



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

Videojuegos para Tod@s en Escuelas de Chile:
Experiencias de Visual Scripting en Unity para el
Módulo de Matemáticas

ROBERTO IGNACIO SAAVEDRA VELÁSQUEZ

Profesor Guía: CRISTIAN VIDAL SILVA

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

Talca – Chile
Enero, 2026



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

Videojuegos para Tod@s en Escuelas de Chile:
Experiencias de Visual Scripting en Unity para el
Módulo de Matemáticas

ROBERTO IGNACIO SAAVEDRA VELÁSQUEZ

Profesor Guía: CRISTIAN VIDAL SILVA

Miembro Comisión Examinadora: CAMILA ALIAGA, CLAUDIA DE LA FUENTE

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

El presente documento fue calificado con nota: _____

Talca – Chile

Enero, 2026

AGRADECIMIENTOS

Agradecimientos a:

Mis padres, por apoyarme en todo lo posible, aunque no me entiendan mucho.

Mis amigos, por aliviar los días de trabajo con su compañía y ánimo.

A Ado y Adam Gubman, por amenizar esos días pesados con su grandiosa música.

TABLA DE CONTENIDOS

	página
Agradecimientos	I
Tabla de Contenidos	II
Índice de Figuras	IV
Índice de Tablas	VI
Resumen	VII
 1. Introducción	 1
1.1. Problema	2
1.1.1. Objetivo General	3
1.1.2. Objetivos Específicos	3
1.2. Preguntas de Investigación	4
 2. Estado del Arte	 5
2.1. Gamificación y su Aplicación en Escuelas de Chile	5
2.2. GeoGebra	6
2.3. Khan Academy	7
2.4. Umáximo	8
 3. Marco Teórico	 11
3.1. Competencias de Matemáticas	11
3.2. Competencias de Programación	12
3.3. Visual Scripting de Unity	13
 4. Metodología	 15
4.1. Ejercicios y Evaluaciones	16
 5. Implementación y Resultados	 24
5.1. Implementación	24
5.1.1. Plano Cartesiano	24

5.1.2. Verificación y Extracción de Datos	25
5.2. Resultados	27
6. Conclusión	31
6.1. Trabajo a Futuro	32
6.2. Palabras Finales	32
Bibliografía	33
A: Carta Gantt	37
B: Gráficos de Resultados de Encuesta de Satisfacción	38

ÍNDICE DE FIGURAS

	página
2.1. Interfaz principal de GeoGebra	7
2.2. Interfaz de la zona de práctica de Khan Academy	8
2.3. Interfaz principal del estudiante de Khan Academy	8
2.4. Interfaz del estudiante de Umáximo	9
2.5. Interfaz de asignación de tareas del profesor de Umáximo	10
3.1. Nodos de un Grafo en Visual Scripting de Unity	14
4.1. Diagrama de Flujo del Ejercicio 2	17
4.2. Ejercicio 2: Instanciamiento de puntos y línea en el plano cartesiano en Visual Scripting	18
4.3. <i>DibujarLínea</i> : Nodo personalizado para Dibujar una Línea	18
4.4. Diagrama de Flujo del Ejercicio 3	19
4.5. Ejercicio 3: Suma de Vectores en Visual Scripting	19
4.6. Diagrama de Flujo del Ejercicio 4	20
4.7. Ejercicio 4: Multiplicación de un Vector por un Escalar en Visual Scripting	21
4.8. Diagrama de Flujo del Ejercicio 6	22
4.9. Ejercicio 6: Traslación de un Triángulo en Visual Scripting	23
5.1. Estudiantes del Octavo Básico A de la Escuela Prosperidad de Talca realizando las experiencias. Su profesor, Leonardo Morales, supervisa la otra mitad que no está participando en los talleres	27
5.2. Comentarios de la Encuesta de Satisfacción	30
A.1. Carta Gantt del proyecto	37
B.1. ¿Qué tan probable es que recomiendes el taller “Videojuegos para Tod@s en Escuelas de Chile: Experiencias de Visual Scripting en Unity para el Módulo de Matemática” a tus amigos o familiares?	38
B.2. ¿En general, cómo describirías el taller?	39
B.3. ¿Qué tan claros fueron los objetivos de programación del taller?	39
B.4. ¿Qué tan claros fueron los objetivos de matemáticas del taller?	40

B.5. ¿Cómo calificarías en general al instructor del taller?	40
B.6. ¿Qué tan adecuadas te parecieron las actividades que se realizaron durante el taller?	41
B.7. ¿Te inscribirías a este taller si se volviese a hacer otra vez?	41

ÍNDICE DE TABLAS

	página
5.1. Tabla de resultados de pruebas diagnósticas	28
5.2. Tabla compilatoria de resultados del ejercicio 3.	28
5.3. Tabla compilatoria de resultados del ejercicio 4.	29
5.4. Tabla de resultados de pruebas finales	29

RESUMEN

Los establecimientos educacionales se adaptan cada año con mayor facilidad al uso de las Tecnologías de la Información y la Comunicación (TIC) en la formación básica de niños y jóvenes. La gamificación del aprendizaje se ha consolidado como una herramienta efectiva y atractiva para muchas escuelas, ya que representa una opción accesible y común para captar la atención de los estudiantes. Esto es especialmente evidente en el área de matemáticas, donde los docentes demuestran un creciente interés en incorporar estas tecnologías como recursos didácticos útiles.

Uno de los profesores de la Escuela Prosperidad de Talca, Leonardo Morales, estableció un taller de habilidades digitales basado en Scratch y Arduino, el cual es muy popular entre los estudiantes desde kinder hasta octavo básico. Quiere implementar un sistema similar en el curso donde es profesor jefe, en combinación con la programación, específicamente en el área de las matemáticas.

Este proyecto consiste en el diseño y desarrollo de un prototipo interactivo orientado al fortalecimiento de competencias en geometría en estudiantes de octavo básico de la Escuela Básica Prosperidad de Talca. La propuesta integra el aprendizaje de programación mediante Visual Scripting en el motor Unity, incorporando un sistema de registro de datos en tiempo real. Este mecanismo permite evaluar el desempeño estudiantil a través de métricas como tiempos de ejecución, número de intentos y tasas de éxito, facilitando un análisis comparativo entre el estado inicial y final de los participantes para medir el impacto de la experiencia educativa.

Como resultado de este proyecto, se espera que los estudiantes del grupo experimental, en primer lugar, demuestren un mayor dominio de los contenidos de geometría y, en segundo lugar, que puedan entender conceptos básicos de programación a través de Visual Scripting. Por último, se espera que desarrollen interés en la programación, viéndola como una herramienta capaz de ofrecer diversas utilidades para su futuro.

Aunque hubo percances al registrar los resultados, estos fueron suficientes para comprobar que hubo una mejora en la comprensión de conceptos de geometría en los niños al introducirlos a la programación visual. A futuro, el proyecto puede enfocarse en convertirse en un *Serious Game* como otras plataformas, o en una serie de talleres para enseñar matemáticas a través de la programación.

1. Introducción

La gamificación, como concepto, promueve el diseño de experiencias de aprendizaje en las que la principal tarea del docente consiste en elaborar propuestas didácticas integrales que permitan a los estudiantes aprender de forma activa y significativa [1]. De esta forma, pueden ser vividas como un juego por parte de quien vaya a aprender, recogiendo la idea de utilizar elementos de juego a la hora de diseñar dichas experiencias. En las mismas palabras de Zichermann y Cunningham [2], la gamificación representa el proceso de game-thinking y mecánicas de juego para involucrar a los usuarios y resolver problemas, el cual puede aplicarse a cualquier contexto a través de influir en la motivación y el comportamiento humano.

Los establecimientos educacionales se adaptan cada año con mayor facilidad al uso de las Tecnologías de la Información y la Comunicación (TIC) en la formación básica de los niños [3]. La gamificación del aprendizaje se ha consolidado como una herramienta efectiva y atractiva para muchas escuelas, ya que representa una opción accesible y común para captar la atención de los estudiantes [4]. Una de las áreas donde hay más interés en aplicar la gamificación es en las matemáticas [5]. En Chile, incluso, existe una herramienta de gamificación de matemáticas llamada Umáximo [6], de la cual se hablará más tarde. Sin embargo, la relación del país con las matemáticas siempre ha sido complicada. Los puntajes promedios del SIMCE (Sistema de Medición de la Calidad de la Educación) de matemáticas de Octavo Año Básico oscilaban entre 259 y 263 puntos en la última década [7], siendo más pareja que la de comprensión lectora. No obstante, existe una gran diferencia de resultados entre grupos socioeconómicos altos y bajos. El puntaje promedio de los estudiantes de Octavo Básico del primer grupo en el SIMCE de matemáticas del año 2019 es de 308 puntos, superando en 74 puntos al segundo grupo, con 234 [7]. Y esta diferencia de puntos sigue siendo similar o aún mayor en años anteriores.

La situación en las salas de clases chilenas en torno a cómo se enseñan las matemáticas, según un estudio de Radovic y Preiss [8], se centra en una secuencia triádica de seguimiento o *initiation-response-follow* (IRT). Esta secuencia tiene una dinámica definida: el profesor expone el tema, pregunta a los estudiantes o supervisa su trabajo, el cual se basa en la práctica guiada, repetida y de manera individual, con una baja incidencia de preguntas por parte del alumnado. En el caso de que haya preguntas, son sobre la regulación de la clase o el chequeo de información. Y esta dinámica se intensifica al haber pruebas importantes, como el SIMCE. Los profesores se enfocan en enseñar el contenido que aparecerá en dichas evaluaciones, usando horas de otras asignaturas como Música, Educación Física o Artes, adelantando contenido y dejando el resto para finales de año o incluso no tienen tiempo para revisarlo [9].

1.1. Problema

La Escuela Prosperidad de Talca¹ es un establecimiento de educación pública donde asisten niños de niveles de educación parvularia y básica, siendo la mayoría de familias uniparentales de bajos recursos. Su profesor de matemáticas y actual profesor jefe del octavo básico A, Leonardo Morales, tiene un gran interés en gamificar sus clases. Dicha idea viene de la popularidad que ha tenido uno de los talleres de la escuela, a su cargo: el taller de habilidades. Dentro de éste, los niños se familiarizan con la programación en base a Scratch² y Arduino³, con estudiantes desde octavo básico hasta kinder. Si bien conoce varias herramientas para enseñar matemáticas, o son suscripciones que la escuela no puede permitirse en su presupuesto, o no son lo suficientemente atractivas para los niños. Leonardo admite que es difícil, en ciertas ocasiones, mantener la atención de los estudiantes solamente utilizando la pizarra y el plumón. Es por esa razón que le gustaría implementar la gamificación en las clases de matemáticas, específicamente en el ramo de geometría, al mismo tiempo que introduce a los estudiantes a la programación.

Este trabajo de memoria busca desarrollar un sistema interactivo basado en actividades o experiencias que ayuden a reforzar el contenido de geometría para los

¹<https://escuelaproperidad.cl/>

²<https://scratch.mit.edu/about>

³<https://www.arduino.cc/en/about/>

estudiantes de octavo básico A, mientras aprenden a programar en el motor de videojuegos Unity a través de Visual Scripting, un paquete que incluye bloques de código hechos que uno sólo tiene que unir y asignar variables, como Scratch. El objetivo es diseñar al menos cinco experiencias que comprendan contenidos de geometría, como la identificación de puntos y la traslación de figuras dentro del plano cartesiano, el cual es el tema que se les presenta en su plan de clases al curso a cargo del profesor Leonardo Morales el próximo año. Luego de crear las experiencias, se probarán en un grupo experimental del curso, seleccionado aleatoriamente. Su rendimiento será medido a través de un script hecho en C# para Unity, que imprimirá el tiempo en completar la experiencia, el número de intentos del estudiante y si tuvo éxito en completar cada una de ellas.

1.1.1. Objetivo General

Desarrollar un prototipo de sistema y actividades de programación para reforzar el desarrollo de competencias matemáticas e introducir competencias básicas de programación en los niños de octavo año básico de la Escuela Básica Prosperidad de Talca.

1.1.2. Objetivos Específicos

Objetivo específico 1: Diseñar cinco experiencias de aprendizaje, basadas en el contenido de matemáticas de Octavo Básico, utilizando Visual Scripting de Unity, con sus respectivas guías y soluciones sobre al menos tres tópicos específicos de geometría (ej. Vectores, Traslación).

Objetivo específico 2: Desarrollar un script en C# en Unity que registre de manera persistente (en formato .txt o .csv) y por estudiante: el tiempo total dedicado a cada experiencia, el número de intentos realizados y el resultado (éxito/fallo) para poder registrar su progreso durante las pruebas.

Objetivo específico 3: Diseñar una evaluación diagnóstica inicial y una evaluación final sobre geometría, además de una encuesta de satisfacción enfocada en la percepción de progreso y usabilidad, para poder comparar los resultados y opiniones de cada estudiante al finalizar las pruebas.

Objetivo específico 4: Realizar la prueba y las evaluaciones inicial y final con la muestra (grupos experimental y control) en un plazo de cuatro semanas, para medir sus conocimientos en geometría, los datos del script necesarios para demostrar la viabilidad y efectividad del proyecto, y la percepción de progreso y usabilidad de la herramienta.

Objetivo específico 5: Analizar y comparar los resultados de la evaluación inicial, la evaluación final y entre el grupo experimental y el grupo control, y los datos de telemetría (tiempo, intentos, éxito) del grupo experimental, para determinar el impacto de las experiencias en el aprendizaje de geometría y el desarrollo de competencias de programación.

1.2. Preguntas de Investigación

A continuación, se presenta la siguiente pregunta de investigación, la cual será respondida según los resultados que surjan del desarrollo de esta memoria:

¿Cómo impacta la implementación de experiencias mediante Visual Scripting en el motor de videojuego Unity en el aprendizaje de contenidos de geometría y competencias de programación en estudiantes de octavo básico de la Escuela Básica Prosperidad de Talca?

Este artículo se estructura de la siguiente forma. En la Sección 2 se presenta el Estado del Arte, que muestra las referencias que sirvieron de inspiración para este proyecto. La sección 3 presenta el Marco Teórico, contextualizando los enfoques actuales del contenido de geometría de octavo básico, las competencias de programación comprendidas en el desarrollo del proyecto y la herramienta principal a utilizar. En la sección 4 se presenta la Metodología, incluyendo el diseño de las actividades y las evaluaciones. La sección 5 presenta la implementación del instrumento de recolección de datos, el plano cartesiano, los resultados obtenidos en las pruebas y su análisis. Finalmente, la sección 6 sintetiza los principales aportes, limitaciones y trabajo a futuro.

2. Estado del Arte

Como se mencionó anteriormente, hay varias aplicaciones y páginas web que se especializan en enseñar matemáticas a través de la gamificación. A continuación, se hará mención de algunos ejemplos:

2.1. Gamificación y su Aplicación en Escuelas de Chile

La gamificación o ludificación es un término derivado del anglicismo *gamification*, que hace referencia a una técnica de aprendizaje que adapta la esencia, el pensamiento, los elementos y la mecánica de los videojuegos a contextos educativos y profesionales con el propósito de mejorar los resultados de aprendizaje [10]. En entornos educativos, la gamificación busca transferir los aspectos positivos de las dinámicas de juego al aula, diseñando tareas y actividades que aprovechen la predisposición natural de los estudiantes hacia el juego para inducir y fortalecer su motivación hacia el aprendizaje [11].

El proceso de gamificación aplicado al uso de videojuegos puede sintetizarse en diez componentes fundamentales: tipos de competición, presión temporal, escasez de recursos, resolución de acertijos, novedad, niveles y progresión, presión social, trabajo en equipo, moneda de cambio y la posibilidad de renovar y aumentar el poder dentro del entorno de juego [12]. Estos elementos permiten mantener un equilibrio entre la dificultad y la accesibilidad de las actividades, evitando que las experiencias de aprendizaje resulten imposibles de superar o excesivamente fáciles y carentes de desafío. Además, los videojuegos ofrecen gran flexibilidad para incorporar estrategias de gamificación [10], incluyendo modalidades como *serious games*, *edutainment*, *advergames*, juegos con propósito y el aprendizaje basado en juegos.

En el contexto chileno, se han desarrollado diversos proyectos de gamificación en establecimientos educacionales, tanto en educación básica como superior, siendo el nivel básico el enfoque de este proyecto. Un ejemplo de ello es el programa ConectaIdeas, impulsado por el Centro de Investigación Avanzada en Educación de la Universidad de Chile [13], que implementó sesiones semanales de aprendizaje de matemáticas en estudiantes de cuarto básico, consistentes en 20 a 30 ejercicios en sesiones de 90 minutos en laboratorios de computación, aplicándose en 11 escuelas de la comuna de Lo Prado, Región Metropolitana. Otro caso relevante es el proyecto Calesca Mat ¡Aprendamos Jugando! en la Región de Antofagasta [14], que abordó todos los contenidos curriculares de matemáticas de tercero básico, tales como adición, sustracción, cálculo mental y fracciones, mediante estrategias lúdicas. Ambos proyectos demostraron impactos positivos en el desarrollo de competencias matemáticas en los estudiantes participantes, evidenciando el potencial de la gamificación como herramienta efectiva en el proceso de enseñanza-aprendizaje.

2.2. GeoGebra

GeoGebra es un software matemático en línea dinámico para todos los niveles que reúne álgebra, hojas de cálculo, gráficos, estadísticas y cálculo en un solo motor [15]. Creado por Markus Hohenwarter en el año 2002 como su trabajo de tesis de la universidad, puede plasmar problemas de geometría, cálculo y álgebra a través de sus diferentes vistas, tanto 2D como 3D, algebraica, de probabilidades y estadísticas, y de cálculo simbólico, como se indica en la figura 2.1. GeoGebra destaca entre las herramientas para enseñar geometría ya que es un software gratuito, libre, de código abierto, de fácil acceso y multiplataforma. Además, es muy fácil de usar, sencillo y potente debido a sus diferentes vistas y representaciones gráficas [16]. Gracias a estas cualidades, GeoGebra es el software preferido tanto por profesores como por estudiantes a la hora de enseñar y aprender, respectivamente, geometría. Para el profesor, es una herramienta didáctica que ayuda en el proceso educativo gracias a GeoGebra Classroom y está respaldada por varios investigadores y distinciones. Para el estudiante, es una de las varias TIC's que tienen a su disposición, que les servirá de apoyo y ayudará de manera positiva a su rendimiento académico [17].

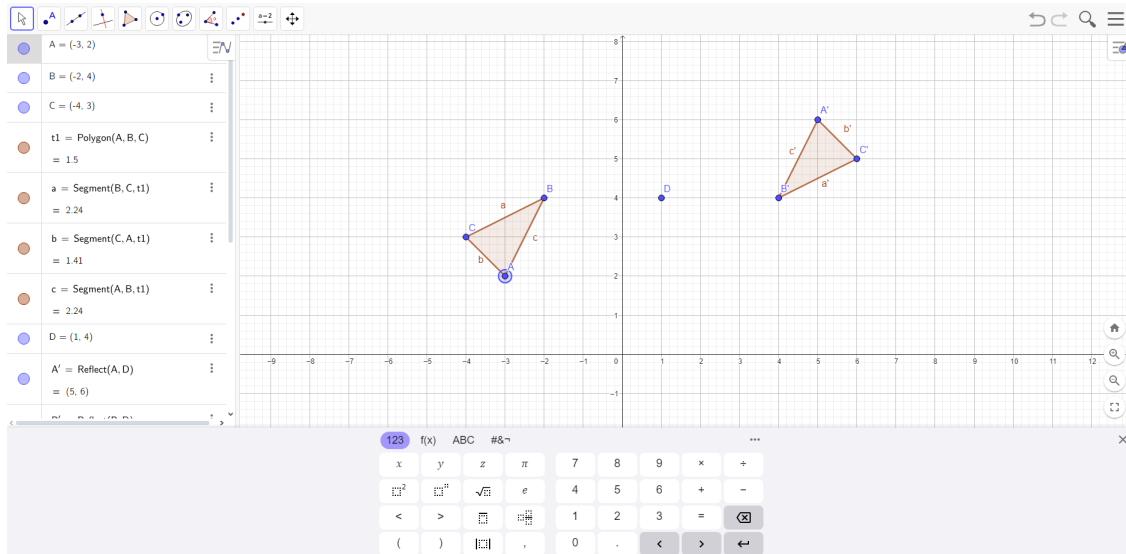


Figura 2.1: Interfaz principal de GeoGebra

2.3. Khan Academy

Khan Academy es una plataforma web creada por Salman Khan en el año 2009 [18]. Khan Academy incluye en su página ejercicios de práctica, videos instructivos y un panel de aprendizaje personalizado, el cual permite a los estudiantes estudiar a su propio ritmo, tanto dentro como fuera de la sala de clases, abarcando desde preescolar hasta educación superior [19]. La colección de videos que existe dentro de Khan Academy no solo abarca matemáticas, sino que también incluye ciencias, lectura, informática, historia, historia del arte, economía y varios otros temas más. La página presenta diferentes secciones, como la zona de práctica, indicada en la figura 2.2, donde se le da al estudiante una serie de ejercicios a resolver, evaluándolo según la cantidad de pistas que haya usado y si tuvo que hacer uso del video explicativo adjunto en la actividad. Y está la biblioteca de videos relacionada con la materia, los cuales se encuentran también en su canal de Youtube.

Como aparece en la figura 2.3, tanto los profesores o tutores a cargo del estudiante, llamados *coaches* dentro de la página, como el alumno tienen total libertad de revisar su progreso a través de gráficos, tablas y logros dentro de Khan Academy [20]. Y cualquiera puede entrar a la plataforma, ya que es una organización sin fines de lucro, gratuita y sencilla de acceder. Solo te hacen falta tus datos personales y ya estás dentro. Gracias a su accesibilidad y su gran catálogo de videos explicativos, sus

resultados como herramienta de reforzamiento para matemáticas son muy positivos y satisfactorios, tanto en niveles medios de enseñanza [21] como a nivel superior [22].

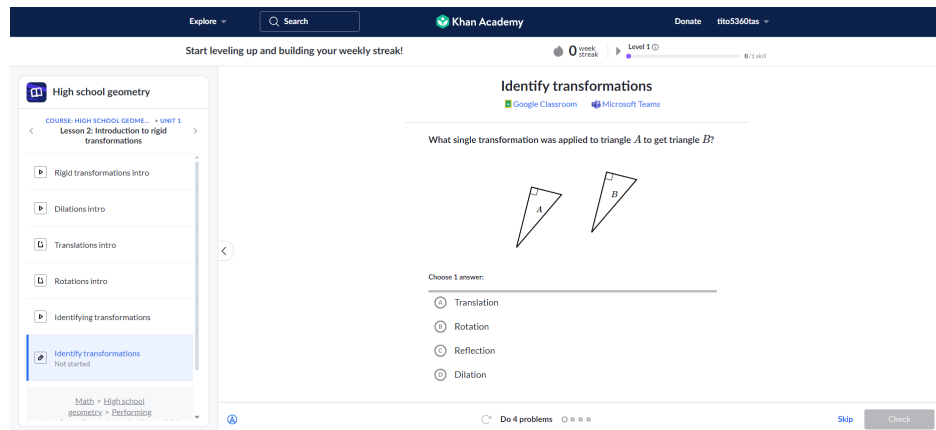


Figura 2.2: Interfaz de la zona de práctica de Khan Academy

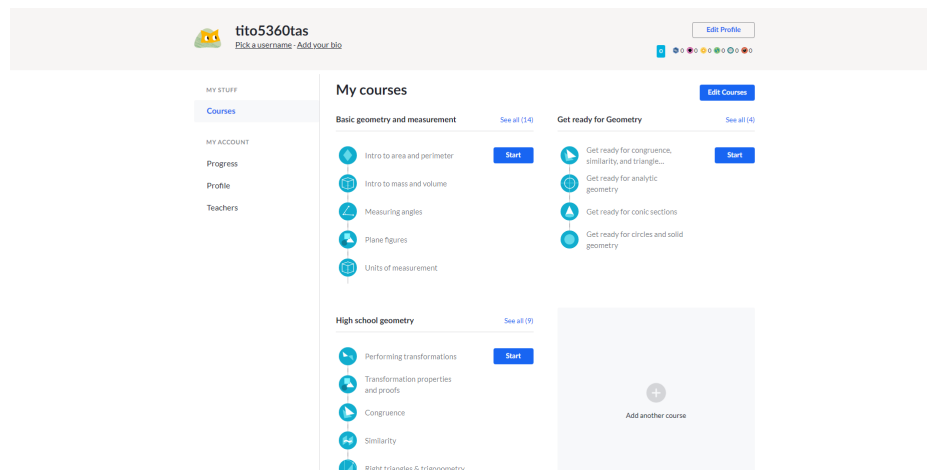


Figura 2.3: Interfaz principal del estudiante de Khan Academy

2.4. Umáximo

Umáximo es una herramienta creada en Chile por Fabián Barriga en el año 2015 como su trabajo de tesis, plataforma que también se encuentra en México, Panamá, Colombia y Perú [6]. Umáximo consiste en una página llamativa en donde están todos los contenidos, el progreso del estudiante y la personalización de su avatar, lo cual actúa como incentivo. Los contenidos se separan en etapas, las cuales se

componen de dos secciones: entrenamiento, como se muestra en la figura 2.4, en donde se presenta la materia al estudiante de manera concisa e interactiva, terminando en un duelo en el que el alumno tiene que responder tres de cinco preguntas de manera correcta para superarlo. La segunda sección se llama práctica, la cual consiste en diez actividades que buscan trabajar las habilidades cognitivas del estudiante de manera personalizada, en base a lo visto en el entrenamiento [23]. El profesor puede asignar tareas dentro de la página, monitorear y calificar dichas tareas, tal como aparece en la figura 2.5, además de otorgar premios en forma de moneda virtual del juego a los estudiantes, la cual les sirve para comprar cosméticos para su avatar.

Umáximo ha tenido éxito dentro de Chile, con varios testimonios de diferentes profesores de diversas zonas del país alabando lo útil, didáctico y entretenido que es [24]. La misma plataforma afirma que el rendimiento académico de los estudiantes aumenta en un 20 % en los primeros dos meses de trabajo con Umáximo. Sin embargo, tiene una barrera de pago que limita todo el contenido descrito en el párrafo anterior. Estos precios pueden ir desde los \$16900 pesos chilenos en el plan de matemáticas por un mes hasta \$49200 por el año completo. No hay más información sobre esta plataforma más allá de su página principal y redes sociales; por lo que no ha habido investigaciones más profundas sobre su rendimiento general.

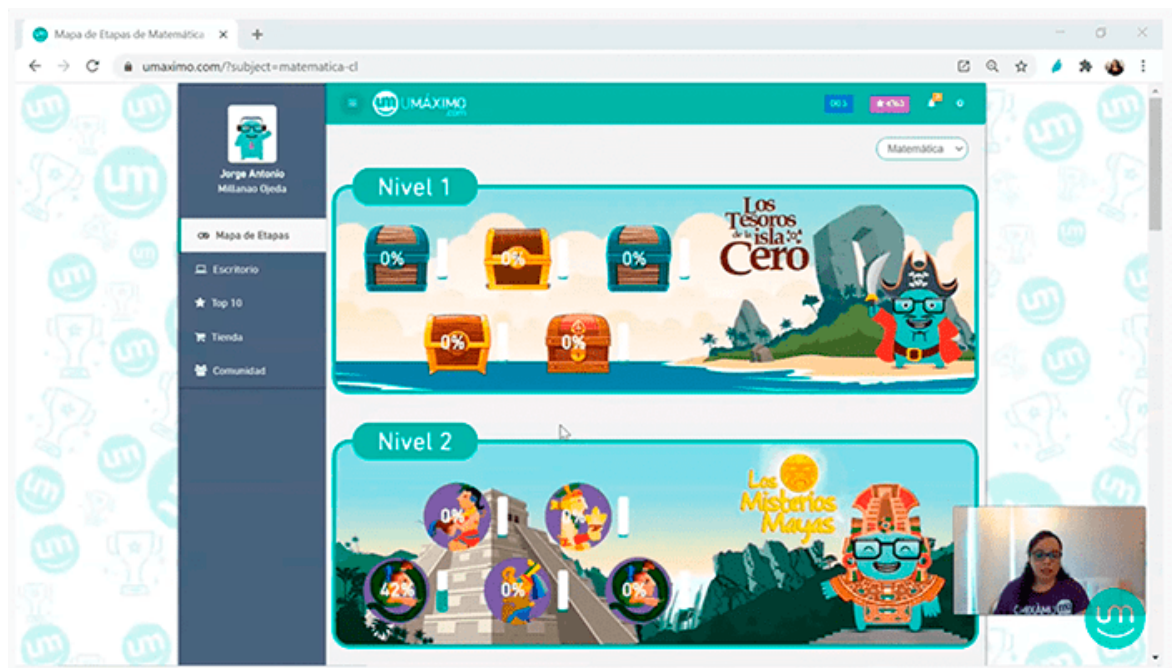


Figura 2.4: Interfaz del estudiante de Umáximo

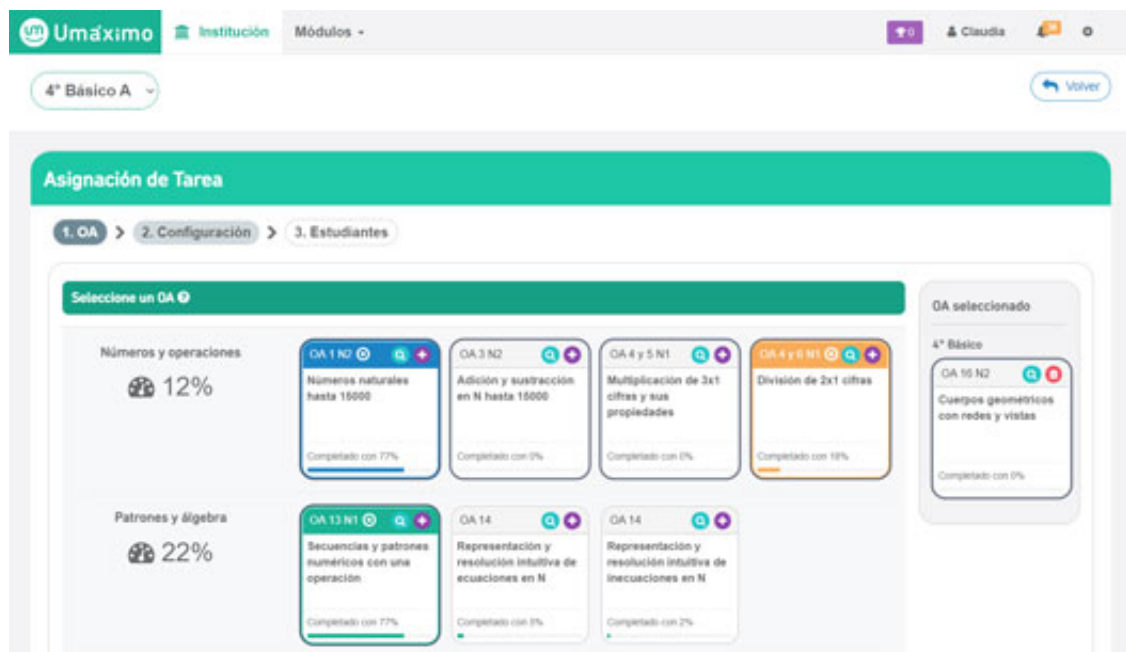


Figura 2.5: Interfaz de asignación de tareas del profesor de Umáximo

3. Marco Teórico

3.1. Competencias de Matemáticas

Los objetivos de la unidad establecidos por el Ministerio de Educación de Chile en el programa de clases de octavo año básico que se verán en este proyecto consisten en lo siguiente [25]:

1. Describir la posición y el movimiento (traslaciones, rotaciones y reflexiones) de figuras 2D, de manera manual y/o con software educativo, utilizando vectores para la traslación, ejes del plano cartesiano como ejes de reflexión, y puntos del plano para las rotaciones.
2. Componer rotaciones, traslaciones y reflexiones en el plano cartesiano y en el espacio, de manera manual y/o con software educativo, y aplicar a las simetrías de polígonos y poliedros, así como a la resolución de problemas geométricos relacionados con el arte.

Entre las habilidades que desarrollarán los estudiantes durante la unidad de geometría se encuentran las siguientes [26]:

1. Utilizar sus propias palabras, gráficos y símbolos matemáticos para presentar sus ideas o soluciones.
2. Explicar y fundamentar tanto las soluciones propias como los procedimientos utilizados, así como los resultados, mediante definiciones, axiomas, propiedades y teoremas.
3. Usar modelos, realizando cálculos, estimaciones y simulaciones, tanto manualmente como con la ayuda de instrumentos, para resolver problemas de otras asignaturas y de la vida diaria.

Por último, se espera que el estudiante demuestre curiosidad, interés, esfuerzo, perseverancia y rigor frente a la resolución de los problemas matemáticos y la búsqueda de nuevas soluciones a éstos [26].

3.2. Competencias de Programación

Según un estudio realizado en la Universidad Goethe de Frankfurt, Alemania, las competencias básicas de programación se resumen en lo siguiente [27]:

- Recordar, comprender y utilizar las funciones básicas de un computador, su arquitectura, compiladores e intérpretes, así como los sistemas informáticos básicos.
- Recordar las herramientas de programación, poder usarlas profesionalmente y justificar y evaluar su uso.
- Recordar, explicar, analizar y evaluar los tipos de datos básicos y las estructuras de datos, su representación, propiedades y aplicación, y la capacidad de crear nuevas estructuras de datos adaptadas a los problemas.
- Recordar, comprender, analizar y evaluar algoritmos simples y patrones de diseño y sus propiedades como solución para problemas conocidos, y ser capaz de diseñar nuevos algoritmos adecuados a problemas dados.
- Recordar, comprender, caracterizar y utilizar las características y construcciones básicas de un lenguaje de programación y su paradigma para generar código de programa.
- Comprender y resolver problemas mediante el desarrollo e implementación de programas adecuados.
- Competencias metacognitivas relacionadas con uno mismo y su implementación de acciones de aprendizaje.
- Competencias no cognitivas, como autocompetencias, habilidades sociocomunicativas, actitud y mentalidad.

Entre las competencias de programación que se quieren enseñar en este proyecto se encuentran, mediante el uso de Visual Scripting, la introducción de los estudiantes a instrucciones de entrada y salida, la secuencialidad de instrucciones, estructuras condicionales y repetitivas, y el uso de variables. Todo eso, utilizando los nodos y herramientas que ofrece Visual Scripting. Además, programar no solo implica las habilidades duras mencionadas anteriormente, sino que también desarrolla habilidades blandas relacionadas con el análisis y la resolución de problemas, la capacidad de comunicarse, enfrentarse a un idioma extranjero (debido a que Visual Scripting solo se encuentra en inglés), saber adaptarse a los cambios, organizar adecuadamente sus ideas y administrar bien su tiempo dedicado al trabajo [28] [29].

3.3. Visual Scripting de Unity

Unity es un motor de videojuego creado en Junio del año 2005 por Nicholas Francis, Joachim Ante y David Helgason, y es una de las plataformas para hacer juegos más populares del mundo, siendo la base de más de la mitad de los videojuegos de cualquier plataforma en el año 2020 [30]. En Mayo del año 2020, Unity Technologies compró la herramienta de secuencias de comandos visuales Bolt, que ahora se llama Visual Scripting de Unity [31], la cual consiste en grafos de flujo donde el usuario puede crear y ejecutar secuencias lógicas a través de nodos, como se puede apreciar en la figura 3.1. El uso de dichos grafos es muy común al crear máquinas de estados para inteligencia artificial.

Unity resalta el uso de Visual Scripting para una rápida implementación de sistemas interactivos, arte y animaciones; empoderar a personas que no programan a crear o modificar proyectos en C# al suavizar la curva de aprendizaje de los conceptos básicos del lenguaje de programación mencionado anteriormente; y crear plantillas y herramientas para compartir con el resto de usuarios de Unity [32]. En el ámbito del desarrollo de videojuegos, los comandos visuales permiten una iteración rápida, un mayor alcance y eficiencia aumentada; permiten a artistas y a desarrolladores indie plasmar ideas y prototipos de manera rápida sin necesidad de tener a un programador dedicado a su lado [30].

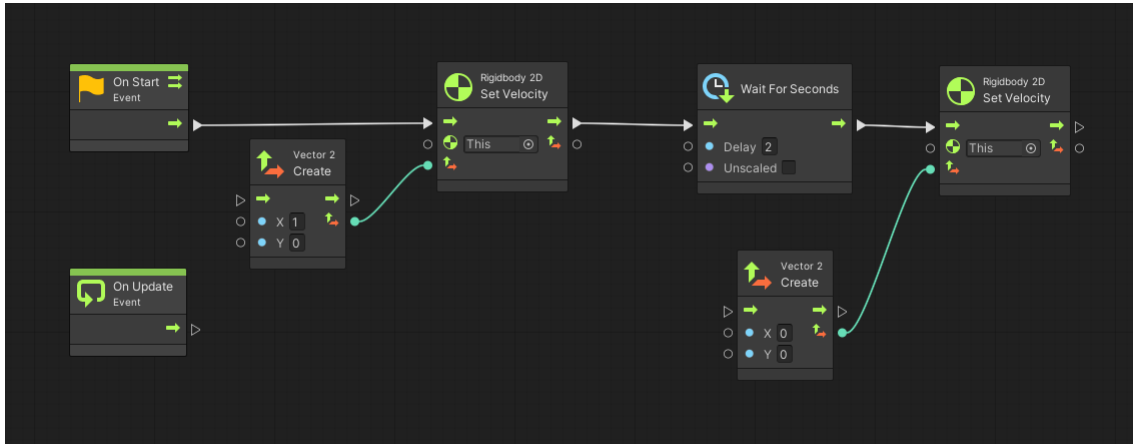


Figura 3.1: Nodos de un Grafo en Visual Scripting de Unity

Teniendo en cuenta todo lo anterior, se necesita realizar un proyecto de una escala menor a los descritos en estos dos capítulos que pueda abordar el contenido de geometría de Octavo Básico, a través de Visual Scripting de Unity en menos de dos meses. Por lo que lo ideal sería hacer al menos cinco experiencias, a través de la metodología de Programación Extrema.

4. Metodología

Para este proyecto se usó Programación Extrema o XP (Extreme Programming) [33]. En dicha metodología, se prioriza la simplicidad; por ende, no es necesario hacer documentos y los requisitos son mínimos; además, es ideal para equipos pequeños. En XP, se programa lo que el cliente pide y luego se prueba para ver si se cumplen los objetivos de las etapas previas del desarrollo del proyecto según la intención de los programadores [34].

La Programación Extrema se basa en cuatro valores: la comunicación, la simplicidad, la retroalimentación y el coraje, es decir, la capacidad de afrontar cualquier problema que surja en el desarrollo del proyecto [35]. Al trabajar con XP, se considera en el desarrollo la disponibilidad del cliente durante todo el proceso, el uso de estándares compartidos por el grupo, la programación dirigida por las pruebas o *Test-driven programming*, la programación en pares, el ritmo sostenido, las integraciones permanentes; en otras palabras, el siempre trabajar en la última versión del proyecto; y la propiedad colectiva del código [36].

Habiendo seleccionado Programación Extrema y adaptándola al desarrollo de las experiencias, se seguirá la siguiente metodología:

- Cada inicio de semana se informará al cliente sobre el avance del proyecto, mostrándole qué se hizo y solicitándole sus comentarios y retroalimentación.
- Luego, se define qué se hará para la siguiente reunión y las correcciones indicadas por el cliente.
- Durante la semana, se trabajará en la tarea asignada para el proyecto y en las sugerencias mencionadas por el cliente.

- Esto se repetirá durante un período de alrededor de un mes, con reuniones cada semana. Como la única persona en el equipo es el desarrollador, él estará encargado de desarrollar el proyecto, reunirse con el cliente y anotar las tareas y correcciones sugeridas.

La Carta Gantt del desarrollo del prototipo y sus detalles se puede ver en el apéndice A al final de este documento.

4.1. Ejercicios y Evaluaciones

Si bien los estudiantes de octavo básico tienen previsto ver transformaciones isométricas en su contenido de geometría, aplicar esto a Visual Scripting puede ser complicado, ya que necesitan tener conocimiento de su entorno de trabajo, el cual será en su mayoría el entorno de Script Graph; qué nodos necesitan y qué hace cada uno, y organizar bien el orden de éstos. Por lo que lo ideal es empezar con ejercicios simples para familiarizarse con el motor de videojuegos, revisando contenido de geometría que ya hayan visto antes, como identificar y posicionar puntos, vectores y figuras en el plano cartesiano. Con eso, se crean tres ejercicios: crear un punto, una línea y un triángulo dentro del plano cartesiano.

Por otra parte, para poder enseñar a los niños ciertos conceptos de programación, como suma, resta, multiplicación y división de variables o ciclos, se tuvo que tomar materia de enseñanza media que aún no han visto; en este caso, suma y multiplicación de vectores, para hacer dos ejercicios más. Se habló con el profesor sobre dicho problema y dió autorización para realizar las experiencias. Como ejercicio final, los estudiantes aplicarán una traslación a un triángulo, combinando los contenidos vistos durante las pruebas.

Las evaluaciones se realizaron de acuerdo con las competencias de matemáticas descritas anteriormente, utilizando *Google Forms* para almacenar los datos obtenidos por correo electrónico. Los ejercicios consistieron en combinar dichos contenidos con conceptos básicos de programación; siendo más específicos, estos son declaración, asignación y almacenamiento de variables, flujo de nodos, dibujar gráficos dentro del motor de juego, operadores aritméticos y ciclos.

Antes de realizar un ejercicio, se presentará un ejemplo paso a paso realizado por el investigador, en una escena aparte donde no se recopila ningún dato, lo que

permite a los estudiantes probar los nodos sin ninguna presión. Los contenidos fueron presentados de la siguiente forma:

- Para los ejercicios uno, dos y cinco, se enseñará cómo declarar y asignar variables, cómo funciona el flujo de nodos y cómo dibujar gráficos dentro del motor de juego, a través de la identificación y posicionamiento de puntos, vectores y figuras en el plano cartesiano. Los estudiantes deben ingresar un valor a las variables, en este caso, los puntos, y asignarlos a las coordenadas de la línea o figura para que se puedan dibujar en el plano cartesiano, como se muestra en el diagrama de flujo de la figura 4.1. Como se observa en la solución del ejercicio dos en la figura 4.2, se creó un nodo personalizado que agrupa el o los *LineRenderer* necesarios para dibujar una línea o una figura, indicado en la figura 4.3, además de usar *GameObject Instantiate* para instanciar el objeto de manera correcta en el plano cartesiano.

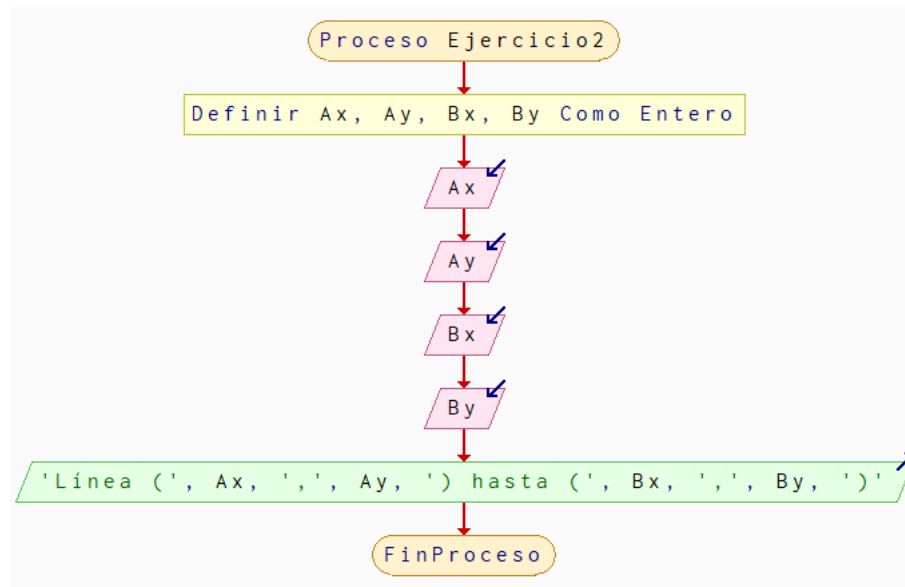


Figura 4.1: Diagrama de Flujo del Ejercicio 2

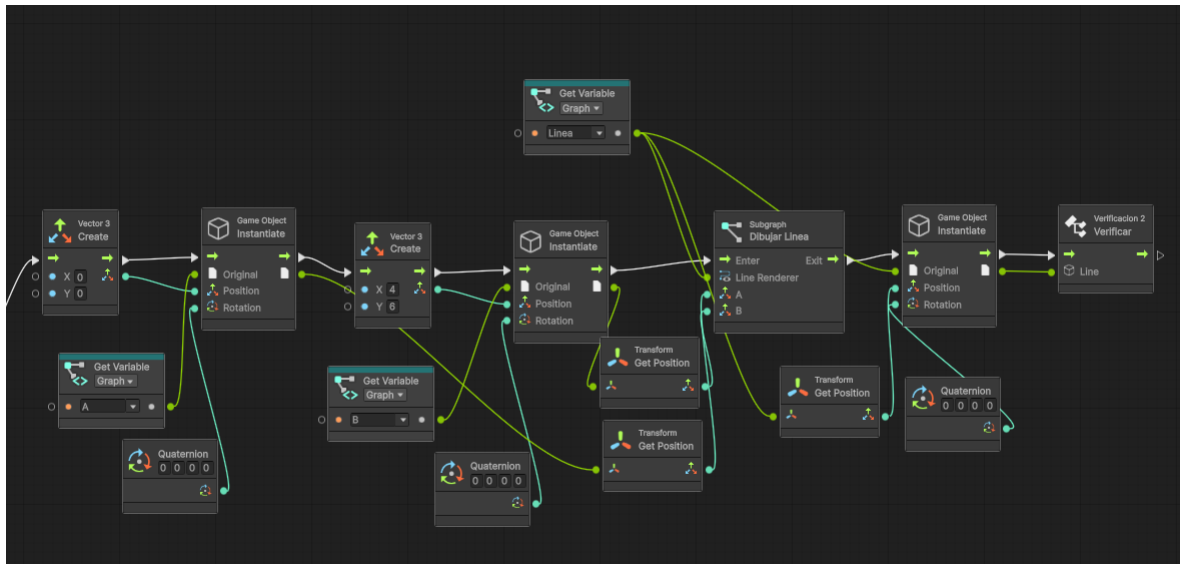


Figura 4.2: Ejercicio 2: Instanciamiento de puntos y línea en el plano cartesiano en Visual Scripting

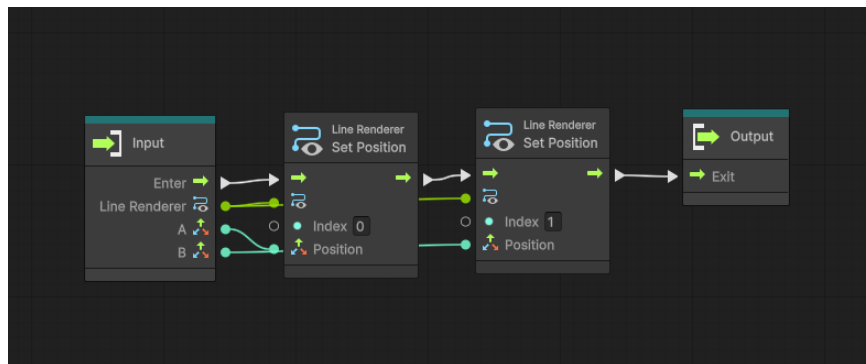


Figura 4.3: *DibujarLínea*: Nodo personalizado para Dibujar una Línea

- Para el ejercicio tres, se enseñará el uso de operadores lógicos para poder realizar, en este caso, sumas de vectores. Los estudiantes, igual que en el ejercicio anterior, deben ingresar valores a sus variables, en este caso, a sus vectores. Como se les indica que ambos vectores empiezan en (0,0), lo único que deben hacer es sumar las coordenadas X e Y de ambos puntos y luego designarlas al *Line Renderer* de línea para dibujarlos, como se observa en el diagrama de flujo de la figura 4.4. La solución del ejercicio se muestra en la figura 4.5.

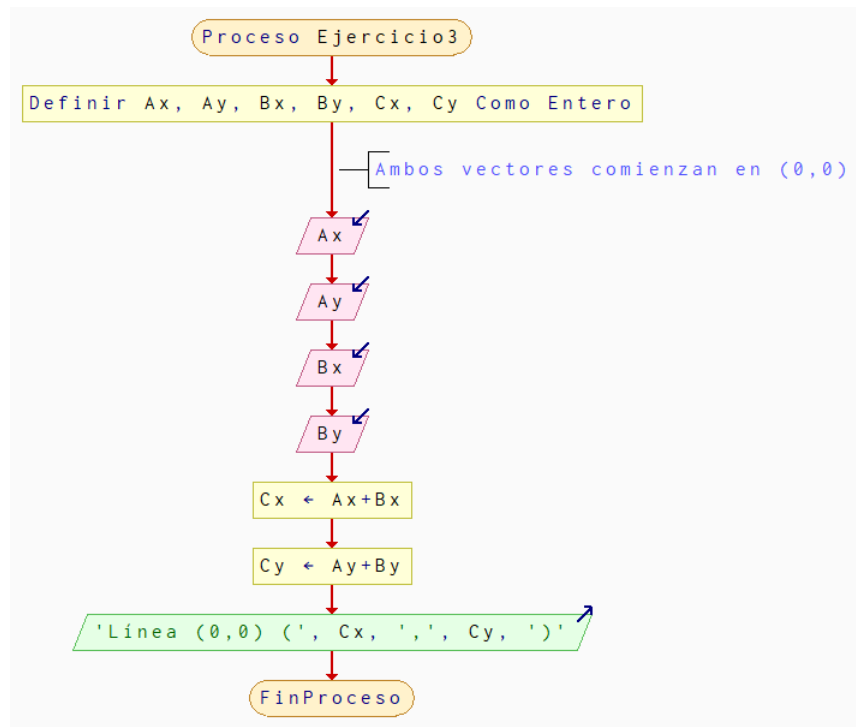


Figura 4.4: Diagrama de Flujo del Ejercicio 3

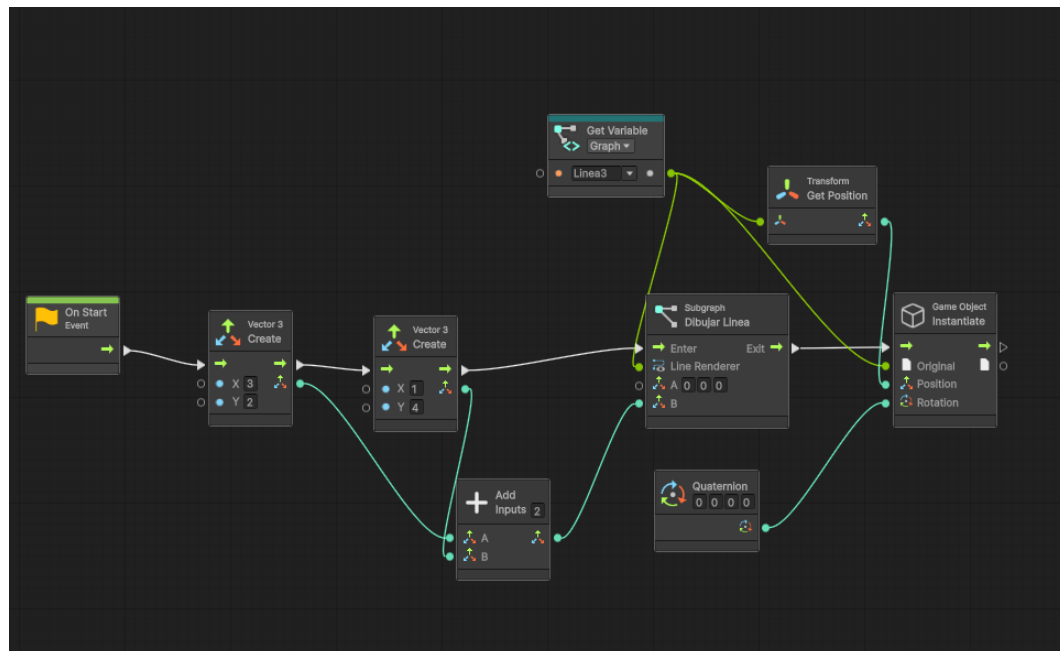


Figura 4.5: Ejercicio 3: Suma de Vectores en Visual Scripting

- Para el ejercicio cuatro, se enseñará el uso de ciclos, específicamente el *For Loop*, y el almacenamiento de variables para realizar la multiplicación de vectores por un escalar. También se podría usar el operador lógico de multiplicación en este ítem, pero se prefirió utilizar este método para ampliar la cantidad de conceptos de programación a enseñar. Los estudiantes declaran su vector, el cual empieza en (0,0), sumando su valor en Bx y By, i veces (en este caso, tres), y en cada iteración, se guardan dichas variables y se imprime la línea. Al llegar a la condición, el ciclo termina y queda la línea final, como se observa en el diagrama de flujo de la figura 4.6. Su solución se muestra en la figura 4.7.

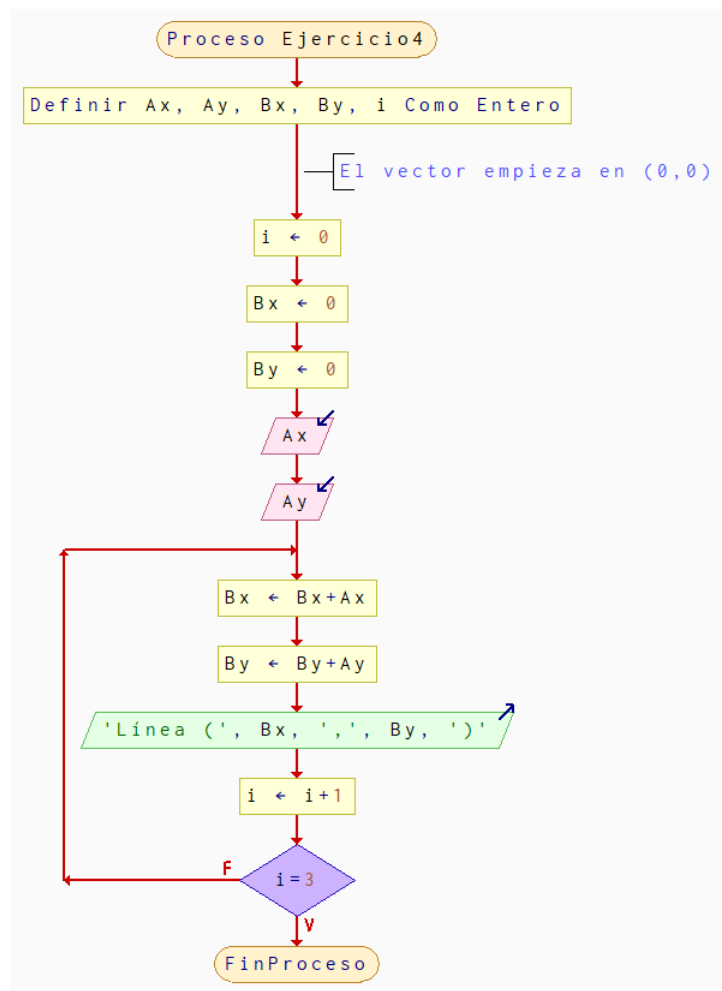


Figura 4.6: Diagrama de Flujo del Ejercicio 4

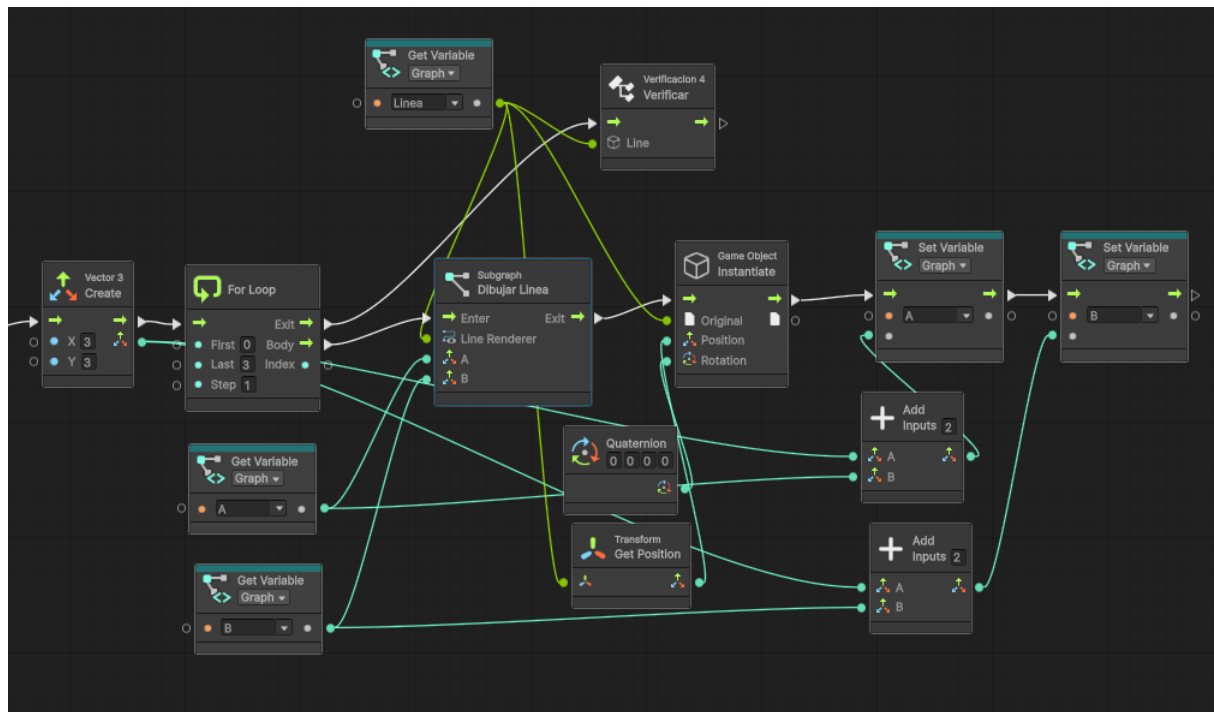


Figura 4.7: Ejercicio 4: Multiplicación de un Vector por un Escalar en Visual Scripting

- Para el ejercicio seis, se enseñará cómo trasladar una figura en el plano cartesiano, combinando la suma de vectores con el dibujo de un triángulo. Los estudiantes ingresan las coordenadas de los vértices del triángulo, luego definen el vector a sumar. Después, se suma dicho vector con cada vértice y finalmente se dibuja el triángulo, como se muestra en el diagrama de flujo de la figura 4.8. En la solución expuesta en la figura 4.9, aparece el nodo personalizado *Dibujar Triángulo*, que es similar al nodo mostrado en la figura 4.3, solo que tiene un *LineRenderer* adicional.

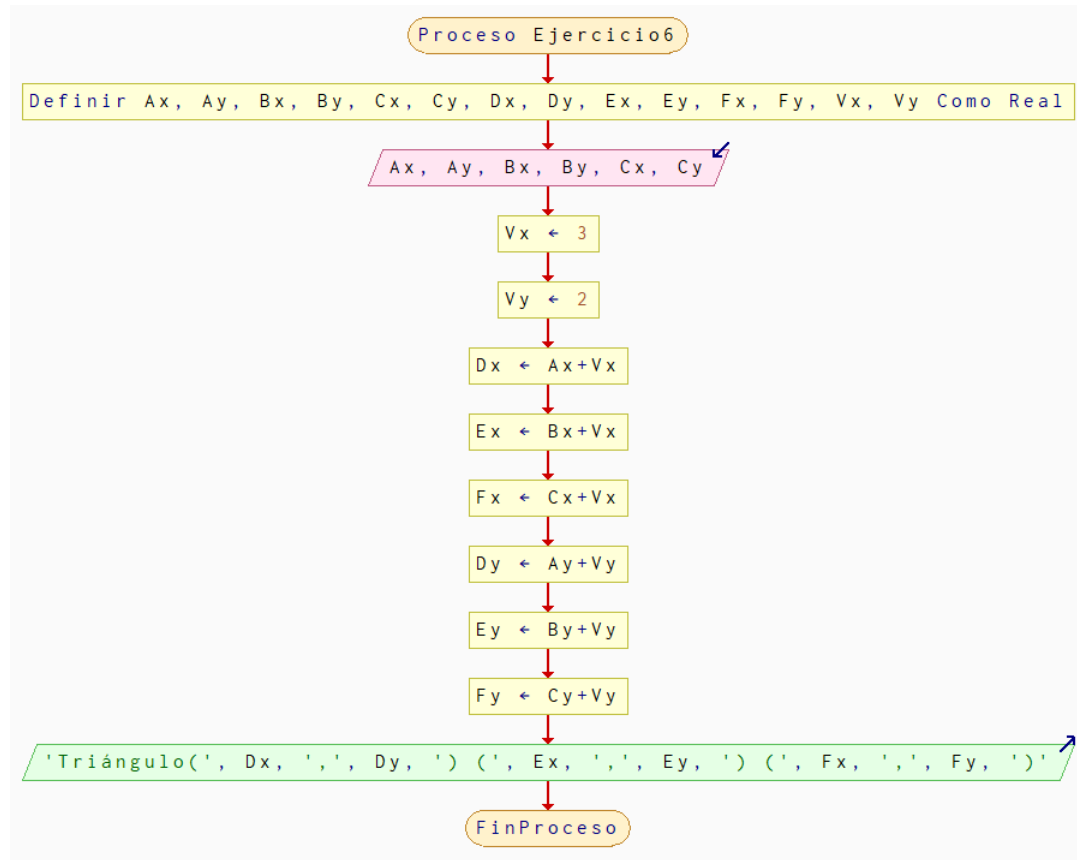


Figura 4.8: Diagrama de Flujo del Ejercicio 6

Por último, la encuesta de satisfacción se realizó utilizando la plataforma *SurveyMonkey* para tener ordenados los resultados en diferentes gráficos. Ésta se realizó en base a la escala de Likert de cinco puntos para recopilar las opiniones de los participantes, después de haber terminado las actividades.

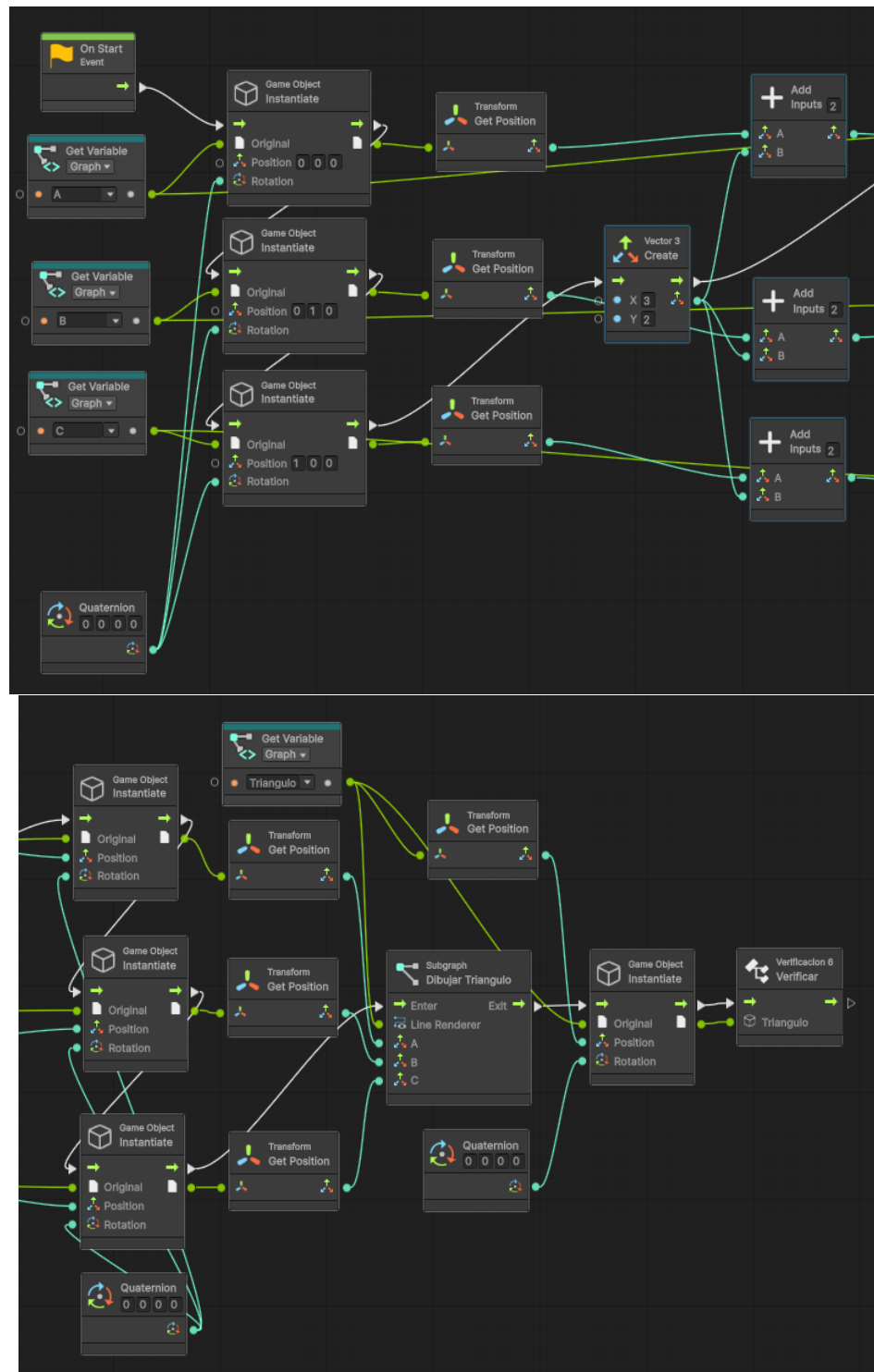


Figura 4.9: Ejercicio 6: Traslación de un Triángulo en Visual Scripting

5. Implementación y Resultados

5.1. Implementación

Para poder empezar a desarrollar el prototipo, se necesitó hacer los scripts que se van a ejecutar al iniciar las experiencias, los cuales contienen cómo se va a dibujar el plano cartesiano al ejecutar el juego y cómo se van a recolectar y verificar las respuestas de los usuarios al completar las experiencias, los cuales se hicieron en C# dentro del mismo motor de desarrollo de videojuegos Unity.

5.1.1. Plano Cartesiano

El código se compone principalmente de dos funciones: *CreateLine()* y *CreateAxisLabel()*. La primera función, como se muestra en el Listing 5.1.1, crea un objeto y le agrega y configura el componente *Line Renderer* para poder hacer uno de los ejes y las líneas paralelas a éste. La segunda función, a través de un ciclo, crea un objeto con un componente llamado *TextMeshPro*, el cual es una etiqueta de texto. Ésto es para colocar los números de cada coordenada en su respectiva posición en el eje.

```
1 void CreateLine(Vector3 start, Vector3 end, Color color, float width, string name)
2 {
3     GameObject lineObj = new GameObject(name);
4     lineObj.transform.parent = transform;
5
6     LineRenderer lr = lineObj.AddComponent<LineRenderer>();
7     lr.positionCount = 2;
8     lr.SetPositions(new Vector3[] { start, end });
9     lr.material = lineMaterial ??
10     new Material(Shader.Find("Sprites/Default"));
11     lr.startColor = lr.endColor = color;
12     lr.startWidth = lr.endWidth = width;
13     lr.sortingOrder = 1;
14 }
15
```

Listing 5.1: Método para hacer líneas en pantalla

5.1.2. Verificación y Extracción de Datos

Los datos que se van a recolectar, como se ha mencionado anteriormente, son el tiempo en que se ha completado la actividad, a través de un *string* en el que se convierte la hora actual a texto; la cantidad de intentos, a través de un *int* guardado dentro de *EditorPrefs*, ya que es una variable persistente dentro del desarrollo de las experiencias; y si el usuario completó o no la actividad, la cual es una variable *booleana*. La verificación y extracción de datos se realiza dentro del mismo código que se muestra en el Listing 5.1.2 a través de la función *Verificacion()*, la cual, dependiendo del ejercicio a resolver, tendrá diferentes condiciones iniciales, ya sea pidiendo una coordenada en el caso de un punto o un conjunto de coordenadas dentro de un *LineRenderer*, en el caso de una línea o una figura, comparándose con la respuesta a obtener. Al cumplirse la condición, se manda a imprimir un archivo de texto con las variables mencionadas al inicio y se guarda dentro de la carpeta en donde se encuentra la experiencia.

```

1 public static void Verificar(GameObject punto)
2 {
3     Vector3 posicionPunto = punto.transform.position;
4     Vector3 posicionObjetivo = new Vector3(2f, 5f);
5
6     if (Vector3.Distance(posicionPunto, posicionObjetivo) < 0.01f)
7     {
8         aprobado = true;
9     }
10
11     if (aprobado && !yaRegistrado)
12     {
13         string path = Application.dataPath + "/Ejercicios/1/Ex 1/Ejercicio1.txt";
14         string currentTime = DateTime.Now.ToString("dd-MM-yyyy HH:mm:ss");
15
16         // Determina el estado de aprobacion como texto
17         string estado = aprobado ? "APROBADO" : "PENDIENTE";
18
19         // Crea el mensaje que se anexara al archivo
20         string message =
21             $"-----\n" +
22             $"Fecha y Hora: {currentTime}\n" +
23             $"Intento No.: {_conteoPlay}\n" + // Valor del contador
24             $"Resultado: El ejercicio ha sido {estado}.\n" +
25             $"-----\n\n";
26         File.AppendAllText(path, message);
27
28         yaRegistrado = true;
29     }
30 }
31

```

Listing 5.2: Método para verificar el ejercicio 1

Para poder usar la función de verificación dentro de los grafos, se tuvo que hacer cada clase estática y agregarlas a las opciones de tipo de nodos dentro de las configuraciones de Visual Studio, para que así puedan ser accesibles dentro de cualquier parámetro del motor.

De manera conjunta, existen dos funciones de control en cada script de verificación: *ReiniciarConteo()*, que reinicia la variable de intentos en cada experiencia a cero, e *ImprimirInforme()*, la cual genera de manera forzosa el archivo de texto si el usuario no es capaz de completar la actividad en el horario correspondiente. Ambos están dentro de un *MenuItem* nuevo llamado "Herramientas".

5.2. Resultados

Las pruebas empezaron oficialmente el 8 de octubre y terminaron el 12 de noviembre de 2025. Los participantes firmaron su respectivo consentimiento informado; por lo tanto, sus identidades serán anónimas para esta investigación y respondieron la prueba diagnóstica el primer día. Los resultados de los participantes que fueron seleccionados para el taller se presentan en la Tabla 5.1. Las clases se dividieron entre los estudiantes que fueron escogidos para las pruebas, supervisados por el investigador, y el resto que estaba a cargo del profesor Leonardo, realizando otras actividades acordes a la asignatura de TIC's, como se ve en la figura 5.1. Luego, se les introdujo al ambiente de programación del motor de juego, el cual es el grafo donde van a trabajar con los nodos y la pantalla de *Game*, donde verían lo que habían programado anteriormente. Como actividades introductorias, tanto el primer como el segundo día, se les pidió posicionar tanto puntos como vectores en el plano cartesiano, los cuales realizaron con entusiasmo e interés, algunos más rápido que otros, lo que se verá más reflejado en las siguientes actividades.

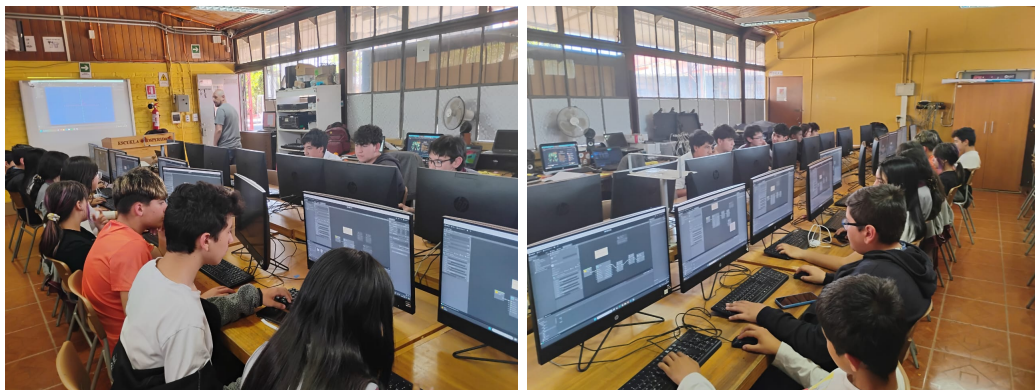


Figura 5.1: Estudiantes del Octavo Básico A de la Escuela Prosperidad de Talca realizando las experiencias. Su profesor, Leonardo Morales, supervisa la otra mitad que no está participando en los talleres

La siguiente semana lograron sumar la magnitud de dos vectores para poder crear uno con dicho resultado. También en la clase del 5 de noviembre pudieron multiplicar un vector por un escalar dentro del plano cartesiano. Y aquí vienen ciertas observaciones, según los datos recopilados en las Tablas 5.2 y 5.3. Si bien todos lograban, al final de la clase, realizar el ejercicio, había dos tipos de estudiantes en el taller: los que intentaban, por prueba y error, cada vez que tenían una secuencia de nodos, y

Diagnóstico				
Estudiante	Pregunta 1 (Puntos en el plano)	Pregunta 2 (Suma de vectores)	Pregunta 3 (Multipli- cación de vectores por un escalar)	Pregunta 4 (Traslación de figura en el plano)
1	4/4	0,5	0	2/3
2	0/4	0,5	0,5	1/3
3	3,5/4	0	0	1/3
4	2/4	0	0	1/3
5	3/4	0	0	3/3
6	4/4	0	0	0/3
7	1/4	0	0	0/3
8	2/4	0	0	0/3
9	3/4	0	0	0/3

Tabla 5.1: Tabla de resultados de pruebas diagnósticas

los que esperaban que algún compañero, el profesor o el instructor los guiase, sin dar directamente la respuesta al ejercicio. Hay algunos niños que resolvían los problemas rápidamente, aunque con varios intentos, y otros se tomaban su tiempo o preguntaban mucho tanto al profesor como al instructor, con un sólo intento a su haber. Lo importante es que todos entendían qué hacía cada nodo y cómo llegar al resultado, aunque había que hacer una retroalimentación cada semana para refrescarles los contenidos vistos en las clases anteriores.

Ejercicio 3			
Estudiante	Intentos	Tiempo	Aprobado
1	7	16:28	Sí
2	7	17:57	Sí
3	1	27:51	Sí
4	1	31:17	Sí
5	1	35:28	Sí
6	4	21:52	Sí
7	1	26:02	Sí
8	1	38:24	Sí
9	9	26:28	Sí

Tabla 5.2: Tabla compilatoria de resultados del ejercicio 3.

En la última clase hubo una baja asistencia de los estudiantes que estaban proban-

Ejercicio 3			
Estudiante	Intentos	Tiempo	Aprobado
1	3	14:08	Sí
2	3	15:37	Sí
3	1	18:26	Sí
4	1	31:17	Sí
5	1	30:28	Sí
6	1	15:25	Sí
7	1	23:28	Sí
8	1	34:11	Sí
9	8	24:50	Sí

Tabla 5.3: Tabla compilatoria de resultados del ejercicio 4.

do las experiencias, por lo que hubo pocos datos con los que comparar los diagnósticos iniciales. Viendo la Tabla 5.4, si bien hay algunos estudiantes que siguen confundiendo al identificar coordenadas en el plano cartesiano, las preguntas que no lograron responder en la prueba diagnóstica las respondieron de manera correcta. Por lo tanto, implementar ejercicios de programación en el entorno de matemáticas, en este caso geometría, sí ayuda a que puedan entender la materia mejor.

Examen Final				
Estudiante	Pregunta 1 (Puntos en el plano)	Pregunta 2 (Suma de vectores)	Pregunta 3 (Multipli- cación de vectores por un escalar)	Pregunta 4 (Traslación de figura en el plano)
3	1/4	1	1	3/3
5	3,5/4	1	1	3/3
6	4/4	1	1	3/3
8	2/4	1	1	3/3

Tabla 5.4: Tabla de resultados de pruebas finales

Los resultados de la encuesta de satisfacción, los cuales estarán en el anexo B en forma de gráficos, indican que a la mayoría de los participantes les gustó el taller y las actividades, declarando que los objetivos tanto de programación como los de matemáticas se explicaron bien y que las actividades fueron interesantes para los estudiantes. Entre los comentarios hechos por los niños, como se ve en la figura 5.2, se confirma lo que se expresó anteriormente; incluso uno de ellos pide una clase más

avanzada de programación en código en vez de visual. Varios de ellos recomendarían el taller a familiares y amigos.

¿Hay algo que quieras compartir sobre el taller? Deja tus comentarios abajo

Respondidas: 7 Omitidas: 1

Respuestas (7)

NUBE DE PALABRAS

Etiquetas (0)



Buscar respuestas



Filtrar: por etiquet



Se muestran 7 respuestas



Muy buen taller y facil

12-11-2025 03:33 p. m.

[Ver respuesta completa](#)

[Agregar eti](#)



De que es muy entretenido y es muy facil de aprender

12-11-2025 03:33 p. m.

[Ver respuesta completa](#)

[Agregar eti](#)



que me gusto mucho y me ayudo a aprender mas sobre las programaciones visuales

12-11-2025 03:26 p. m.

[Ver respuesta completa](#)

[Agregar eti](#)



no, gracias

12-11-2025 03:26 p. m.

[Ver respuesta completa](#)

[Agregar eti](#)



Fue muy entretenido, aunque a veces me pierdo mucho

Figura 5.2: Comentarios de la Encuesta de Satisfacción

6. Conclusión

El propósito de esta memoria es determinar si, a través de experiencias gamificadas, se podían enseñar tanto matemáticas como conceptos de programación a los estudiantes de octavo básico de la Escuela Prosperidad de Talca, usando Visual Scripting del motor de juego Unity. Luego, se desarrollaron dichas experiencias en conjunto con su profesor y se pusieron en marcha las pruebas y la recopilación de datos a través de un script de C# dentro de las actividades.

Sobre los objetivos descritos en la memoria, tanto los objetivos uno, dos y tres se lograron sin problemas, incluso alcanzando a realizar seis experiencias en lugar de las cinco descritas en el primero. Sin embargo, los objetivos cuatro y cinco se lograron de manera parcial; primero, debido a interrupciones que no estaban previstas en la planificación del proyecto, lo que dejó uno de los ejercicios sin realizar. Segundo, aunque sí se pudieron analizar los resultados, no se pudieron comparar entre el grupo experimental y el grupo control, en virtud de que los estudiantes del último grupo no respondieron la evaluación final, ya que les asignaron otras actividades esa semana.

Sobre la respuesta a la pregunta de investigación, según lo entregado en las tablas 5.1, 5.2, 5.3 y 5.4, sí hubo un impacto positivo y una mejora significativa en la enseñanza de conceptos de geometría en los estudiantes de octavo básico, a través de la introducción de conceptos introductorios de programación y el uso de programación visual como medio de enseñanza.

Finalmente, el éxito alcanzado en la mayoría de los objetivos propuestos confirma que la programación visual y la geometría pueden reforzarse mutuamente. La experiencia en la Escuela Prosperidad de Talca deja en evidencia que los estudiantes de octavo básico pueden enfrentar desafíos de programación visual, siempre que el diseño instruccional sea colaborativo y situado.

6.1. Trabajo a Futuro

Este trabajo se puede considerar más como un prototipo que como un producto más detallado, no por el hecho de que hubo una sola persona involucrada, sino porque se pudo hacer más con este proyecto. Pudo enfocarse en ser un *Serious Game* con una mascota, metas, logros y otras características que poseen otras plataformas, como Khan Academy o Umáximo; o bien ser una serie de talleres donde las matemáticas y la programación se combinan para hacer más llamativo el enseñar algo tan denso para algunas personas como son las matemáticas, como se hizo en esta tesis. El éxito significativo de este proyecto sólo indica que se puede seguir adelante, con un proyecto a mayor escala, considerando otros colegios, cursos o al público general.

6.2. Palabras Finales

En estos últimos años, ha habido varios eventos que han hecho que el proceso de tesis sea una carga considerable, en lugar de una oportunidad para hacer algo que pueda destacar dentro del ámbito de los videojuegos y/o la programación. Pérdidas, arrepentimientos, ideales personales que descendieron abruptamente y cambios de rumbo. Hubo un momento durante el desarrollo de esta tesis en el que me decía que no lo iba a lograr, ni siquiera a llegar al promedio. Pero la prueba está aquí. No será una obra maestra; sin embargo, esta tesis me dice que sí puedo. El apoyo de varios familiares, el profesor Leonardo e incluso los chicos con los que trabajé, quienes me preguntaban sobre las experiencias de un próximo ingeniero en desarrollo de videojuegos y realidad virtual, no buscaba seguir directamente mis pasos, sino que estaban interesados en estudiar algo similar a lo que hago yo. Hay otras cosas con las que quiero seguir y empezar; no obstante, no implica que este proyecto se quede en la nada. Ésto puede seguir mejorando, al igual que mi confianza en mí mismo. Sólo hay que seguir intentando.

Bibliografía

- [1] Pere Cornellà, Meritxell Estebanell, and David Brusi. Gamificación y Aprendizaje Basado en Juegos. *Enseñanza de las Ciencias de la Tierra*, 28(1):5–19, 2020.
- [2] G. Zichermann and C. Cunningham. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O’Reilly Series. O’Reilly Media, 2011.
- [3] Claudia Jaramillo and Jorge Chávez. TIC y Educación en Chile: Una Revisión Sistemática de la Literatura. 2015.
- [4] Pere Ramon Marquès Graells. Impacto de las TIC en la Educación: Funciones y Limitaciones. *3 c TIC: cuadernos de desarrollo aplicados a las TIC*, 2(1):2, 2013.
- [5] Katibe Gizem Yiğ and Sezan Sezgin. An Exploratory Holistic Analysis of Digital Gamification in Mathematics Education. *Journal of Educational Technology and Online Learning*, 4(2):115–136, 2021.
- [6] Umáximo. Umáximo — Nosotros. <https://www.umaximo.com/web/nosotros>. Último acceso 28 de Junio de 2025.
- [7] Agencia de Calidad de la Educación. SIMCE - Agencia de Calidad de la Educación. Presentación Resultados Educativos 8° Básico. https://archivos.agenciaeducacion.cl/PPT_Nacional_Resultados_educativos_2019.pdf, 3 2020.
- [8] Darinka Radovic and David Preiss. Patrones de Discurso Observados en el Aula de Matemática de Segundo Ciclo Básico en Chile. *Psykhē (Santiago)*, 19(2):65–79, 2010.

- [9] Carolina Ruminot Vergara. Los Efectos Adversos de una Evaluación Nacional sobre las Prácticas de Enseñanza de las Matemáticas: El Caso de SIMCE en Chile. *Revista Iberoamericana de Evaluación Educativa*, 10(1):69–87, 2017.
- [10] Mery Luz Pacheco Bohórquez Robert and Eduardo Causado Escobar. El Aprendizaje Basado en Videojuegos y la Gamificación como Estrategias para Construir y Vivir la Convivencia Escolar. *CEDOTIC Revista de Ciencias de la Educación, Docencia, Investigación y Tecnologías de la Información*, 3(1):59–80, 2018.
- [11] Diego S Melo-Solarte and Paula A Díaz. El Aprendizaje Afectivo y la Gamificación en Escenarios de Educación Virtual. *Información tecnológica*, 29(3):237–248, 2018.
- [12] Daniel Parente. Gamificación en la Educación. *Gamificación en aulas universitarias*, 11(15):14, 2016.
- [13] Roberto Araya, Elena Arias Ortiz, Nicolas L Bottan, and Julian Cristia. ¿Funciona la Gamificación en la Educación? Evidencia Experimental de Chile. Technical report, IDB Working Paper Series, 2019.
- [14] Guillermo A Guevara, Leslie C Madariaga, Camila A Reyes, and Carolina A Zuleta. Gamificación para el Desarrollo del Aprendizaje de las Operaciones Matemáticas en Tercero Básico. *Información tecnológica*, 34(4):31–44, 2023.
- [15] Geogebra. About Geogebra. <https://www.geogebra.org/about>. Último acceso 28 de Junio de 2025.
- [16] Eloy Arteaga Valdés, Juan Felipe Medina Mendieta, and Jorge Luis del Sol Martínez. El Geogebra: una Herramienta Tecnológica para Aprender Matemática en la Secundaria Básica Haciendo Matemática. *Conrado*, 15(70):102–108, 2019.
- [17] Fredy Barahona AVECILLA, Olga Barrera Cárdenas, Byron Vaca Barahona, and Blanca Hidalgo Ponce. Geogebra para la Enseñanza de la Matemática y su Incidencia en el Rendimiento Académico Estudiantil. *Revista Tecnológica-ESPOL*, 28(5), 2015.
- [18] Marc Prensky. Khan Academy. *Educational Technology*, 51(5):64, 2011.
- [19] KhanAcademy. Khan Academy. <https://es.khanacademy.org/about>. Último acceso 28 de Junio de 2025.

- [20] KhanAcademy. Khan Academy. <https://es.khanacademy.org/>. Último acceso 17 de Noviembre de 2024.
- [21] Ana Teresa Antequera Guerra. Khan Academy: Una Experiencia de Aula en Secundaria. *Números*, 83:199–209, 2013.
- [22] LUIS ADRIAN LASSO CARDONA and Karen Natalia Conde Rodriguez. Khan Academy como Herramienta en el Aprendizaje de las Matemáticas y la Programación. 14:225–250, Jan. 2021.
- [23] Umáximo. Umáximo — Cómo Funciona — Demostración. <https://www.umaximo.com/web/comofunciona>. Último acceso 28 de Junio de 2025.
- [24] Umáximo. Umáximo — Testimonios — Casos de Éxito. <https://www.umaximo.com/web/testimonios>. Último acceso 28 de Junio de 2025.
- [25] Ministerio de Educación. Programa de Estudio Matemática Octavo Básico. https://www.curriculumnacional.cl/614/articles-18983_programa.pdf, 2016.
- [26] Ministerio de Educación de Chile. Unidad 3: Teorema de Pitágoras. Prismas y Cilindros. Posición y Movimientos. <https://www.curriculumnacional.cl/docentes/Educacion-General/Matematica/Matematica-8-basico/21191:Unidad-3-Teorema-de-Pitagoras-Prismas-y-cilindros-Posicion-y-movimientos>. Último acceso 28 de Junio de 2025.
- [27] Natalie Kiesler. On Programming Competence and Its Classification. In *Proceedings of the 20th koli calling international conference on computing education research*, pages 1–10, 2020.
- [28] Dana Sorana Urs. Soft Skills for the Engineering Students. *Synergy*, (2):137–142, 2013.
- [29] Faheem Ahmed, Luiz Fernando Capretz, Salah Bouktif, and Piers Campbell. Soft Skills and Software Development: A Reflection from the Software Industry. *CoRR*, abs/1507.06873, 2015.
- [30] Kristófer Ívar Knutsen. Visual Scripting in Game Development. *Metropolia*, 2021.

- [31] Alejandro Martín Ortega. Implementation of Visual Scripting Module for Unity Projects: Enhancing Learning and Development. 2025.
- [32] Unity. Secuencias de Comandos Visuales de Unity — Unity. <https://unity.com/es/features/unity-visual-scripting#overview>. Último acceso 28 de Junio de 2025.
- [33] Anchit Shrivastava, Isha Jaggi, Nandita Katoch, Deepali Gupta, and Sheifali Gupta. A Systematic Review on Extreme Programming. In *Journal of Physics: Conference Series*, volume 1969, page 012046. IOP Publishing, 2021.
- [34] John Erickson, Kalle Lyytinen, and Keng Siau. Agile modeling, Agile Software Development, and Extreme Programming: The State of Research. *Journal of Database Management (JDM)*, 16(4):88–100, 2005.
- [35] Rostislav Fojtik. Extreme Programming in Development of Specific Software. *Procedia Computer Science*, 3:1464–1468, 2011.
- [36] José Joskowicz. Reglas y Prácticas en Extreme Programming. *Universidad de Vigo*, 22, 2008.

A. Carta Gantt

A continuación, se presenta la Carta Gantt del proyecto, el cual se divide en tres secciones: Desarrollo del Proyecto (Scripts y Ejercicios), Pruebas de las Experiencias y Análisis de Resultados. Todo esto se llevó a cabo en el periodo desde el primero de septiembre hasta el 21 de Noviembre.

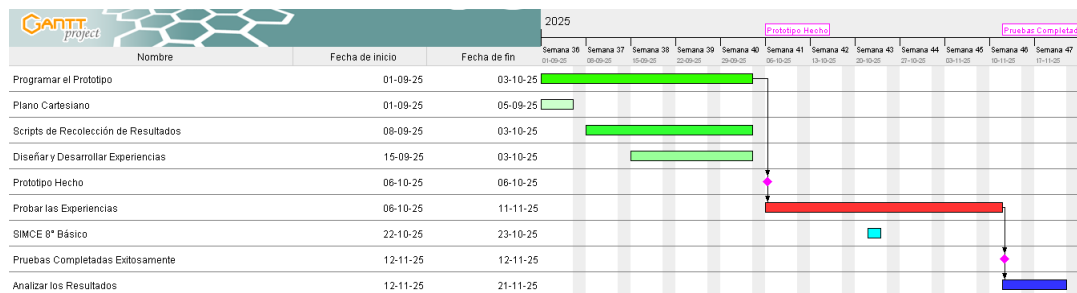


Figura A.1: Carta Gantt del proyecto

B. Gráficos de Resultados de Encuesta de Satisfacción

¿Qué tan probable es que recomiendes el taller "Videojuegos para Tod@s en Escuelas de Chile: Experiencias de Visual Scripting en Unity para el Módulo de Matemáticas" a tus amigos o familiares?

Respondidas: 8 Omitidas: 0

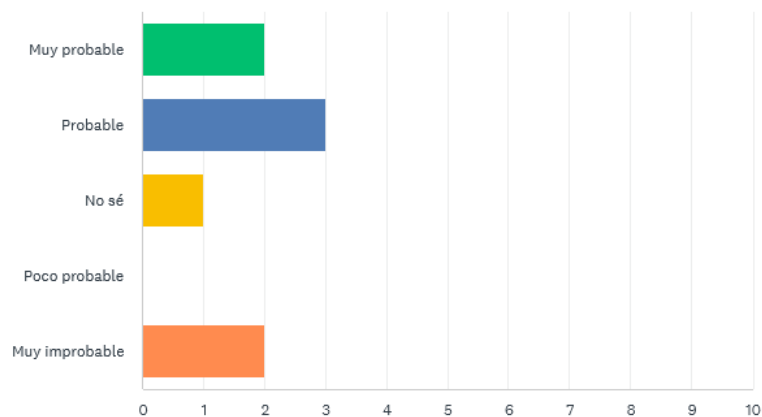


Figura B.1: ¿Qué tan probable es que recomiendes el taller “Videojuegos para Tod@s en Escuelas de Chile: Experiencias de Visual Scripting en Unity para el Módulo de Matemática” a tus amigos o familiares?

¿En general, cómo describirías el taller?

Respondidas: 8 Omitidas: 0

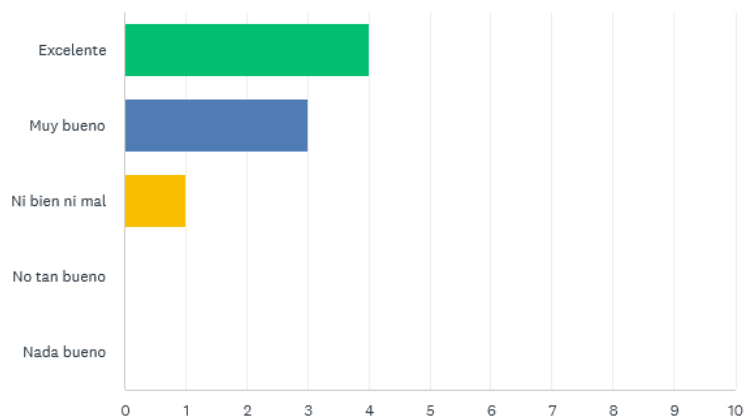


Figura B.2: ¿En general, cómo describirías el taller?

¿Qué tan claros fueron los objetivos de programación del taller?

Respondidas: 8 Omitidas: 0

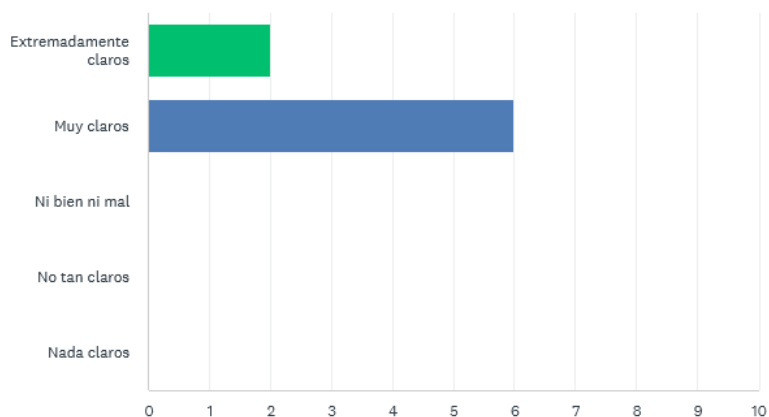


Figura B.3: ¿Qué tan claros fueron los objetivos de programación del taller?

¿Qué tan claros fueron los objetivos de matemáticas del taller?

Respondidas: 8 Omitidas: 0

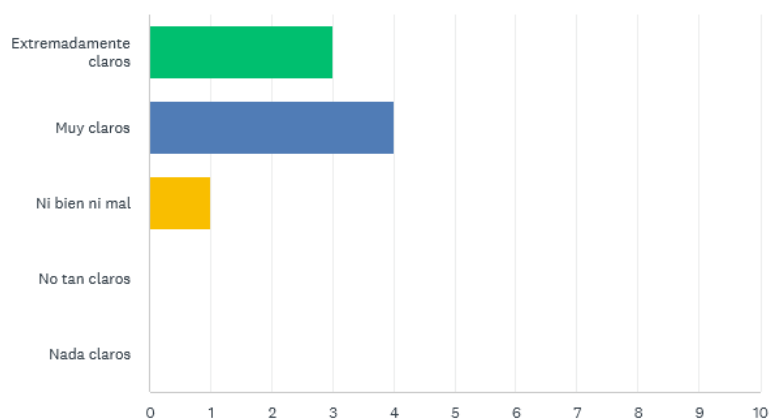


Figura B.4: ¿Qué tan claros fueron los objetivos de matemáticas del taller?

¿Cómo calificarías en general al instructor del taller?

Respondidas: 8 Omitidas: 0

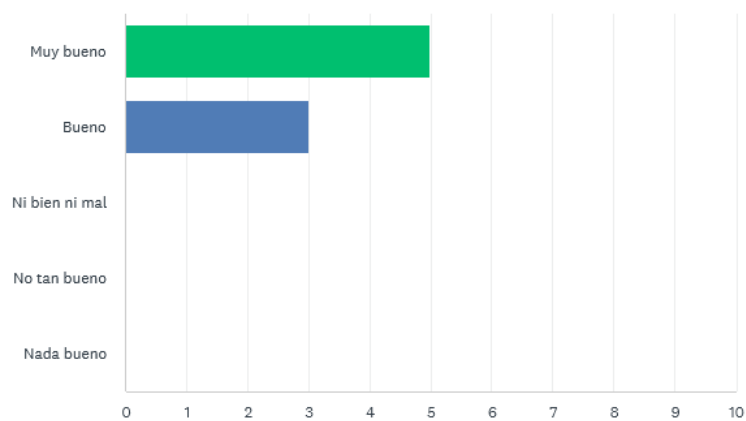


Figura B.5: ¿Cómo calificarías en general al instructor del taller?

¿Qué tan adecuadas te parecieron las actividades que se realizaron durante el taller?

Respondidas: 8 Omitidas: 0

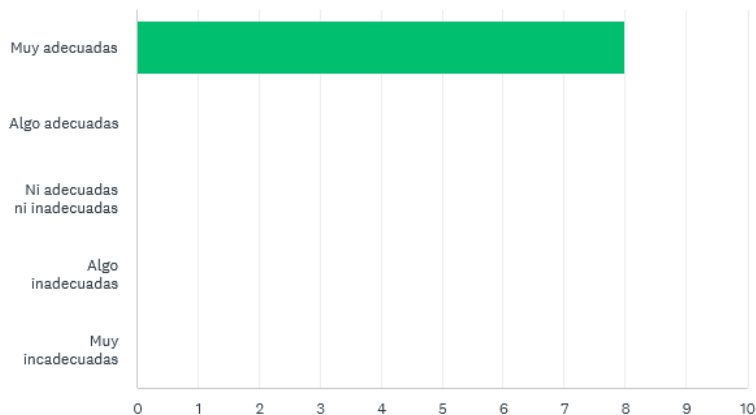


Figura B.6: ¿Qué tan adecuadas te parecieron las actividades que se realizaron durante el taller?

¿Te inscribirías a este taller si se volviese a hacer otra vez?

Respondidas: 8 Omitidas: 0

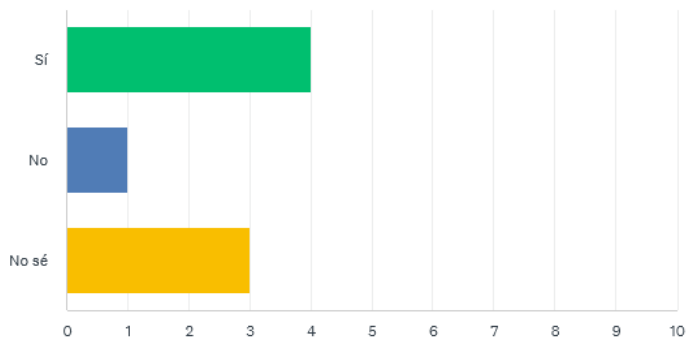


Figura B.7: ¿Te inscribirías a este taller si se volviese a hacer otra vez?