduvativo>

Ruiz, L. (2023). Rezago educativo en Puebla alcanza el 35.5 por ciento. El Universal Puebla. <https://www.eluniversalpuebla.com.mx/educacion/rezago-educativo-en-puebla-alcanza-el-355-por-ciento/>

Gamificación en el aula: por qué y cómo aplicarla. (2025). Santander Open Academy. <https://www.santanderopenacademy.com/es/blog/gamificacion-en-el-aula.html>

Mecánicas de juego más habituales en los videojuegos | Tokio School. (s. f.). Tokio School. <https://www.tokioschool.com/noticias/mecanicas-de-juego-habituales-en-videojuegos/>

3, E. (2019). ¿Qué es la gamificación y cuáles son sus objetivos? | EDUCACIÓN 3.0. EDUCACION 3.0. <https://www.educaciontrespuntocero.com/noticias/gamificacion-que-es-objetivos/>

García Peñalvo, F. J. (2009). Software educativo: evolución y tendencias. Aula, 14. <https://doi.org/10.14201/1300>. <https://revistas.usal.es/tres/index.php/0214-3402/article/view/1300/1371>

Merino, M. (2023). Qué fue de la Encarta, la «enciclopedia virtual» de los que fuimos al colegio cuando no había Wikipedia. Genbeta. <https://www.genbeta.com/a-fondo/que-fue-encarta-enciclopedia-virtual-que-fuimos-al-colegio-cuando-no-habia-wikipedia>

Ceupe, A. (2023). Ceupe. Ceupe. <https://www.ceupe.com/blog/razonamiento-matematico.html>

LA IMPORTANCIA DEL PENSAMIENTO MATEMATICO el. . . | Formando formadores. (s. f.). <https://www.formandoformadores.org.mx/colabora/publicaciones/la-importancia-del-pensamiento-matematico-el>

Derivando. (2022). Pero. . . ¿PARA QUÉ SIRVEN LAS MATEMÁTICAS? [Vídeo]. YouTube. <https://www.youtube.com/watch?v=RIUZv0MoVWk>

Uees, & Uees. (2022). Las matemáticas en la vida cotidiana. UEES - Universidad Espíritu Santo. <https://uees.edu.ec/las-matematicas-en-la-vida-cotidiana/>

Análisis de matrícula, abandono escolar, eficiencia terminal y cobertura en educación básica de 2019 a 2023. (2024). Secretaría de Educación Pública. https://educacionbasica.sep.gob.mx/wp-content/uploads/2024/06/Abandono-escolar-en-Educacion-Basica-2019-2023_.pdf

La pérdida de matrícula escolar en México: entre el abandono escolar, la privatización y los “otros datos”. (2025). Animal Político. <https://animalpolitico.com/analisis/organizaciones/aprender-es-mi-derecho/perdida-matricula-escolar>

García, A. (2025). Deserción escolar afecta a 4 millones de estudiantes en México. El Sol de Toluca | Noticias Locales, Policiacas, Sobre México, Edomex y el Mundo. <https://oem.com.mx/elsoldetoluca/local/desercion-escolar-afecta-a-4-millones-de-estudiantes-en-mexico-24947735>

Principales cifras del sistema educativo nacional 2022 - 2023. (2023). Secretaría de Educación Pública. https://www.planeacion.sep.gob.mx/Doc/estadistica_e_indicadores/principales_cifras/principales_cifras_2022_2023_bolsillo.pdf

Sistema límbico: MedlinePlus enciclopedia médica ilustración. (s. f.). https://medlineplus.gov/spanish/ency/esp_imagepages/19244.htm

Dopamine. (2022). Physiopedia. <https://www.physio-pedia.com/index.php?title=Dopamine&oldid=317693>.

Castillero Mimenza, O. (2025). Corteza prefrontal: funciones y trastornos asociados. Psicología y Mente. <https://psicologiymente.com/neurociencias/corteza-prefrontal>

insightpsychology. (2024). Procrastination and the Brain: A Neuroscience Guide. Insights Psychology. <https://insightpsychology.org/the-neuroscience-of-procrastination/>

Belsasso, G., Estañol, B., & Juárez, H. (2002). Nuevas estrategias farmacológicas en el tratamiento de las adicciones (Primera edición). Secretaría de Salud. http://www.biblioteca.cij.gob.mx/Archivos/Materiales_de_consulta/Drogas_de_Abu_so/Articulos/Tratamiento_adicciones.pdf

Aprendizaje a través del juego: Reforzar el aprendizaje a través del juego en los programas de educación en la primera infancia. (2018). UNICEF. <https://www.unicef.org/sites/default/files/2019-01/UNICEF-Lego-Foundation-Aprendizaje-a-traves-del-juego.pdf>

Beneficios de los videojuegos en el aprendizaje: ¿Por qué tus hijos sí deberían jugar a videojuegos? (s. f.). Iberdrola. <https://www.iberdrola.com/talento/beneficios-videojuegos-aprendizaje>

Playmotiv, & Playmotiv. (2022). Gamificación: cifras y casos de éxito - Playmotiv. Playmotiv - Gamificación Para Empresas. <https://playmotiv.com/nada-mejor-que-una-dosis-de-datos-para-comprender-mejor-la-repercusion-de-la-gamificacion-en-los-negocios/>

blogs-es.vorecol.com & Vorecol Editorial Team. (s. f.). Evaluación del impacto de la gamificación en la retención de conocimientos. <https://blogs-es.vorecol.com/articulo-evaluacion-del-impacto-de-la-gamificacion-en-la-retencion-de-conocimientos-3783>

Universidad Europea, Universidad presencial (Madrid, Valencia, Alicante, Canarias, Málaga) y Online. (2023). Tipos de metodologías educativas: cómo elegir la mejor. Universidad Europea. <https://universidadeuropea.com/blog/tipos-metodologias-educativas/>

Metodologías de enseñanza innovadoras: ¿cuáles son las más efectivas? (2025). Santander Open Academy. <https://www.santanderopenacademy.com/es/blog/metodologias-de-ensenanza.html>

Apoyo a la apropiación del plan y programas de estudio 2022: Enfoque STEAM. (2024). Secretaría de Educación Pública. https://educacionbasica.sep.gob.mx/wp-content/uploads/2024/06/Lenguajes_Enfoque-STEAM.pdf

MS, E. O. (2025). ¿Qué es la repetición espaciada y por qué es buena para aprender? Duolingo Blog. <https://blog.duolingo.com/es/repeticion-espaciada-para-aprender/>

Jann Mardenborough - GT Academy. (s. f.). gran-turismo.com. <https://www.gran-turismo.com/mx/academy/graduates/jann/>

Los 6 principales frameworks de JavaScript que debes conocer | UNIR. (2023). UNIR. <https://www.unir.net/revista/ingenieria/frameworks-javascript/>

Universidad Europea (2024). ¿Qué es un motor de juegos? Universidad Europea Creative Campus. <https://creativecampus.universidadeuropea.com/blog/motor-juegos/>

Navarro, J. G. (2021). Phaser 3: Mi primer juego HTML5/JS (Parte 1). <https://www.enmilocalfunciona.io/phaser-3-mi-primer-juego-html5-javascript/>

Octo-rl @Octo-rl. Toyota supra. (s. f.). Pixilart. https://www.pixilart.com/art/toyota-supra-sr20ec24b5fe6aws3?ft=topic&ft_id=22

BrookyLnG @BrookyLnG. F1 Battle. (s. f.). Pixilart. https://www.pixilart.com/art/f1-battle-4b0b04b87715?ft=topic&ft_id=22

crabCrabCrab @ crabCrabCrab. Village well. (s. f.). Pixilart.
<https://www.pixilart.com/art/village-well-sr5z080417eff1aws3>

MasterBeany @MasterBeany. Snek. (s. f.). Pixilart.
<https://www.pixilart.com/art/snek-sr2eee0167048aws3>

SauceK. FREE Original Sonic Sprite Sheets V3. (s. f.). Patreon.
<https://www.patreon.com/posts/original-sonic-5-92998423?l=es>

UCM-Proyecto de Innovación Software libre para ciencias e ingenierías. (s. f.).
<https://www.ucm.es/pimcd2014-free-software/latex>

Fuerte, K. (2022). ¿Qué son los Serious Games? Observatorio / Instituto Para el Futuro de la Educación. <https://observatorio.tec.mx/que-son-los-serious-games/>

Mika, A. (2025). Web Application vs. Desktop Application. Ramotion.
<https://www.ramotion.com/blog/web-application-vs-desktop-application/>

Castro, L. A., & Rodríguez, M. D. (Eds.). (2020). Interacción Humano-Computadora y Aplicaciones en México (Segunda edición). Academia Mexicana de Computación, A, C. <https://amexcomp.mx/media/publicaciones/int-humano-comp-apps.pdf>

Castaño, H. (2023). La Curva (de dificultad). Bitwares. <https://bitwares.net/la-curva-de-dificultad/>