

PEQUEÑOS NAVEGANTES TIC

YEISON SMITH ROJAS ALDANA

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD)  
ESCUELA DE CIENCIAS BASICAS TECNOLOGICAS E INGENIERIA  
INGENIERIA DE SISTEMAS  
NEIVA (HUILA)  
2018

PEQUEÑOS NAVEGANTES TIC

YEISON SMITH ROJAS ALDANA

ASESOR  
JAIME RUBIANO LLORENTE  
INGENIERO DE SISTEMAS

PROYECTO DE GRADO PARA OPTAR EL TÍTULO:  
INGENIERO DE SISTEMAS

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD)  
ESCUELA DE CIENCIAS BASICAS TECNOLOGICAS E INGENIERIA  
INGENIERIA DE SISTEMAS  
NEIVA (HUILA)  
2018

Nota de Aceptación

---

---

---

---

Presidente del Jurado

---

Jurado

---

Jurado

Neiva -10, 09, 2018

## **AGRADECIMIENTOS**

Quiero agradecer primeramente a Dios porque me dio la oportunidad de vivir y de realizar una carrera profesional, además de estar presente en las dificultades. También quiero agradecer a mis padres por brindarme su cariño y apoyo para seguir adelante.

Un agradecimiento al ingeniero Jaime Rubiano y a la red de tutores de la universidad por sus asesorías y consejos, dedicando el tiempo requerido para el desarrollo del proyecto.

## CONTENIDO

	Pág.
1. INTRODUCCIÓN.....	14
2. OBJETIVOS .....	16
2.1 OBJETIVO GENERAL.....	16
2.2 OBJETIVOS ESPECÍFICOS .....	16
3 PLANTEAMIENTO DEL PROBLEMA.....	17
3.1 DEFINICIÓN DEL PROBLEMA .....	17
3.2 JUSTIFICACIÓN.....	17
4. MARCO TEÓRICO .....	19
5. MATERIALES Y MÉTODOS .....	21
5.1 MATERIALES.....	21
5.2 METODOLOGÍA .....	21
6. DESARROLLO DEL PROYECTO.....	23
6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO.....	52
6.2 CRONOGRAMA .....	55
7. CONCLUSIONES.....	56
8. RECOMENDACIONES.....	57
9. BIBLIOGRAFÍA.....	58
10. ANEXOS .....	61

## LISTA DE TABLAS

	Pág.
<b>Tabla1. Materiales .....</b>	<b>21</b>
<b>Tabla 2. Cronograma .....</b>	<b>55</b>

## LISTA DE GRÁFICAS

	Pág.
<b>Grafica 1. Frecuencia de uso de dispositivos tecnológicos .....</b>	<b>52</b>
<b>Grafica 2. Facilidad de uso en el colegio .....</b>	<b>52</b>
<b>Grafica 3. Rede sociales que se utiliza con más frecuencia .....</b>	<b>53</b>
<b>Grafica 4. Interés en los juegos en la Internet.....</b>	<b>53</b>
<b>Grafica 5. Interés de crear o diseñar juegos.....</b>	<b>53</b>
<b>Grafica 6. Tipos de juegos que le gustaría diseñar .....</b>	<b>54</b>

## LISTA DE FIGURAS

	Pág.
<b>Figura 1. tomada de scratch (programación) .....</b>	<b>37</b>
<b>Figura 2 tomada de Kodu (programación) .....</b>	<b>39</b>
<b>Figura 3 tomada de Code .....</b>	<b>40</b>
<b>Figura 4 tomada de CodeCombat .....</b>	<b>41</b>
<b>Figura 5 tomada GameMaker .....</b>	<b>42</b>
<b>Figura 6 tomada de Construct 2 .....</b>	<b>44</b>
<b>Figura 7 tomada de GameDevelop .....</b>	<b>45</b>
<b>Figura 8 tomada de Alice .....</b>	<b>47</b>
<b>Figura 9 tomada de CODINGAME .....</b>	<b>48</b>
<b>Figura 10 tomada de Code Hunt .....</b>	<b>49</b>
<b>Figura 11 tomada de Code Monkey .....</b>	<b>50</b>

## LISTA DE ANEXOS

	Pág.
<b>Anexo 1 Estrategia.....</b>	<b>61</b>
<b>Anexo 2. Formato de datos .....</b>	<b>65</b>
<b>Anexo 3. Formato de encuesta para recolección de información .....</b>	<b>66</b>
<b>Anexo 4. Formato resultados socialización del proyecto .....</b>	<b>67</b>
<b>Anexo 5. Manual de creación.....</b>	<b>68</b>
<b>Anexo 6. Manual de usuario.....</b>	<b>81</b>
<b>Anexo 7. Resultados socialización del proyecto .....</b>	<b>86</b>
<b>Anexo 8. Evidencia .....</b>	<b>89</b>

## GLOSARIO

**BLOG:** Página web, generalmente de carácter personal, con una estructura cronológica que se actualiza regularmente y que se suele dedicar a tratar un tema concreto.

**CONSTRUCTIVISTA** (Pedagógica): Que postula la necesidad de entregar al estudiante las herramientas necesarias (generar andamiajes) que le permitan construir sus propios procedimientos para resolver una situación problemática, lo que implica que sus ideas puedan verse modificadas y siga aprendiendo.

**E-LEARNING:** Procesos de enseñanza-aprendizaje que se llevan a cabo a través de Internet, caracterizados por una separación física entre profesorado y estudiantes, pero con el predominio de una comunicación tanto síncrona como asíncrona, a través de la cual se lleva a cabo una interacción didáctica continuada. Además, el alumno pasa a ser el centro de la formación, al tener que auto gestionar su aprendizaje, con ayuda de tutores y compañeros.

**MULTILENGUAJE:** Referido a una aplicación informática, se dice que esta es multilinguaje cuando su Interfaz de usuario puede ser mostrada a elección del usuario en cualquiera de diferentes idiomas

**NAVEGANTES:** utilizado normalmente para describir a los usuarios habituales de Internet o red.

**TIC:** Las Tecnologías de la Información y la Comunicación. Son el conjunto de tecnologías desarrolladas para gestionar información y enviarla de un lugar a otro. Abarcan un abanico de soluciones muy amplio. Incluyen las tecnologías para almacenar información y recuperarla después, enviar y recibir información de un sitio a otro, o procesar información para poder calcular resultados y elaborar informes (Servicios TIC, 2006).

**WEB 2.0:** se refiere al fenómeno social surgido a partir del desarrollo de diversas aplicaciones en Internet. El término establece una distinción entre la primera época de la Web (donde el usuario era básicamente un sujeto pasivo que recibía la información o la publicaba, sin que existieran demasiadas posibilidades para que se

generara la interacción) y la revolución que supuso el auge de los blogs, las redes sociales y otras herramientas relacionadas.

**WIKIS:** Sistema de trabajo informático utilizado en los sitios web que permite a los usuarios modificar o crear su contenido de forma rápida y sencilla.

**FEEDBACK:** que se utiliza para nombrar al regreso de una parte de la salida de un sistema o circuito a su propia entrada.

## RESUMEN

“Pequeños navegantes TIC” es un proyecto que busca implementar una estrategia para el desarrollo de juegos didáctico - educativos, guiado en un modelo de desarrollo de software basado en metodologías ágiles que permitan facilitar el proceso de enseñanza aprendizaje desde el análisis hasta la implementación.

La implementación se realizará en estudiantes de formación primaria de los grados tercero a quinto inicialmente y los temas a trabajar son herramientas web 2.0 y sus complementos.

El proyecto está concebido en seis (6) fases. En donde se identifican las metodologías ágiles, las aplicaciones web 2.0, se identifican conocimientos básicos de manejo tic y se desarrolla una estrategia para ser aplicada en el diseño de juegos.

Se realiza la implementación de las estrategias de enseñanza aprendizaje en instituciones educativas de la ciudad de Neiva. Finalizando con la Fase de evaluación del impacto de logros del proyecto. Por último, se presentan las conclusiones a las que se llegó al final del presente proyecto.

¿Con la implementación de una estrategia para el diseño de juegos basada en metodologías ágiles, se puede fortalecer el proceso de diseño de juegos en estudiantes de 3 a 5 de primaria?

## ABSTRACT

"Small ICT navigators" is a project that seeks to implement a strategy for the development of didactic-educational games, guided by a software development model based on agile methodologies that facilitate the learning teaching process from analysis to implementation.

The implementation will be carried out in primary school students from third to fifth grades initially and the topics to work on are web 2.0 tools and their complements.

The project is conceived in six (6) phases. Where the agile methodologies are identified, the web 2.0 applications, basic knowledge of tic management is identified and a strategy is developed to be applied in the design of games.

The implementation of teaching and learning strategies is carried out in educational institutions in the city of Neiva. Finishing with the Phase of evaluation of the impact of achievements of the project. Finally, the conclusions reached at the end of this project are presented.

¿Can the implementation of a strategy for the design of games based on agile methodologies strengthen the process of game design in students from 3 to 5 years of primary school?

## 1. INTRODUCCIÓN

El presente proyecto de grado trata sobre la implementación de una estrategia para el desarrollo de juegos didácticos – educativos guiado en un modelo de desarrollo de software basado en metodologías ágiles que permitan facilitar el proceso de enseñanza-aprendizaje desde el análisis hasta la implementación. El objetivo de la implementación es contribuir en el aprendizaje de los niños en sus primeras instancias y motivarlos a través de la WEB 2.0 en desarrollo de nuevas tecnologías. De esta manera, el estudiante obtendrá conocimientos y bases con un nivel óptimo de educación para su vida profesional.

Partiendo de lo anterior, el principal planteamiento que se pretende demostrar a lo largo del proyecto es afirmar que a través de recursos y complementos de la web 2.0 los estudiantes logran adquirir nuevos conocimientos aplicado al emprendimiento y al desarrollo de procesos cognitivos para su formación. Todo ello se justifica principalmente en la pedagogía utilizada actualmente en las Instituciones Educativas, las cuales no son suficientemente orientadas a nuevas herramientas tecnológicas que busquen contribuir al crecimiento formativo de los estudiantes. Para profundizar la investigación se realiza con una serie de encuestas diseñadas para obtener la información referente a 4 temas de importancia para el proyecto (conocimiento, evaluación, identificación y valoración); cumpliendo con las fases del proyecto.

La primera fase, es identificar y analizar metodologías ágiles para el desarrollo de software que se ajusten al diseño de juegos didácticos - educativos. Con la correspondiente selección de una metodología para su aplicación.

La siguiente fase, identificará las aplicaciones o herramientas web 2.0 y complementos que se ajusten al proceso de enseñanza aprendizaje planteado para el proyecto.

En la fase tres, identificar el conocimiento básico del estudiante sobre el uso de herramientas web 2.0, frecuencia de acceso a dispositivos tecnológicos y su gusto por los juegos por internet y el tipo de juego que le gustaría diseñar

Continuando con la Fase cuatro, se desarrolla la metodología seleccionada e implementación del proceso, aplicado al diseño de material de apoyo enfocado en las herramientas web 2.0 y complementos en la creación de juegos.

Pasando a la Fase quinta, la implementación de las estrategias de enseñanza aprendizaje en instituciones educativas de la ciudad de Neiva y Algeciras.

Finalizando con la Fase de evaluación del impacto de logros del proyecto. Por último, se presentan las conclusiones a las que se llegó al final del presente proyecto.

## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Implementar una estrategia para el desarrollo de juegos didácticos - educativos, guiado por un modelo de desarrollo de software basado en metodologías ágiles que permitan facilitar el proceso de enseñanza aprendizaje desde el análisis hasta la implementación de aplicaciones.

### **2.2 OBJETIVOS ESPECÍFICOS**

- Identificar y analizar metodologías ágiles para el desarrollo de software que se ajusten al diseño de juegos didácticos – educativos y seleccionar una que se ajuste a la propuesta.
- Identificar las aplicaciones o herramientas web 2.0 y complementos que se ajusten al proceso de enseñanza-aprendizaje planteado para el proyecto.
- Identificar el conocimiento básico del estudiante sobre el uso de herramientas web 2.0, frecuencia de acceso a dispositivos tecnológicos y su gusto por los juegos por internet y el tipo de juego que le gustaría diseñar.
- Desarrollo de una estrategia para el diseño de juegos, basada en la metodología seleccionada, y diseño de material de apoyo enfocado en las herramientas web 2.0 y complementos en la creación de juegos.
- Implementación de la estrategia diseñada en dos instituciones educativas de la ciudad de Neiva y Algeciras Huila.

### **3 PLANTEAMIENTO DEL PROBLEMA**

#### **3.1 DEFINICIÓN DEL PROBLEMA**

El uso de las TIC en la vida moderna y con mucho énfasis en la educación, están dando apoyo a procesos de enseñanza aprendizaje, que permiten generar algunos procesos de socialización del conocimiento, en donde los estudiantes desarrollan competencias comunicativas y afectivas. así mismo la presentación de informes, blog, wikis, desconociendo algunas herramientas complementarias que generan en el estudiante procesos cognitivos y constructivistas básicos para la formación de los estudiantes de hoy que estén en capacidad de resolver problemas del futuro.

Las instituciones educativas cuentan en su gran mayoría con recursos tecnológicos y existen dispositivos móviles en donde los estudiantes pueden manejar este tipo de recursos de web 2.0 y sus complementos, en el aula y en su vida cotidiana se maneja a diario una variedad de estos recursos, pero se está perdiendo la oportunidad que mediante el uso herramientas especializadas, se pueda orientar la forma de generar nuevo conocimiento aplicado al emprendimiento.

Los estudiantes y profesores no aplican procesos ordenados para el desarrollo de herramientas o aplicativos que fomenten el emprendimiento, se tiene un vacío en los procesos de análisis, diseño, desarrollo, implementación y evaluación de un proyecto computacional aplicado.

#### **3.2 JUSTIFICACIÓN**

Actualmente la pedagogía que se presenta en las instituciones educativas no es lo suficientemente orientada hacia las nuevas herramientas que la web 2.0 ofrece para el aprendizaje y crecimiento formativo de los estudiantes y proyecto de vida. Se debe trabajar en herramientas complementarias que desarrollen y fortalezcan la capacidad de análisis y la lógica para plantear y dar soluciones prácticas a problemas del futuro cercano. Todo ello guiado de procesos de análisis, diseño, desarrollo, implementación y evaluación.

El proyecto de “Pequeños Navegantes TIC” pretende elevar el nivel de pedagogía en las instituciones educativas de Neiva involucradas en el proyecto, posicionando la ciudad como principal semillero de futuros desarrolladores informáticos con

condiciones de análisis lógico para el desarrollo aplicaciones didácticos – educativas.

Desarrolla el potenciamiento en manejo de tic y de la programación lógica, para el desarrollo de aplicativos computacionales, fomenta el gusto por la ingeniería de sistemas en estudiantes desde los niveles básicos de formación.

#### 4. MARCO TEÓRICO

La Educación es uno de los campos en los que la Web 2.0 tiene una gran participación y está influyendo e influirá todavía más en todos los aspectos de la formación de los nuevos nativos digitales.

El término Web 2.0 (desde el 2004) está asociado con un fenómeno de evolución social de la comunicación, su función principal es la interacción que se logra a partir de diferentes aplicaciones en la web existentes que permiten compartir, e interactuar con información para el usuario con sitios web, a diferencia de la web no interactiva en donde los usuarios solo visualizaban la información.

Las herramientas complementarias para el desarrollo de aplicaciones didácticas, juegos, domótica entre otras, están al alcance de los estudiantes para que fortalezcan sus procesos lógicos hoy para que solucionen problemas de futuro.

“El análisis de las producciones orales y escritas de los estudiantes de los grados 5 y 7 durante el desarrollo de situaciones didácticas mediadas por el uso de aplicaciones web 2.0 permitió identificar una diversidad importante en los tipos de argumentos producidos por los estudiantes de ambos grados, En el contexto de esta investigación se observa que los Estudiantes De Básica Primaria y Secundaria tienen la suficiente capacidad cognitiva y lingüística para elaborar distintos tipos de argumentos en el marco de situaciones de aprendizaje mediadas por aplicaciones de la web 2.0.” Bolívar Buritica Wilson (2017).

“Hemos podido observar, mediante el análisis descriptivo, que la mayoría de los maestros utilizan el ordenador diariamente para cuestiones personales, tanto los de primaria como los de Educación Especial. En cuanto a la utilización de las herramientas de la Web 2.0, los maestros de Educación Especial parecen usarlos más a menudo que otros maestros, sin embargo, aún desconocen o no utilizan una gran variedad de estas herramientas, recurriendo principalmente a software educativo específico en las actividades con los alumnos con NEE. Los maestros de enseñanza primaria usan mayoritariamente YouTube y los Blogs como recursos en el aula. De acuerdo con la literatura, estas son realmente las herramientas más frecuentes (Coutinho & Alves, 2010; Cruz, 2008; Grosseck, 2009), posiblemente porque son fáciles de utilizar con los alumnos, o bien, porque cuentan con muchos recursos previamente creados.” (García, Carla; Días, Paulo; Sorte, Antonio; Díaz Pérez, Julian; Rita Leal, Ana; Gandra, Mario (2014))

La propuesta metodológica para el desarrollo de software educativo describe un conjunto de fases que permiten guiar el proceso de desarrollo de productos software que apoyen el proceso enseñanza aprendizaje. Esta propuesta tiene la particularidad que permite articular aspectos educativos, tecnológicos, conceptuales, metodológicos y organizacionales de una manera rápida sin caer en principios de las metodologías tradicionales como es la orientación al proceso. Letelier, P. (2006).

Las metodologías ágiles son muy útiles para el rápido desarrollo de software, empleándolo para proyectos pequeños, ya que por disponibilidad de tiempo y flexibilidad es muy útil. Las características de los proyectos para los cuales las metodologías ágiles han sido especialmente pensadas se ajustan a un amplio rango de proyectos industriales de desarrollo de software; aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y/o basados en nuevas tecnologías como: SCRUM, Crystal Methodologies, Dynamic Systems. Development.Method (DSDM), Adaptive Software Development (ASD), Feature-Driven Development (FDD). Lean Development (LD) Metodologías ágiles para el desarrollo de software (2017).

El trabajo con Scratch ofrecerá a los estudiantes con NEE oportunidades de construir activamente sus conocimientos, planificar proyectos, plantear dudas y preguntas y trabajar en la resolución de problemas, todo ello les permitirá un aprendizaje activo y significativo. Otro concepto importante es que al trabajar con Scratch, los alumnos con NEE se igualarán a otros en el error, en el hecho de que todos nos equivocamos, además aprenderán que los errores tienen sentido, sirven para aprender y se pueden corregir. López-Escribano, C., & Sánchez-Montoya, R. (2012).

## 5. MATERIALES Y MÉTODOS

### 5.1 MATERIALES

RECURSO	DESCRIPCIÓN	PRESUPUESTO
Equipo Humano	Aspirantes al título académico, diseñador del proyecto	0
Equipos y Software	Computador, herramientas web	0
Viajes y Salidas de Campo	Visitas a colegios	0
<b>TOTAL : 0</b>		

*Tabla 1. Materiales*

### 5.2 METODOLOGÍA

El método de investigación empleado es descriptivo, tipo encuesta. Para la recolección de datos se realiza una encuesta diseñada para obtener la información referente al proyecto. Se trabajará 5 temas de importancia.

- El primer tema está enfocado en identificar y analizar metodologías ágiles para el desarrollo y diseño de juegos didácticos – educativos.
- El segundo tema identifica las herramientas web 2.0 enfocadas a la educación, producción de conocimiento e interacción social.
- El tercer tema está destinado identificar el conocimiento básico del estudiante sobre el uso de herramientas web 2.0, frecuencia de acceso a dispositivos tecnológicos y su gusto por los juegos por internet y el tipo de juego que le gustaría diseñar
- El cuarto tema está direccionado a desarrollar una estrategia basada en la metodología seleccionada para el diseño de un material didáctico-educativo.
- El quinto tema está enfocado en aplicar y evaluar el impacto de la implementación de las estrategias de enseñanza-aprendizaje.
- Con el análisis de la información obtenida por este instrumento se tendrá las bases necesarias para construir una estrategia para el diseño juegos y material educativo de apoyo para facilitar el proceso de adquisición del conocimiento por los estudiantes.

La población objetivo son los estudiantes de primaria del grado 3 a 5 de la institución educativa Angel Maria Paredes de la ciudad de Neiva y de la institución educativa Juan XXIII de Algeciras con un total 50 estudiantes.

## 6. DESARROLLO DEL PROYECTO

Pequeños navegantes, busca implementar una estrategia de aprendizaje/enseñanza, con el fin de que los niños aprendan a desarrollar su lógica y creatividad, para que su nivel de educación sea óptimo. Utilizando las herramientas web 2.0 e implantación de metodologías ágiles.

Las herramientas de la Web 2.0, contribuye al desarrollo cognitivo de toda la comunidad, a partir de las opiniones e ideas individuales que cada miembro aporta a la misma. Y así es como tiene lugar la construcción social del conocimiento, es decir, el enriquecimiento de la comunidad se sustenta en las valiosas aportaciones de las individualidades que la conforman, originado un intercambio multidireccional del conocimiento. Moral Pérez, M. E. D., & Villalustre Martínez, L. (2007).

Las oportunidades derivadas de la Web 2.0 aplicadas al proceso de enseñanza-aprendizaje en la red da lugar a una reformulación del paradigma del e-Learning, el cual pasa a denominarse e-Learning 2.0, que viene definido por nuevas características:

- El aprendizaje se entiende como un proceso de carácter informal, marcado por la dinámica de crecimiento y desarrollo de las redes, donde los contenidos provienen de múltiples fuentes, son facilitados por los usuarios y distribuidos en diversos formatos o medios.
- La clave para la construcción del conocimiento se centra en el aprovechamiento de las experiencias y sinergias ajenas, y dota de un nuevo significado al aprendizaje inherente a la exploración de patrones de información.
- Aporta una nueva dimensión a la colaboración, más allá de la mera constitución de grupos de aprendizaje, priorizando las conexiones e interacciones fluidas entre redes abiertas.
- La comunicación y, por tanto, las herramientas que la facilitan dentro de la Web 2.0 (wikis, blogs, e-portfolios, etc...) son imprescindibles para propiciar el aprendizaje.
- El aprendizaje tiene lugar en entornos en permanente cambio, actualizándose constantemente. Pérez, M. E. D. M., Cernea, D. A., & Martínez, L. V. (2010).

Países como Francia comenzarán a impartir, ya en este próximo curso académico, programación en la educación primaria al igual que también lo hará Reino Unido

(tanto en educación primaria como en educación secundaria). Precisamente, en Reino Unido se está viviendo una "revolución tecnológica" en las escuelas y entidades privadas están impulsando *code camps* o subvencionando la compra de placas Arduino y Raspberry Pi para que los docentes no se vean frenados por la falta de recursos y puedan formar a los escolares en ciencias de la computación.

Estonia, desde el año pasado, enseña programación a los alumnos de primaria, Alemania ya lo hace en 3 regiones, Australia lo hará próximamente y países como Finlandia, Israel, Corea del Sur, Nueva Zelanda o Grecia llevan tiempo trabajando en programas piloto.

En Estados Unidos, figuras clave de la industria tecnológica como Bill Gates, Mark Zuckerberg o Jack Dorsey apoyan el proyecto Code.org que busca, precisamente, concienciar a alumnos y profesores en las ventajas de la enseñanza de la programación en las escuelas.

## METODOLOGÍAS AGILES

Las metodologías ágiles son muy eficiente al rápido desarrollo de software y responde a los cambios que pueden surgir a lo largo de un proyecto. A continuación, se ara mención de las metodologías identificadas.

### **Adaptive Software Development**

La metodología Desarrollo adaptativo de software (ASD) fue desarrollada por Jim Highsmith y Sam Bayer a comienzos de 1990. Esta metodología se adapta al cambio en lugar de luchar contra él. Se basa en la adaptación continua a circunstancias cambiantes. En ella no hay un ciclo de planificación-diseño-construcción del software, sino un ciclo especular colaborar-aprender.

Sus principales características del son:

- Iterativo.
- Orientado a los componentes de software (la funcionalidad que el producto va a tener, características, etc.) más que a las tareas en las que se va a alcanzar dicho objetivo.
- Tolerante a los cambios.
- Guiado por los riesgos.
- La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

### Ventajas

- Gracias a la colaboración y el aprendizaje, se mejoran mucho los procesos y se solucionan errores.
- Muy tolerante al cambio.
- Gestión del riesgo.

### Desventajas

Errores y cambios que no son detectados con anterioridad afectan la calidad del producto y su costo total.

## Agile Unified Process

Proceso Unificado Ágil (AUP) es una versión simplificada de RUP (Rational Unified Process), el cual fue desarrollado por IBM. Describe una manera simple de entender el desarrollo de aplicaciones de negocio usando técnicas ágiles y conceptos heredados del RUP.

AUP se basa en las siguientes filosofías:

1. Los empleados saben lo que están haciendo. La gente no va a leer documentación del proceso detallada, pero quieren algo de orientación a alto nivel y/o formación de vez en cuando. El producto AUP proporciona enlaces a muchos de los detalles, pero no fuerza a ellos.
2. Simplicidad. Todo está descrito de forma concisa.
3. Agilidad. AUP se ajusta a los valores y principios de desarrollo de software ágil y la Alianza Ágil.
4. Foco en las actividades de alto valor. El foco está en las actividades que realmente cuentan, no en todas las posibles cosas que pudieran pasar en un proyecto.
5. Independencia de herramientas. Se puede usar cualquier conjunto de herramientas. La recomendación es que se usen las herramientas que mejor se adapten al trabajo, que son con frecuencia herramientas simples.
6. Habrá que adaptar AUP para cumplir con las necesidades propias.

En los proyectos que usan AUP, normalmente se entregan versiones de desarrollo al final de cada iteración. Una versión de desarrollo de una aplicación es una versión que potencialmente puede ser lanzada en producción si pasa la garantía de calidad de pre-producción, supera la fase de pruebas y los procesos de despliegue.

El ciclo de vida de AUP consta de 4 fases: Inicio, Elaboración, Construcción y Transición. Además, existen 7 disciplinas que trabajan en cada fase del ciclo de vida.

## Ventajas

- El personal sabe lo que está haciendo: no obliga a conocer detalles (alto nivel)
- Simplicidad: apuntes concisos.
- Agilidad: procesos simplificados del RUP.
- Centrarse en actividades de alto valor: esenciales para El desarrollo.
- No fuerza a usar ninguna técnica o herramienta.

## Desventajas

- El AUP es un producto muy pesado (simplificación de RUP), en relación a otros ágiles.
- Como es un proceso simplificado, muchos desarrolladores eligen trabajar con RUP, por tener a disposición más detalles en el proceso.

## Crystal

Concebido por Alistair Cockburn (uno de los firmantes del manifiesto ágil), este modelo no describe una metodología cerrada, sino un conjunto de ellas, junto con los criterios para seleccionar y adecuar la más apropiada al proyecto.

El nombre de metodologías Crystal viene de que cada proyecto software puede caracterizarse según dos dimensiones, tamaño o dimensión (número de personas en el proyecto) y criticidad (consecuencia de los errores), al igual que los minerales se caracterizan por dos dimensiones, color y dureza. Y esta es una de las bases de las metodologías Crystal.

7 propiedades fundamentales de las metodologías Crystal, que son:

- Entregas frecuentes, en base a un ciclo de vida iterativo e incremental. En función del proyecto puede haber desde entregas semanales hasta trimestrales.
- Mejora reflexiva. Que viene a ser mejora continua. Las iteraciones ayudan a ir ajustando el proyecto, a ir mejorándolo.
- Comunicación osmótica. Que el equipo esté en una misma ubicación física, para lograr la comunicación cara a cara.

- Seguridad personal. Todo el mundo puede expresar su opinión sin miedos, teniéndosele en cuenta, considerándose su opinión, etc.
- Enfoque. Períodos de no interrupción al equipo (2 horas), objetivos y prioridades claros, definiendo así tareas concretas.
- Fácil acceso a usuarios expertos. Las Crystal (a diferencia de otras como XP) no exigen que los usuarios estén continuamente junto al equipo de proyecto (no todas las organizaciones pueden hacerlo), sí que, como mínimo, semanalmente debe haber reuniones y los usuarios deben estar accesibles.
- Entorno técnico con pruebas automatizadas, gestión de la configuración e integración continua.

### Ventajas

Se ajusta a la criticidad o tipo de proyecto y al tamaño del equipo.

### Desventajas

Coordinación difícil de equipos grandes.

## **Dynamic Systems Development Method**

método de desarrollo de sistemas dinámicos (DSDM) es la única de las metodologías aquí planteadas surgida de un Consorcio, formado originalmente por 17 miembros fundadores en enero de 1994. El objetivo del Consorcio era producir una metodología de dominio público que fuera independiente de las herramientas y que pudiera ser utilizado en proyectos de tipo RAD (Rapid Application Development).

La estructura del método fue guiada por estos nueve principios:

1. El involucramiento del usuario es imperativo: La participación del usuario es la clave principal para llevar a cabo un proyecto eficiente y efectivo, donde ambos, usuarios y desarrolladores compartan un sitio de trabajo, de tal forma que las decisiones se puedan hacer de la forma más exacta.
2. Los equipos de DSDM deben tener el poder de tomar decisiones: Se le deben otorgar poderes al equipo del proyecto para tomar decisiones que son importantes para el progreso del proyecto, sin tener que esperar a aprobaciones de más alto nivel.

3. El foco está puesto en la entrega frecuente de productos: Entregar algo suficientemente bueno pronto es siempre mejor que entregar todo perfecto al final. Entregando el producto con frecuencia desde una etapa temprana del proyecto, el producto se puede probar y revisar y el registro de pruebas y el documento de revisión se pueden tener en cuenta en la siguiente iteración o fase.
4. La conformidad con los propósitos del negocio es el criterio esencial para la aceptación de los entregables: El principal criterio de aceptación de un entregable es que entregue un sistema que trate las necesidades del negocio actuales. Entregar un sistema perfecto que trate todas las necesidades del negocio posibles es menos importante que centrarse en las funcionalidades críticas.
5. El desarrollo iterativo e incremental y conducido por la retroalimentación del usuario es necesario para converger hacia una correcta solución del negocio.
6. Todos los cambios durante el desarrollo son reversibles.
7. Los requerimientos están especificados a un alto nivel: Se debería hacer una línea base del alcance y los requisitos a alto nivel antes de que el proyecto empiece.
8. El testing es integrado a través del ciclo de vida: Las pruebas se llevan a cabo a lo largo de todo el ciclo de vida del proyecto.
9. Un enfoque colaborativo y cooperativo entre todos los interesados es esencial

DSDM consta de tres fases:

- Fase pre-proyecto: En esta fase nos aseguraremos que todo está configurado y preparado correctamente para que el proyecto comience y llegue a ser un éxito.
- Fase de ciclo de vida del proyecto: Está subdividida en 5 etapas: estudio de viabilidad, estudio de negocio, iteración de modelo funcional, iteración de diseño y construcción e implementación.

- Fase post-proyecto: La principal tarea de esta fase es el mantenimiento y corrección de errores.

### Ventajas

- Permite alcanzar nivel 2 CMMI.
- Uso de prototipos para obtener requisitos.
- Todos los cambios durante el desarrollo son reversibles
- Los requisitos están especificados a un alto nivel
- El testing está Integrado a través del ciclo de vida.
- Desarrollo rápido

### Desventajas

- Documentación abundante.
- Los equipos de DSDM deben tener el poder de tomar decisiones.

## **Extreme Programming**

Extreme Programming (XP) o metodología de la programación extrema fue ideada por Kent Beck, a partir de un conjunto de buenas prácticas y un modo de realizar el desarrollo de software, fundamentado en entregar constantemente, el mayor valor al cliente. XP tuvo su origen a finales de 1980. A partir de entonces, sus prácticas fueron perfeccionadas y sus ideas se centraron en el desarrollo de un software que fuera, a la vez, adaptativo y orientado a las personas. El furor de las metodologías ágiles, a finales de la década del 90, tuvo como una de sus mayores protagonistas a XP.

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1

El ciclo de vida ideal de XP consiste en 6 fases: exploración, planificación de la entrega, iteraciones, producción, mantenimiento y muerte del proyecto.

### Ventajas

- El cliente es parte del equipo.
- Mejores prácticas bien definidas.
- Simplicidad.
- Potencia la mejora continua.
- Da prioridad a la acción.
- Propiedad colectiva del código

### Desventaja

- Documentación pobre.
- Puede llevar a poca disciplina.
- La reflexión puede quedar demasiado relegada frente a la acción.
- Presencia obligatoria e importante del cliente.
- Hacer las cosas simples requiere experiencia.
- Requiere coraje para asumir ciertas prácticas.

## **Feature Driven Development**

Desarrollo basado en funcionalidades (FDD) ideada por Jeff De Luca y Peter Coad a mediados de los 90, combina el desarrollo dirigido por modelos con el desarrollo ágil. Se centra en el diseño de un modelo inicial, cuyo desarrollo será dividido en función a las características que debe cumplir el software e, iterativamente, se diseñará cada una de estas características. Por tanto, cada iteración consta de dos partes, diseño e implementación de cada característica. A diferencia de otras metodologías ágiles no cubre todo el ciclo de vida sino sólo las fases de diseño y construcción.

### Características:

- Se preocupa por la calidad, por lo que incluye un monitoreo constante del proyecto.
- Ayuda a contrarrestar situaciones como el exceso en el presupuesto, fallos en el programa o el hecho de entregar menos de lo deseado.

- Propone tener etapas de cierre cada dos semanas. Se obtienen resultados periódicos y tangibles.
- Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear.
- Define claramente entregas tangibles y formas de evaluación del progreso del proyecto.
- No hace énfasis en la obtención de los requisitos sino en cómo se realizan las fases de diseño y construcción.

### Ventajas

- Monitoreo constante del avance del proyecto.
- Énfasis en la calidad.
- Ofrece soluciones a situaciones de riesgo (cambios en requisitos, pérdidas de tiempo...)

### Desventajas

- Propiedad individual del código.
- No cubre todo el ciclo de vida. Necesita de otras metodologías que la complementen

## **Kanban**

La metodología de desarrollo Kanban (Kanban Software Development) está basada en la metodología de fabricación industrial del mismo nombre. Su objetivo es gestionar de manera general como se van completando tareas, pero en los últimos años se ha utilizado en la gestión de proyectos de desarrollo software.

En Kanban no existen unas fases definidas, sino que se habla de un flujo. No existen iteraciones de tiempo definido, sino que trabaja con cadencias, es decir, se liberan versiones o entregables en base a necesidades o eventos. Tampoco existen unas reuniones diarias, o de inicio de sprint (porque no existen). Lo extremista que es esta metodología en cuanto a lo adaptativa que es y lo poco que define procesos, roles e incluso la forma de hacer el tablero Kanban, hace que, en algunos círculos,

sobre todo los más cercanos a Scrum, consideren a Kanban una técnica de señalización más que una metodología.

### Ventajas

- Facilita el que todo el mundo sepa que hacer, quien hace qué y cómo va el proyecto.
- Limita el número de tareas a hacer al mismo tiempo. Control del flujo de trabajo.
- Fácil de aplicar.
- Flexible ante cambios.

### Desventajas

- Se deben definir las fases del ciclo de trabajo.
- No define roles, ni fases, ni tampoco profundiza en el tablero Kanban.
- Algunos lo ven más como una técnica que como metodología.

## **Scrum**

Scrum es un proceso ágil que se puede usar para gestionar y controlar desarrollos complejos de software y productos usando prácticas iterativas e incrementales, así define un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.) mediante la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar la predictibilidad y el riesgo a niveles aceptables.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar sus pensamientos sobre lo que quieren y necesitan, y de que los desafíos que no se pueden predecir no se pueden tratar fácilmente de una forma predictiva o planificada tradicional. Por esto, Scrum adopta un enfoque empírico, aceptando que el problema no se puede entender o definir completamente, centrándose en cambio en maximizar las habilidades del equipo para entregar rápidamente y responder a los requisitos emergentes.

Scrum es un esqueleto de proceso que incluye un conjunto de prácticas y roles predefinidos. Los roles principales en Scrum son el “ScrumMaster” que mantiene los procesos y trabaja junto con el jefe de proyecto, el “Product Owner” que representa a las personas implicadas en el negocio y el “Team” que incluye a los desarrolladores. Sáez Martínez, P. J. (2013).

Hay diferentes roles de scrum como: Product owner, Scrum master, Scrum team.

- **Product owner:** Este rol, representa la voz del cliente (a veces incluso puede ser el mismo cliente, si procede). El cliente debe estar dispuesto a llevar a cabo el proceso scrum. Este personaje, se asegura de que el equipo trabaja en la dirección de negocio adecuada. El product owner, escribe casos de uso, los prioriza, y los coloca en el product backlog. Debe tener un amplio conocimiento en el área del proyecto a desarrollar, y debe conocer el negocio subyacente.

#### **Características**

- Representante del cliente
  - Trabaja con el equipo
  - Él / Ella mantiene la cartera de pedidos del producto
  - Prioriza los requisitos del producto
  - Interfaz entre el equipo y la empresa
  - Definir las características del producto
  - Decidir sobre la fecha de lanzamiento del producto y el contenido
  - Ser responsable del beneficio
  - Priorizar y ajustar características, según sea necesario
  - Responsable de la aprobación final
  - Aceptar o rechazar resultados de trabajo
- 
- **ScrumMaster:** El ScrumMaster, hará que el proceso scrum sea posible. Básicamente, quitará los impedimentos que puedan surgir y que impidan al equipo llegar a su objetivo del sprint. El ScrumMaster no es el líder del equipo, ya que éste se auto gestiona, pero actúa de aislante entre el equipo, y las molestias externas que puedan distraer el desarrollo. En equipos inexpertos, el ScrumMáster es absolutamente necesario, y el equipo se apoyará en él durante todo el proceso. Cuando el equipo es experto en Scrum, el ScrumMáster se limitará exclusivamente a eliminar los posibles problemas que pueda tener el equipo y que se detecten en las reuniones diarias principalmente.

### Características

- ayuda a los miembros del equipo y es responsables de
  - eliminar obstáculos.
  - tiene un rol de liderazgo sobre los miembros del equipo
  - es responsable de gestionar el proceso
  - Él/Ella es responsable de ejecutar el scrum diario
  - Mantiene la documentación requerida
  - Protege al equipo de interferencias externas
- 
- **El equipo:** El equipo de trabajo, será el que tiene la responsabilidad de desarrollar el producto. Un equipo, puede componerse habitualmente por entre 5 y 10 personas, guiados por un ScrumMaster. El ScrumMaster, puede guiar a más de un equipo de trabajo. Estas personas, tendrás diferentes habilidades técnicas, que se ajustarán al objetivo de este grupo para este sprint. Los grupos, no tienen por qué ser siempre los mismos, ni del mismo tamaño entre sprints.

### Características

- Desarrolla producto
- Cada miembro del equipo es responsable de su trabajo.
- Ellos aportan lo mejor para cada sprint
- Incluye 5-10 personas en un equipo
- Dos tipos de equipos
- Hay dos tipos de equipos, como auto-
- organización y equipo interfuncional
- Responsable de fracaso o éxito
- Los miembros del equipo pueden ser programadores, probadores, diseñadores de experiencia, etc.

Por otro lado, tenemos:

- **Usuarios:** El software se construye con un fin, y ese fin, es satisfacer las necesidades de los usuarios. Si estos no pueden utilizar el software, o este no hace lo que ellos necesitan, entonces no vale para nada.
- **Stakeholders:** Estas personas, se encargarán de que el proyecto tenga rendimiento, y sea rentable. No tomarán parte directa en el proceso scrum, salvo en los sprint reviews, en los que darán su opinión sobre si el proyecto cumple sus objetivos.
- **Managers:** Serán los miembros implicados, que establezcan el entorno del producto en las organizaciones en las que se desarrollará. Sáez Martínez, P. J. (2013).

## Ventajas

- Equipo auto organizado.
- El equipo sabe lo que tiene que hacer cada día.
- El cliente sabe lo que se le entrega en cada sprint.
- Flexible a cambios.
- Desarrolladores tienen autonomía.
- Minimiza el trabajo de gestión.
- Minimiza el síndrome del estudiante

## Desventajas

- Puede producir stress al sentirse el equipo en un continuo sprint.
- Requiere un equipo formado, motivado y con cierta experiencia.
- Necesidad de involucración del cliente.
- Problemas en equipos distribuidos geográficamente.

## Herramientas web 2.0

### Scratch

Es un lenguaje de programación visual desarrollado por el MIT Media Lab. Scratch es utilizado por estudiantes, académicos, profesores y padres para crear fácilmente animaciones, juegos (también educativos) e interacciones etc. Para las escuelas se convierte en una oportunidad para ayudar a los estudiantes en el desarrollo de habilidades mentales mediante el aprendizaje de la programación sin necesidad de saber del programa. Sus características ligadas al pensamiento computacional han hecho que sea muy difundido actualmente en la educación de niños adolescentes y adultos.

### Las características principales de scratch son:

- La interfaz es muy intuitiva y muy gráfica
- Existencia de un banco de recursos en el propio programa: objetos, personajes, escenarios, sonidos...
- Las acciones de los personajes se forman apilando piezas a modo de puzzle
- Visualización instantánea de lo que hacemos en el programa



## **Kodu**

Es un entorno de desarrollo integrado de programación Se ejecuta en Xbox 360 y Microsoft Windows XP, Windows Vista, Windows 7, Windows 8 y Windows 10.

Fue lanzado en el mercado de Xbox Live el 30 de junio de 2009.

Kodu permite a los niños crear juegos en la PC y Xbox a través de un lenguaje de programación visual simple. Kodu se puede utilizar para enseñar creatividad, resolución de problemas, narración, así como la programación. Cualquier persona puede usar Kodu para hacer un juego, los niños pequeños, así como los adultos sin ningún diseño o habilidades de programación.

### **Características:**

- Interfaz de usuario accesible sin necesidad de ser experto en programación
- Programación basada en eventos
- Lenguaje de programación visual con 200 herramientas de programación
- Orientado al diseño de pequeños juegos
- Desarrollo rápido e intuitivo para seguir la secuencia de las instrucciones
- Soporte de controlador de Xbox, teclado y mouse
- Considera a la programación como un medio creativo
- Introduce la lógica y la resolución de problemas
- Introduce condiciones y secuencias, que enseña la causa y el efecto
- Ayuda a resolver situaciones paso a paso, e iterar sobre el proceso de diseño
- Permite la creación de mundos de tamaño y forma arbitraria, con ciertos elementos primitivos de la vida real.

### **Ventajas:**

- Evita escribir código al hacer que los usuarios construyan programas usando elementos visuales a través de un controlador de juego
- En lugar de una visualización de mapa de bits o 2D, los programas se ejecutan en un entorno de simulación 3D
- No requiere conexión a internet

### **Desventajas:**

- El programa al inicio es un poco engorroso porque no hay un tutorial

Kodu Game Lab también se ha utilizado como una herramienta de aprendizaje educativo en escuelas seleccionadas y centros de aprendizaje

Muchos tipos diferentes de juegos se pueden hacer en Kodu, tales como carreras, estrategia, aventura, plataforma, rompecabezas, tiradores de primera persona, y otros.

Entorno de trabajo



Figura 2 tomada de Kodu (programación)

## Code

Code.org® es una organización sin fines de lucro dedicada a ampliar el acceso a la informática ya aumentar la participación de las mujeres y niños insuficientemente representados. Su visión es que cada estudiante en cada escuela debe tener la oportunidad de aprender la informática, al igual que la biología, la química o el álgebra. Code.org organiza la campaña anual “Hour of Code”, que ha contratado al 10% de todos los estudiantes del mundo y ofrece el currículo principal para la informática K-12 en los distritos escolares más grandes de los Estados Unidos.

## Características

- Enfocado al aprendizaje progresivo.
- Múltiples opciones de actividades de aprendizaje.
- Plataforma WEB.
- Trabaja en cualquier navegador.
- Todas las opciones prediseñadas.
- Permite agregar más funciones por medio de código.

## Ventajas

- Interface sencilla
- Permite aprender la lógica de manera gráfica o con código
- Gran variedad de juegos y entornos para el aprendizaje
- Soporta cualquier plataforma
- No requiere la instalación de aplicaciones adicionales
- Ofrece entorno para estudiantes y maestros.

## Desventajas

- Requiere estar siempre conectado a internet

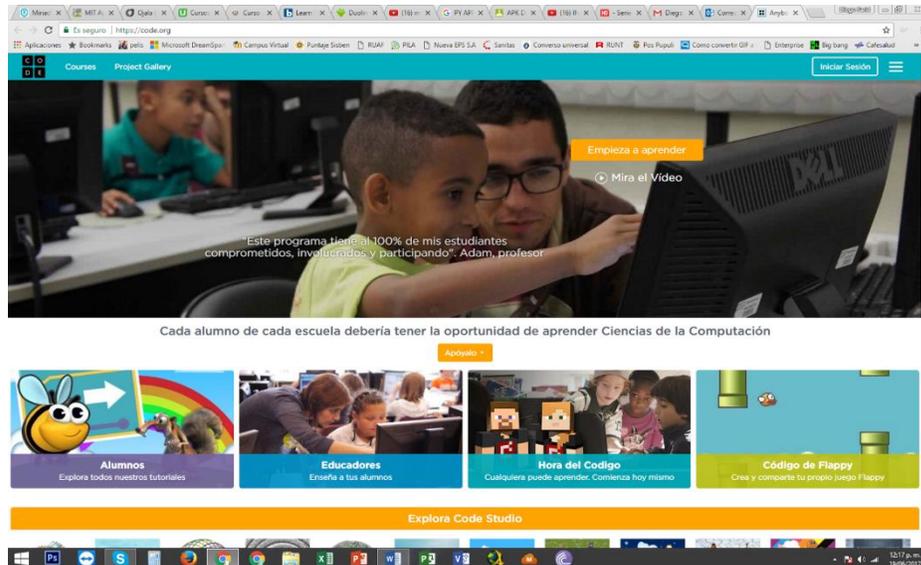


Figura 3 tomada de Code

## CodeCombat

CodeCombat es una plataforma para que los estudiantes aprendan informática mientras están jugando. CodeCombat fue creado para dar a los estudiantes la sensación de poder de la magia en la punta de sus dedos mediante el tipeo de código, mientras se juega, se aprende.

Como resultado, eso les permite también aprender más rápido. De una forma más rápida. Es como tener una conversación en vez de leer un manual. CodeCombat busca llevar esa conversación a todas las escuelas y a cada estudiante, porque todos deberían tener la oportunidad de aprender la magia de la programación.

### Características:

- Usa el lenguaje de JavaScript.
- Utiliza HTML5.
- Trabaja en cualquier navegador

## Ventajas

- El entorno es muy atractivo y llamativo para los niños

- Maneja una lógica sencilla
- Trabaja en cualquier entorno y sistema operativo
- No requiere instalar aplicaciones adicionales

## Desventajas

- Requiere siempre estar conectado a internet
- Requiere tener cuenta de usuario y se debe loguear para ingresar
- Para tener todas las características se debe pagar
- Cuando no se paga se necesita esperar un tiempo para continuar



Figura 4 tomada de CodeCombat

## Game Maker

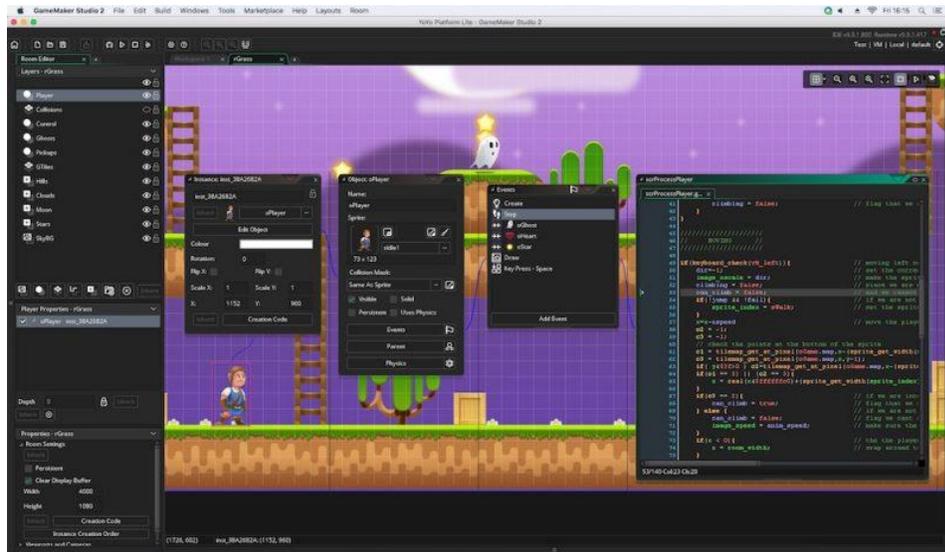


Figura 5 tomada GameMaker

Es una herramienta basada en un lenguaje de programación interpretado y un kit de desarrollo de software (SDK) para desarrollar videojuegos, creado por el profesor Mark Overmars en el lenguaje de programación Delphi, y orientado a usuarios novatos o con pocas nociones de programación. El programa es gratuito, aunque existe una versión comercial ampliada con características adicionales. Overmars liberó la primera versión pública el 15 de noviembre de 1999.

### Características:

El programa está diseñado para permitir a sus usuarios desarrollar fácilmente videojuegos sin tener que aprender un lenguaje de programación como C++ o Java. Para los usuarios experimentados Game Maker contiene un lenguaje de programación de scripts llamado Game Maker Language (GML), que permite a los usuarios personalizar aún más sus videojuegos y extender sus características. Los videojuegos pueden ser distribuidos bajo cualquier licencia sujeta a los términos del EULA de Game Maker, en archivos ejecutables no editables ".exe", paquetes Android ".apk", y conjuntos de script HTML5, como código fuente con extensión .gmx (GM: Studio y GM: HTML5), .gm81 (Versión 8.1), .gmk (versiones 7.x y 8.x), .gm6 (versión 6.x), .gmd (versiones 5.x y 4.x), y .gmf (versión 3 e inferiores). Los usuarios de Game Maker tienen permitido distribuir e incluso vender sus creaciones mientras cumplan con los términos del EULA de Game Maker, que prohíbe un número de programas ilegales como los que involucran uso no autorizado de material con derechos de autor o los que rompan la ley en general.

La interfaz principal para el desarrollo de videojuegos de Game Maker usa un sistema de "arrastrar y soltar", que permite a los usuarios que no están familiarizados con la programación tradicional crear videojuegos intuitivamente organizando íconos en la pantalla. Game Maker viene con un conjunto de bibliotecas de acciones estándar, que cubren cosas como movimiento, dibujo básico, y control simple de estructuras. Para extender la funcionalidad de arrastrar y soltar de Game Maker, los usuarios pueden construir bibliotecas de acciones personalizadas para agregar nuevas acciones a sus videojuegos. Estas pueden ser creadas (generalmente por usuarios más avanzados) usando la herramienta de generación especial de bibliotecas.

### **Ventajas:**

- Capacidad de usar sprites únicos e integrarlos en el entorno de la programación y a su vez declarar "posiciones" o "fases" del sprite.
- Programación amigable de eventos, se pueden hacer eventos repetitivos y únicos.
- Apertura a hacer todo tipo de juego, no tiene restricciones de no poder hacer juegos incluso online, o juegos estratégicos, todo está en cuestión de la programación.
- Capacidad de crear menús y otras características personalizadas al agregar programación por tecla.

### **Desventajas:**

- Su principal desventaja es su precio, aunque en un principio era una alternativa más accesible ante otros motores hoy en día su versión completa con todos los módulos de exportación no está al alcance de todos, cuenta con una versión gratuita, pero igualmente no posee los módulos de exportación y tiene otros limitantes.

## Construct 2



*Figura 6 tomada de Construct 2*

Es un software para Windows que nos permite desarrollar juegos de calidad en HTML5 de una forma fácil y rápida, sin necesidad de experiencia en programación. Con este software podemos crear juegos a través de una interfaz intuitiva, la cual nos ofrece muchas opciones y nos permite controlar los eventos, que es donde se va desarrollando el juego.

### **Características:**

Posee un motor de efectos físicos que nos permite agregar esos efectos a los juegos. Ofrece 3 licencias: Free, Standard y Business. La primera con limitaciones, la podemos descargar y usar por el tiempo que se nos ocurra, pero no podemos comercializar los juegos. La "Standard" ofrece más características y nos permite comercializar los juegos como desarrolladores particulares. Por último, la licencia de "Business" posee las mismas características de la versión "Standard" pero su licencia es para empresas dedicadas al desarrollo, además de permitir comercializar los juegos sin ningún tipo de restricciones.

El principal método de programación de juegos y aplicaciones en Construct es a través de 'hojas de eventos', que son similares a los archivos de origen utilizados en los lenguajes de programación. Cada hoja evento tiene una lista de eventos, que contienen las sentencias condicionales o desencadenantes. Una vez que estos se cumplan, acciones o funciones pueden ser llevadas a cabo. Los eventos lógicos tales como Or y And, así como sub-eventos (en representación de alcance) permitir sistemas sofisticados para ser programado sin tener que aprender un lenguaje de



Es un software gratuito de creación de juegos, que permite hacer juegos para la web (HTML5) o para Windows y Linux. No se requiere programación o codificación de habilidad. Es fácil de usar y lo suficientemente potente como para usuarios avanzados. Es una herramienta que permite crear juegos en 2D con una facilidad impresionante, sin necesidad de conocer ningún lenguaje de programación y lo mejor de todo, es que el juego que se crea estará perfectamente personalizado para cumplir los requerimientos del creador.

### **Características:**

- Game Develop nos permitirá crear juegos en 2D con una facilidad impresionante, sin necesidad de conocer ningún lenguaje de programación y lo mejor de todo, es que el juego que crearemos estará perfectamente personalizado para cumplir nuestros requerimientos.
- Game Develop cuenta con una interfaz de usuario sumamente intuitiva y muy sencilla de utilizar. Con esta aplicación podemos ir creando las ambientaciones y los personajes que integrarán nuestro juego de manera visual, sin tener que teclear ningún tipo de código.
- Una de las características más destacables de Game Develop es la capacidad de personalización del juego a crear, ya que podemos especificar cada uno de los eventos creados, incluir unos de Mouse o joystick, adicionar efectos de sonido y banda sonora, así como también incluir el tipo de dificultad y obstáculos que se nos antoje.
- Aclarar que Game Develop posee su propio compilador, capaz de exportar los juegos realizados a aplicaciones para Windows, Linux y Mac OS. ¡Y no es necesario que otras personas tengan el Game Develop para poder jugarlos! ya que los exporta de forma completa.

### **Ventajas:**

Entre sus ventajas podemos encontrar el hecho de que este traducido al español al igual que gran parte de su documentación, su comunidad es muy activa y bastante grande por lo que siempre se podrá encontrar ayuda con cualquier duda extra que se tenga, y sobre todo la facilidad con la que se puede aprender y crear algo de manera rápida. Cuenta con un motor de físicas y un sistema de partículas propio, los cuales son muy fáciles de utilizar y aplicar, está orientado a creaciones en 2D.

### **Desventajas:**

Sus desventajas, son la complejidad a la que puede llegar un proyecto grande dificulta un poco su manejo, y no recibe nuevas actualizaciones.

## Alice

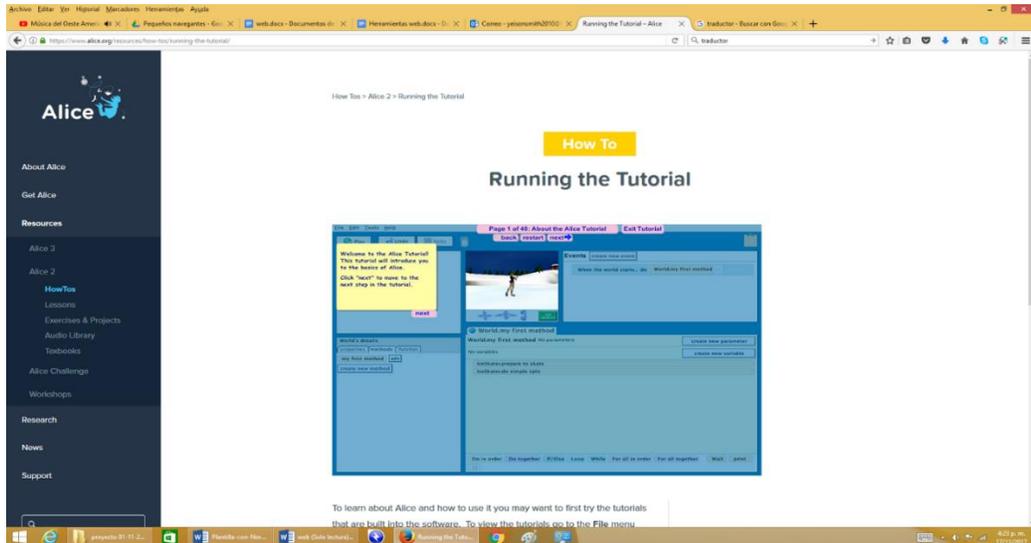


Figura 8 tomada de Alice

Alice es un innovador entorno de programación basado en bloques que facilita la creación de animaciones, la creación de narrativas interactivas o la programación de juegos simples en 3D. A diferencia de muchas de las aplicaciones de codificación basadas en rompecabezas, Alice motiva el aprendizaje a través de la exploración creativa. Alice está diseñada para enseñar habilidades de pensamiento lógico y computacional, principios fundamentales de programación y para ser una primera exposición a la programación orientada a objetos. El Proyecto Alice proporciona herramientas y materiales complementarios para enseñar utilizando a Alice en un espectro de edades y materias con beneficios comprobados en la participación y retención de grupos diversos y desatendidos en educación en ciencias de la computación.

### Características

- utiliza un entorno sencillo basados en arrastrar y en saltos.
  - actualmente es un programa que se ejecuta en Microsoft Window, Linux.
  - puede hacer que los objetos se muevan, giren y cambien de color
- Detail 3.

### Ventajas:

- la interfaz de arrastrar y soltar proporciona un método de construcción de programa que impide que los usuarios realicen errores de sintaxis.

- muestra secuencias del programa como animaciones para que los usuarios puedan ver sus errores y más fácil solucionarlos.
- permite a los estudiantes ver de inmediato cómo se ejecutan sus programas de animación

### Desventajas:

- es solamente en inglés.
- dificultades inherentes al rigor sintáctico a los primeros pasos en la programación orientada a objetos.
- solo acepta tanto el modelo de programación orientada a objetos como la dirigida a eventos.

## CODINGAME



Figura 9 tomada de CODINGAME

Es un sitio dedicado a la programación divertida de computadoras, que ofrece, por un lado, rompecabezas de dificultad creciente para resolver en uno de los veinticinco lenguajes de programación disponibles, y en los otros juegos de inteligencia artificial multijugador. o desafíos de resolución de problemas de tiempo limitado, o código de golf.

### Características:

Las características importantes están definidas por el sistema de logros, modo multijugador, gamificado y gratuito.

### **Ventajas:**

Es compatible con 20 lenguajes de programación entre esos (java, php, Swift), cuenta con sistema de niveles (novato, estudiante, programador profesional) y tiene la opción de elegir entre edición o lenguaje de programación.

### **Desventajas:**

una de sus principales desventajas es necesario conocer al menos un lenguaje de programación.

### **CODE HUNT**



*Figura 10 tomada de Code Hunt*

Code Hunt es una plataforma de juegos serios para la codificación de concursos y la práctica de habilidades de programación. Se basa en el motor de ejecución de caja blanca simbólica, Pex. Code Hunt es único como una plataforma de codificación en línea en el que cada juego se presenta con casos de prueba solamente, ninguna especificación. Los jugadores tienen que trabajar primero el

patrón y luego codificar la respuesta. Code Hunt ha sido utilizado por más de 350.000 jugadores a partir de agosto de 2016. Los datos de los últimos concursos ha sido código abierto para el análisis de la comunidad educativa.

### **Características:**

- Utiliza acertijos que los jugadores tienen que explorar a través de pistas que se presentan como casos de prueba.
- Los jugadores tendrán que modificar el código para que coincida con el resultado esperado del programa.
- Con Code Hunt se tendrá que analizar las pistas que se presentan y modificar el código que se presenta para lograr el resultado correcto.
- El juego está dividido en 14 niveles cada una con un tema de programación cómo, funciones matemáticas, ciclos, condicionales, arreglos, etc.
- El juego va almacenando cuantos intentos se realizan hasta pasar al siguiente nivel y mantiene un tablero de líderes.
- El juego está en el idioma inglés.

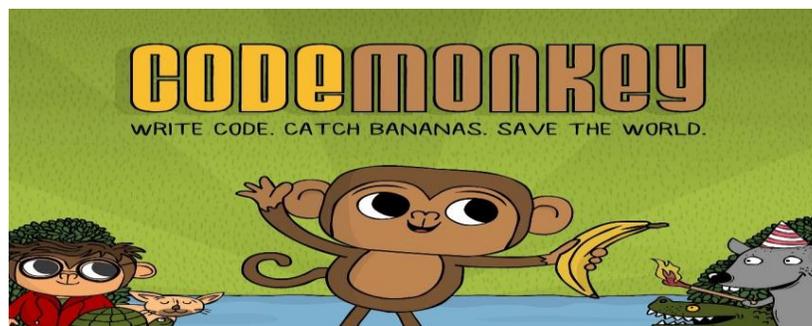
### **Ventajas:**

Permite practicar el analizar, refactorizar, modificar y mejorar piezas de programación, el enunciado del reto está realizado por medio de pruebas unitarias. Es decir, el propio código me indica qué es lo que tiene que hacer.

### **Desventajas:**

Code Hunt no te enseña a programar como tal, pero si te ayuda a mejorar el nivel de programación, sobre todo a aquellos estudiantes que apenas comienzan con esa materia ya que mejorarían su nivel de programación de manera divertida.

## **CODE MONKEY**



*Figura 11 tomada de Code Monkey*

Es una excelente alternativa que podemos sumar a los cientos de servicios y aplicaciones pensados para que los niños se inicien en la programación.

La dinámica que propone Code monkey es simple y entretenida, ya que basa en un juego con una serie de retos que los niños tienen que superar.

**Características:**

La consigna es que el mono siempre obtenga sus plátanos, y para ello habrá que sortear treinta desafíos mientras se van mostrando conceptos básicos de programación. Antes de comenzar cada reto hay pequeños tutoriales, que se repetirán tantas veces sea necesario.

No es necesario crear una cuenta para activar el juego, aunque podemos registrarnos a través de nuestro correo electrónico o simplemente con un nombre de usuario y contraseña. Tanto si los niños juegan en casa con la asistencia de sus padres o forma parte del plan de estudio escolar, es una opción a tener en cuenta. En el caso de los educadores, Codemonkey cuenta con diferentes características que permiten integrar un grupo de estudiantes y dar seguimiento de su progreso. Y además, se integra con Edmodo, lo que nos permite combinar sus funciones y crear un entorno de aprendizaje productivo.

**Ventajas:**

No es necesario crear cuenta, utiliza lenguaje de programación real, aprendizaje basado en juego y constructor de juegos.

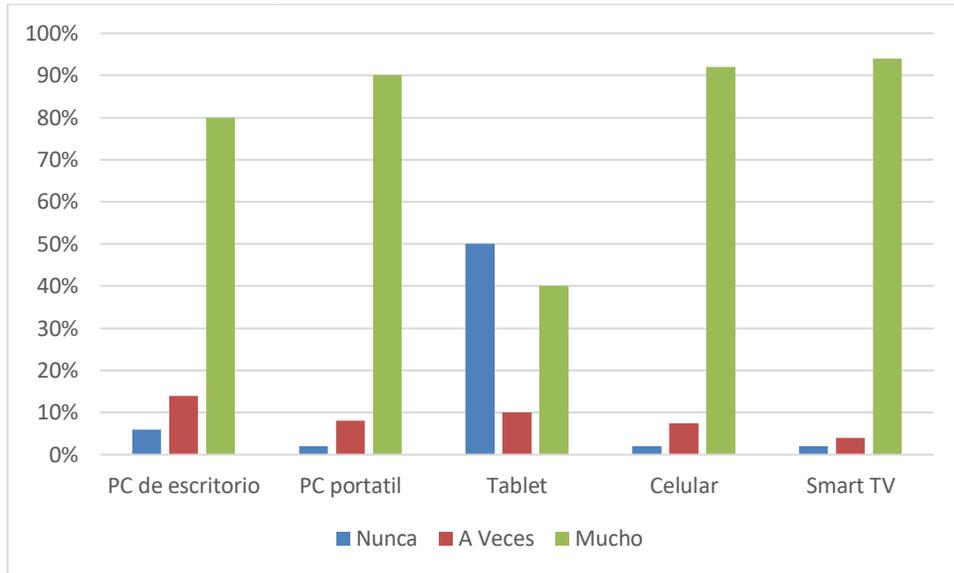
**Desventajas:**

Para tener la versión completa es necesario pagar.

## 6.1 ANÁLISIS DEL DESARROLLO DEL PROYECTO

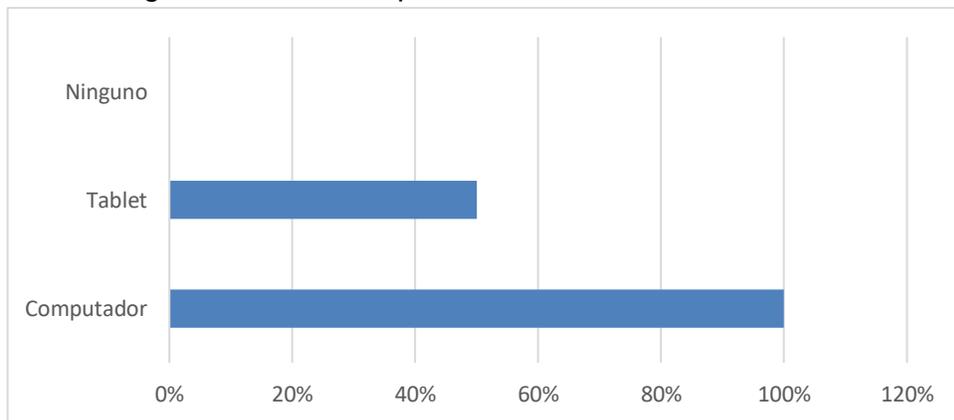
Se realizó una encuesta en la institución educativa Ángel María Paredes de la ciudad de Neiva y en la institución educativa Juan XXIII de Algeciras donde se obtuvo los siguientes resultados:

1. En su día, ¿con qué frecuencia considera que utiliza los siguientes dispositivos tecnológicos? (Marque con una X).



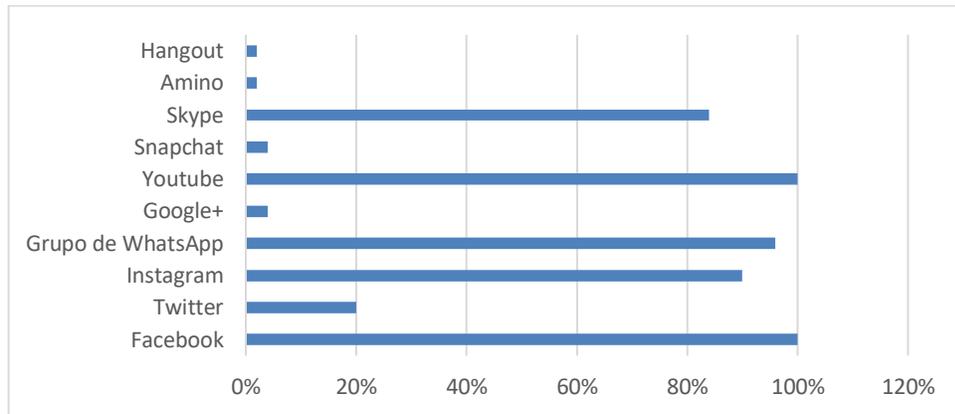
*Grafica 1. Frecuencia de uso de dispositivos tecnológicos*

2. En su colegio tiene facilidad para el uso de:



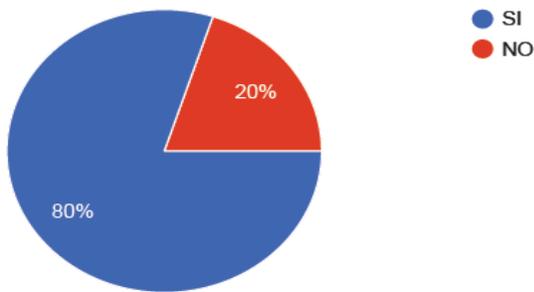
*Grafica 2. Facilidad de uso en el colegio*

3. De las siguientes redes sociales seleccione las que utiliza



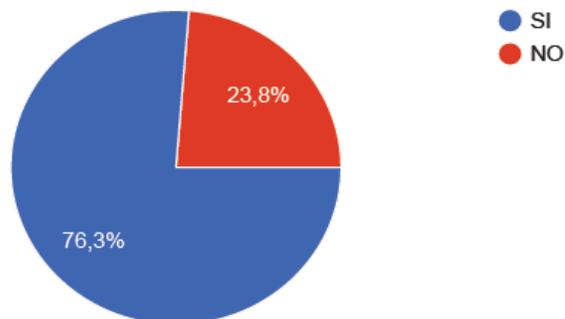
Grafica 3. *Rede sociales que se utiliza con más frecuencia*

4. ¿Le gustan los juegos por Internet?



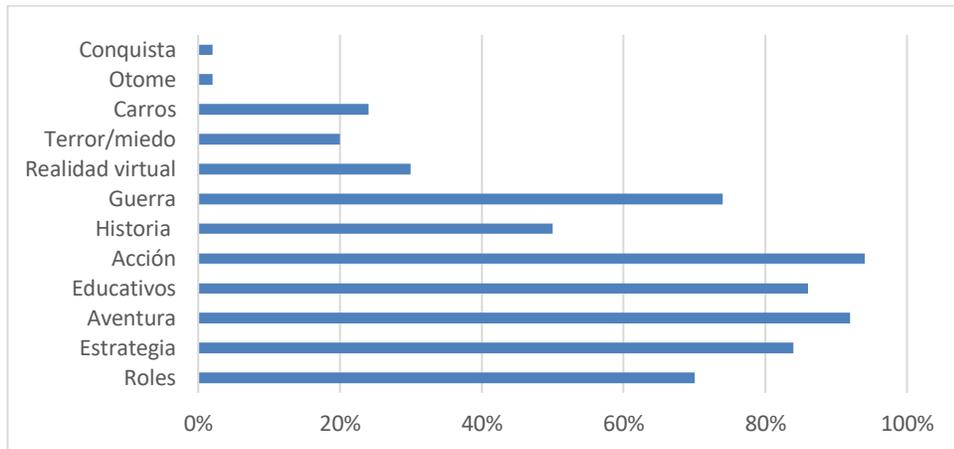
Grafica 4. *Interés en los juegos en la Internet*

5. ¿Le gustaría crear o diseñar sus propios juegos?



Grafica 5. *Interés de crear o diseñar juegos*

Si selecciono (SI) en la pregunta anterior, indique que tipo de juegos le gustaría diseñar.



*Grafica 6. Tipos de juegos que le gustaría diseñar*

## 6.2 CRONOGRAMA

1. CRONOGRAMA DE ACTIVIDADES												
ACTIVIDAD	ME S 1	ME S 2	ME S 3	ME S 4	ME S 5	ME S 6	ME S 7	ME S 8	ME S 9	ME S 10	ME S 11	ME S 12
Búsqueda de información						X						
Diseño de materia de apoyo							X					
Implementación								X				
Evaluación de resultados									X	X		

Tabla 2. Cronograma

## 7. CONCLUSIONES

- En la primera parte del proyecto se logró identificar la metodología ágil Scrum como la más adecuada para el desarrollo del proyecto, esta se ajusta por sus características y adaptación al cambio como base para desarrollo de la estrategia propuesta, para el desarrollo de juegos.
- Se logró identificar conocimientos sobre el uso de herramientas web 2.0, frecuencia de acceso a dispositivos tecnológicos y su gusto por los juegos por internet así mismo el tipo de juego que le gustaría diseñar, de la población seleccionada como muestra.
- Se identifican algunas herramientas o aplicaciones como las más apropiadas para el diseño y desarrollo de juegos de manera fácil y rápida.
- Se diseñó la estrategia para el desarrollo de juegos basada en metodologías ágiles, teniendo como base Scrum.
- Se desarrolla una serie de manuales o guías para el desarrollo de juegos para que los estudiantes pudieran desarrollara los talleres.
- Se hizo la socialización e implementación de la estrategia diseñada en dos colegios, en la institución educativa Ángel María Paredes de la ciudad de Neiva y en la institución educativa Juan XXIII de Algeciras Huila, donde se realizó talleres con un total de 50 estudiantes y se obtuvo resultados óptimos.
- Se logra mediante la encuesta realizada, determinar un alto grado de aceptación a la actividad y se registra el interés por continuar con estas actividades.

## **8. RECOMENDACIONES**

- Continuar con la socialización de la propuesta en colegios para fortalecer la creatividad y el pensamiento lógico, mediante el diseño de juegos.
- Esta actividad se debe programar por los semilleros de investigación para fortalecer la presencia y participación social de la universidad.

## 9. BIBLIOGRAFÍA

Alice – Tell Stories. Build Games. Learn to Program.. (2017). Alice.org. Recuperado de: <https://www.alice.org>

Bolívar Buriticá, W., Chaverra Fernández, D. I., Upegui, M., & Elicenia, M. (2015). Argumentation and use of web 2.0 applications in school. Revista Lasallista de Investigación, 12(1), 58-64. Recuperado de: [http://www.scielo.org.co/scielo.php?pid=S179444492015000100006&script=sci\\_arttext&tlng=pt](http://www.scielo.org.co/scielo.php?pid=S179444492015000100006&script=sci_arttext&tlng=pt)

Canós, J. H., & Letelier, M. C. P. P. (2012). Metodologías ágiles en el desarrollo de software. Recuperado de: <http://roa.ult.edu.cu/bitstream/123456789/476/1/TodoAgil.pdf>

Code.org: Anybody can Learn. (2017). Code.org. Recuperado de: <https://code.org/>

CodeCombat: Learn to Code by Playing a Game. (2017). CodeCombat. Recuperado de: <https://codecombat.com/>

Coding Games and Programming Challenges to Code Better. (2017). CodinGame. Recuperado de: <https://www.codinggame.com/start>

Code Hunt - A game to learn coding!. (2017). Code Hunt. Recuperado de: <https://www.codehunt.com/>

CodeMonkey. (2017). Playcodemonkey.com. Recuperado de: <https://www.playcodemonkey.com/>

GameMaker | YoYo Games. (2017). Yoyo Games. Recuperado de: <https://www.yoyogames.com/gamemaker>

Make Your Own 2d Games With Construct 2. (2017). Scirra.com. Recuperado de: <https://www.scirra.com/construct2>

GDevelop - Crea juegos sin programar - Creador de juegos HTML5 y nativos de código abierto. (2017). En.compilgames.net. Recuperado de: <http://www.en.compilgames.net/main-es.html>

García, Carla; Días, Paulo; Sorte, Antonio; Díaz Pérez, Julian; Rita Leal, Ana; Gandra, Mario (2014) El uso de las TIC y herramientas de la Web 2.0 por maestros portugueses de la educación primaria y educación especial. Recuperado de: <http://digibug.ugr.es/handle/10481/31676>

Kodu | Home. (2017). Kodugamelab.com. Recuperado de: <https://www.kodugamelab.com/>

La guía (2017). Aprendizaje cognitivo. Recuperado de: <http://educacion.laguia2000.com/tipos-de-educacion/aprendizaje-cognitivo>

Letelier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Recuperado de: <http://www.redalyc.org/html/1331/133115027022/>

López-Escribano, C., & Sánchez-Montoya, R. (2012). Scratch y necesidades educativas especiales: Programación para todos. RED, Revista de Educación a Distancia, 34, 1-14. Recuperado de: <http://www.um.es/ead/red/34/scratch.pdf>

Mahalakshmi, M., & Sundararajan, M. (2013). Traditional SDLC Vs Scrum Methodology—A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 192-196. Recuperado de: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.2992&rep=rep1&type=pdf>

Metodologías ágiles para el desarrollo de software (2017). Recuperado de: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Moral Pérez, M. E. D., & Villalustre Martínez, L. (2007). Herramientas de la web 2.0 y desarrollo de proyectos colaborativos en la escuela rural. *Aula Abierta*, 35 (1-2). Recuperado de: <http://digibuo.uniovi.es/dspace/bitstream/10651/26931/1/AulaAbierta.2007.35.1-2.105-16.pdf>

Niños programadores: para qué sirve la enseñanza de programación en las escuelas. (2014). eldiario.es. Retrieved 26 October 2017, from [http://www.eldiario.es/turing/Ninos-programadores-ensenanza-programacion-escuelas\\_0\\_293970921.html](http://www.eldiario.es/turing/Ninos-programadores-ensenanza-programacion-escuelas_0_293970921.html)

Orjuela Duarte, A., & Rojas, M. (2008). Las metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. Revista Avances en Sistemas e Informática, 5(2). Recuperado de: <http://www.redalyc.org/html/1331/133115027022/>

Pérez, M. E. D. M., Cernea, D. A., & Martínez, L. V. (2010). Objetos de aprendizaje 2.0: Una nueva generación de contenidos en contextos conectivistas. Revista de educación a distancia, (25). Recuperado de: <http://revistas.um.es/red/article/view/125301/117241>

Sáez Martínez, P. J. (2013). Identificación y valoración de técnicas ágiles de gestión de proyectos software. Recuperado de: <http://digibuo.uniovi.es/dspace/bitstream/10651/18211/10/TFMPedroJSaezMartinezProteg.pdf>

Scratch - Imagine, Program, Share. (2017). Scratch.mit.edu. Recuperado de: <https://scratch.mit.edu/>

## 10. ANEXOS

### Anexo 1 Estrategia

#### Estrategia fácil rápido

Se deben tener un equipo de mínimo 8 personas.

**Multidisciplinario** (Pueden participar personas con diferentes conocimientos tecnológicos o disciplinares de apoyo).

Seleccionar entre los docentes o personal externo un asesor en la temática de juegos, diseño, didáctica, pedagogía.

#### Entre los integrantes seleccionar:

##### Roles:

- **Líder equipo:** Encargado de la constante comunicación con los integrantes del equipo de trabajo, coordina tiempos en equipo, para el desarrollo de cada una de las fases.  
Es un interlocutor directo con usuario final o su representante.
- **Diseñador** (Participación de dos personas, que toman la responsabilidad del diseño, pero al momento de presentar la idea reciben la realimentación del todo el equipo de trabajo).
- **Programador** (Seleccionar dos integrantes del grupo para la tarea de realizar el programa, Seguir la guía de desarrollo).
- **Representantes del usuario final** (Pueden participar grupo de estudiantes que podrán revisar y hacer aportes).

#### Lugar de trabajo

En cada proceso se llevan a cabo acciones de control y realimentación.

**Ciclo de acciones:** En donde se hace una continua realimentación del proceso, mediante reuniones periódicas, programas según necesidades. Para el mejoramiento, revisión de avances, proponer cambios, asignar metas.

## **Fase 1 inicial**

### **Fase de concepto o Fase de Concepción**

Donde todo comienza a partir de una idea con la cual se conformarán los aspectos fundamentales del juego a diseñar.

- Selección temática a trabajar.
- Establecer tiempos de trabajo.
- Selección del equipo de trabajo.

Allí intervienen todo el equipo de trabajo y de un asesor temático, especialista en una disciplina o un jugador.

Se determina el género o géneros del videojuego, cómo será el proceso de juego, y también se constituye un guion gráfico en el que se tratan todo tipo de ideas preconcebidas que pueden ir adaptándose, como por ejemplo el estilo de los personajes, el ambiente, la música, niveles, etc.

### **Ciclo de acciones.**

Tiempo estimado 1 semana y se ejecuta el ciclo de acciones que se requieran

## **Fase 2**

### **Fase de diseño**

Que elementos se van a usar en el juego. En que está basado el juego, se crean bocetos de guiones para determinar los objetivos, se deciden los personajes principales, el contexto, etc.

Se toman los datos a forma de guion, (paso a paso la actividad), Utilizando estos guiones los diseñadores inician a crear conceptos de cómo se verá el juego, la forma en que se verán los personajes, los escenarios, objetos, etc. Su trabajo es presentar propuestas visuales para ir dando forma a la idea original. Si hay un fondo sonoro, proponerlo para ser implementado.

### **Ciclo de acciones**

Se planifica el funcionamiento del videojuego los personales y los objetivos alcanzados dentro del juego. Para luego hacer el diseño de la programación, basado inicialmente en un mapa mental, en donde se representa la idea general.

### **Ciclo de acciones**

Se hace el diseño de la programación, que describe la manera en la que se implementará el videojuego, el lenguaje, lenguajes de programación o programas que se utilizarán, y si es necesario licencias o costos, las metodologías que se seguirán, etc. Se genera un documento en el cual se especificará el desarrollo del arte, las mecánicas y la programación del juego.

### **Ciclo de acciones**

Tiempo requerido según el proyecto, previa planeación y se ejecuta el ciclo de acciones que se requieran.

## **Fase 3**

### **Planificación**

Tiene como objetivo identificar las diferentes tareas para desarrollar el juego. Se reparte el trabajo entre los distintos componentes del equipo de desarrollo, se fijan plazos de entregas, se planifican reuniones de seguimiento, caracteriza fundamental de una metodología Scrum, aunque no se sigue esta metodología al pie de la letra, está basada en ella.

### **Ciclo de acciones**

## **Fase 4**

### **Elaboración o desarrollo**

- Puesta en marcha de la programación realizada,
- Planificación de las interacciones
- Desarrollo de la Interacción
- Seguimiento a la interacción
- Revisión de la interacción
- Cierre de interacción
- evaluación de cada una de ellas

### **Versiones intermedias documentadas**

- Hacer ajustes de ser necesarios.
- Ajustar errores

## **Ciclo de acciones**

### **Fase 5**

#### **Prueba**

En este espacio se pone a prueba el juego con varios usuarios

### **Fase 6 Cierre**

## Anexo 2. Formato de datos

Formatos inicial

INTEGRANTE	
NOMBRE	
CIUDAD FECHA	
ROL	
CORREO	
FORMACION	
COMENTARIO	

Numero de formato 1

PROPUESTA GENERAL		
GENERALIDADES		
Nombre del juego		
Objetivo del juego		
Número de jugadores		
Tipo de juego		
COMPONENTE PRINCIPAL		
Nombre de la temática		
Propósito		
Instrucciones		
Temática		
MATERIALES		
Nombre	Cantidad	Descripción
REGLAS DE JUEGO		
COMENTARIOS PROXIMA EVALUACION		
1.		
2.		
3.		

### Anexo 3. Formato de encuesta para recolección de información



UNIVERSIDAD NACIONAL ABIERTA AY A DISTANCIA UNAD

Semillero de investigación "Gruslin"

Grupo de investigación "Byte In Design"

Proyecto "Pequeños Navegantes TIC"

1. ¿con qué frecuencia considera que utiliza los siguientes dispositivos tecnológicos? (Marque con una X)

	Nunca	A veces	Mucho
PC de escritorio			
Pc portatil			
Tablet			
Celular			
Smart Tv			

2. En su colegio tiene facilidad para el uso de:

Computador       Tablet       Ninguno

3. De las siguientes redes sociales seleccione la que utiliza.

Redes sociales	Marcar con (X)
Facebook	
Twitter	
Instagram	
Grupo de WhatsApp	
Google+	
Otros	

4. ¿Le gustan los juegos por internet?

SI       No

5. ¿Le gustaría crear o diseñar sus propios juegos?

SI       No

Si selecciono (SI) indique que tipo de juegos le gustaría diseñar.

Tipos de juegos	Marca con una (X)	
Roles	Educativos	
Estrategia	Acción	
Aventura	Historia	
Otros ¿Cuál?		

Institución educativa: \_\_\_\_\_

Grado o curso: \_\_\_\_\_

Fecha: \_\_\_\_\_

## Anexo 4. Formato resultados socialización del proyecto



UNIVERSIDAD NACIONAL ABIERTA AY A DISTANCIA UNAD

Semillero de investigación "Gruslin"

Grupo de investigación "Byte In Design"

Proyecto "Pequeños Navegantes TIC"

1. La socialización de la estrategia fue clara y sencilla, ¿logro sus objetivos?  
SI  No
2. La explicación de la estrategia para el desarrollo de juegos ¿cómo la califica?  
Buena  Muy Buena  Regular  Mala
3. Para el desarrollo del taller, el material disponible ¿cómo lo califica?  
Bueno  Muy Bueno  Regular  Malo
4. ¿cómo califica el taller?  
Bueno  Muy Bueno  Regular  Malo
5. ¿Logro desarrollar la actividad propuesta de forma completa?  
SI  No
6. ¿Le gusto el taller?  
SI  No
7. ¿Quiere que se repita el taller?  
SI  No

## Anexo 5. Manual de creación

### MANUAL DE CREACIÓN

#### 1. LOS ELEMENTOS DEL PROGRAMA

- 1.1. **El escenario:** Es en fondo donde se lleva a cabo el juego, sobre este se encuentran los números en imagen y en letras.
- 1.2. **El personaje principal:** El personaje principal se llama “Pico” y es el encargado en narrar el juego al inicio y al final, da la bienvenida y le informa al jugador cuando ha ganado o perdido la partida.
- 1.3. **Imagen del número:** Las imágenes en número representan a los números en su forma simbólica los cuales deben hacer pareja con su respectivo nombre en letras.
- 1.4. **Palabra del número:** Es la representación de los números en forma alfabética, describe a los números en letras y son las que deben hacer pareja con los números en imagen.

#### 2. PROGRAMACIÓN DEL JUEGO

- 2.1. Asignación de variables:

Para iniciar, en el elemento del escenario se deben crear las variables en la opción de datos:

Se inicia con un evento llamado “**al presionar**” el cual da la orden de dar inicio al juego y luego se agregan las variables con los siguientes nombres:

- **Tiempo:** Es la encargada de cronometrar el tiempo que transcurre desde que inicia el juego.
- **Oportunidades:** Es la encargada de fijar el número de fallos permitidos, disminuye cada vez que el jugador hace un par equivocado.
- **Clic:** Es el encargado de guardar los clics dados a las figuras aleatorias. Un clic guarda y el segundo confirma la pareja.
- **Fig:** Hace referencia a cada una de las figuras de los números, tanto alfabético como numérico.
- **Contador:** Es la encargada de contar las veces que se hace clic acertado en cada figura.

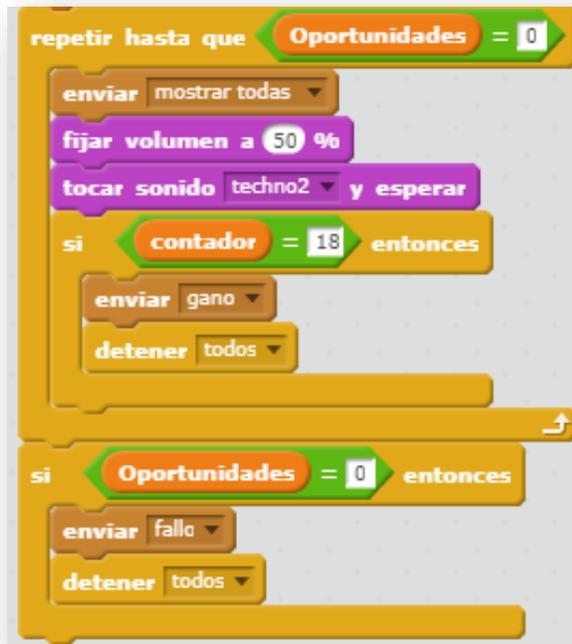


A la variable “**oportunidades**” se le asigna el valor de “3”, el cual corresponde a las opciones de fallas a las que tiene permiso el jugador, una vez terminadas las oportunidades el juego termina.

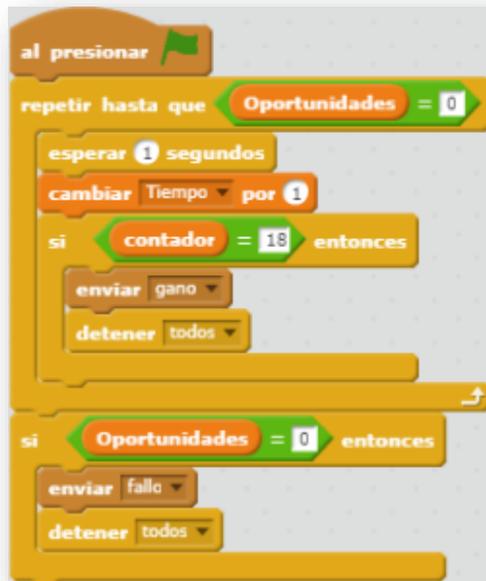
## 2.2. Agregando las condiciones

### 2.2.1. Condiciones del escenario:

- **Condición 1:** Se agrega una opción “**repetir hasta que**” en la cual se declara un operador lógico condicionándolo para que se repita hasta que las oportunidades lleguen hasta “**0 (Cero)**”. Las condiciones fijadas para que se repitan son el inicio del sonido y el volumen.
- **Condición 2:** Se agrega dentro de la “condición 1” para que se repita hasta que las condiciones lleguen a “**0 (Cero)**”. Es la que se encarga de detener el juego si el contador llega a 18 clics acertados. Entonces, si el contador es igual a “**18**”, envía mensaje “ganó” y finaliza el juego.
- **Condición 3:** Se agrega seguida de la “Condición 1” pero afuera. Es la encargada de finalizar el juego si las oportunidades llegan a “**0 (Cero)**”. Entonces, si oportunidades es igual a “**0 (Cero)**” enviar mensaje “Falló” y detener el juego



- **Condición 4:** Es la encargada de llevar el control del tiempo y genera un ciclo que se repite hasta que las oportunidades lleguen a "**0 (Cero)**" o el contador de clics acertados llegue a "**18**". Una vez cumplida cualquiera de estas dos condiciones, el juego se finaliza. Al inicio de la condición se agrega evento "al presionar" para que el ciclo inicie al momento que el jugador inicie el juego.



### 3. AGREGANDO LOS EFECTOS DE SONIDO

#### 3.1. Selección del sonido principal

En el elemento “Escenario” se ubica la pestaña “**Sonido**” y seleccionamos un sonido acorde al ambiente del escenario, en este caso, el sonido seleccionado es:

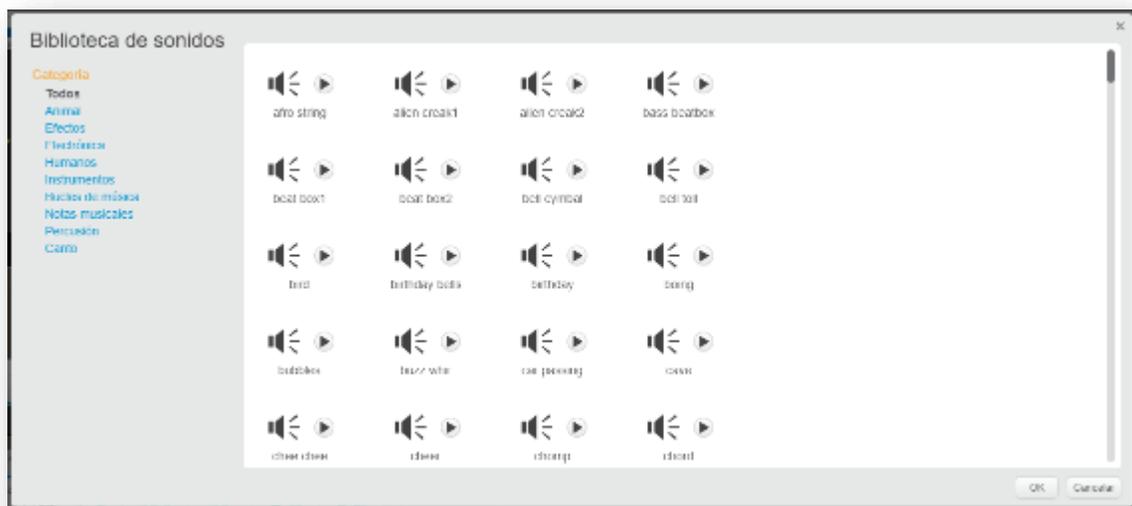
- “Dance child out”



El sonido de ambiente puede ser elegido por el usuario de acuerdo a su criterio.



Una vez abierta la ventana emergente, se selecciona un sonido de la lista.



- 3.2. Programar el inicio de la reproducción del sonido para agregar el sonido se ingresa a la opción de “Sonido” y se agrega la condición de “Volumen” y “comenzar sonido” para que el sonido comience y se repita con las condiciones del ciclo.





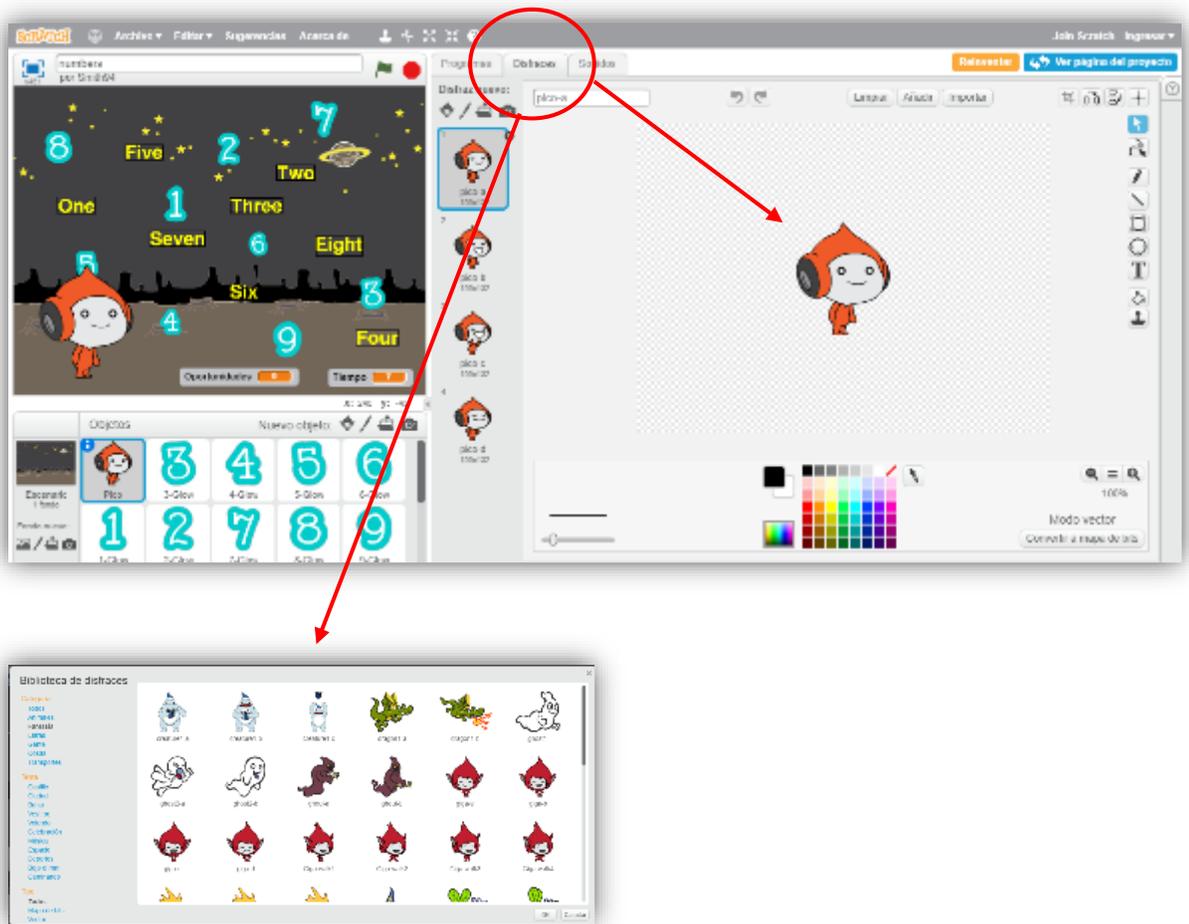
En esta parte del programa se agrega el sonido seleccionado y se ajusta el volumen

#### 4. AGREGAR EL PERSONAJE PRINCIPAL

4.1. Seleccionando el personaje en la pestaña de diseño se selecciona un tipo de personaje acorde al ambiente del escenario y se agregan los gestos para las expresiones programadas.

- Personaje seleccionado: **“Pico”**





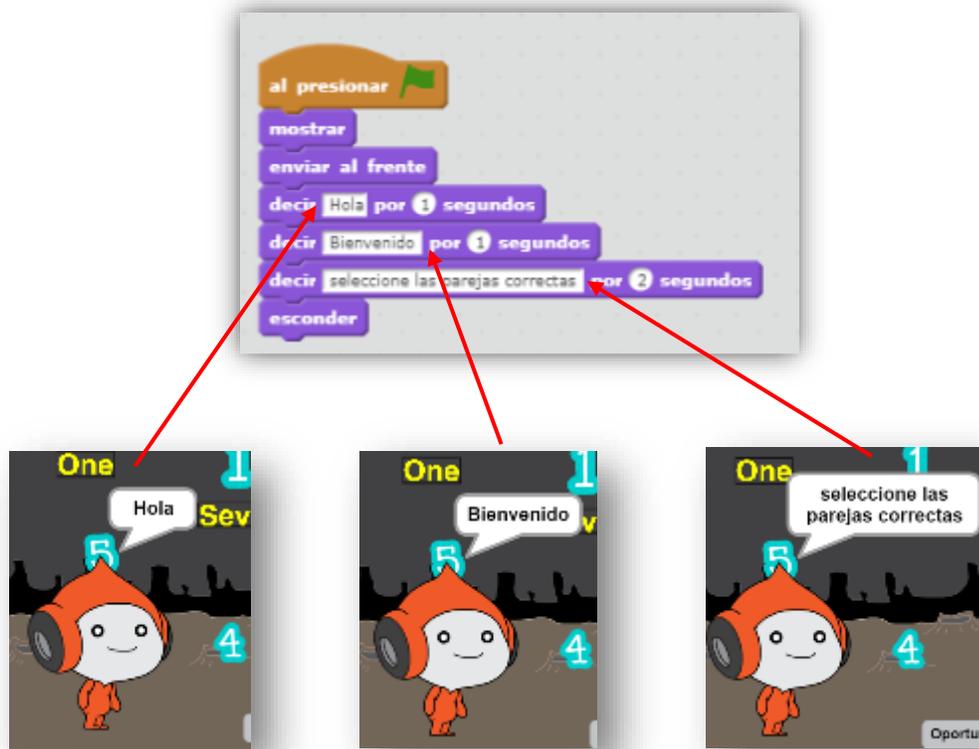
#### 4.2. Programación del personaje principal

El personaje se programa para aparecer en tres momentos del juego:

- Al iniciar
- Al perder
- Al ganar

**Al iniciar:** Para iniciar el personaje, se agrega un evento “Al presionar” para que el personaje sea quien comience con el juego.

Luego se agregan las opciones de “**aparición**” que permita mostrar al personaje y mostrar los mensajes de bienvenida al juego en este caso se programan 3 mensajes cambiantes tras un tiempo determinado y luego se esconda.



**Al perder:** Se agrega un evento y se programa para que al recibir mensaje de “fallo” inicie una condición que muestre al personaje “Pico” al llegar a cero las oportunidades. Al mostrar el personaje se programa para que muestre mensaje “Vuelve a intentarlo”.

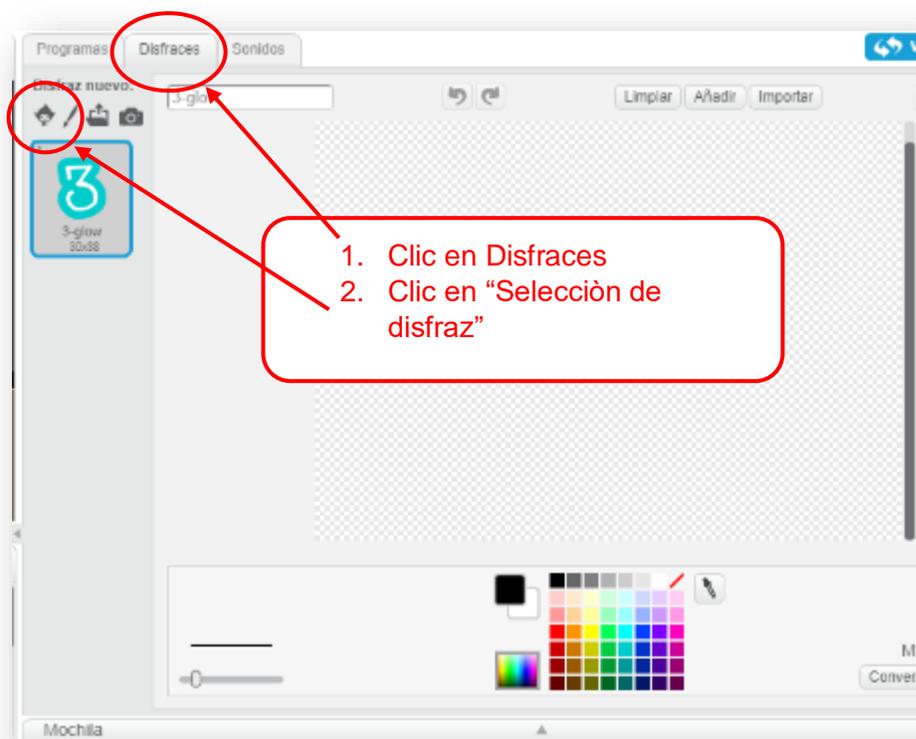


**Al ganar:** Se agrega un evento para que al recibir mensaje “Gano” muestre al personaje con mensaje “Felicidades”.



## 5. LAS IMÁGENES DE LOS NÚMEROS Y LOS NÚMEROS EN LETRAS

5.1. Seleccionando los números y las palabras en la pestaña de "Disfraces" abrimos para seleccionar las imágenes de los números tanto en formato numerico como alfabetico.



Seleccionamos los tipos de letras para los números en texto y numérico del listado de la biblioteca que se abre en una nueva ventana.



Se seleccionan las figuras de los números en formato textual y numérico.



Una vez seleccionados todos los números y palabras, se procede a agregar la programación:

## 6. PROGRAMACIÓN DE LOS NÚMEROS Y LETRAS.

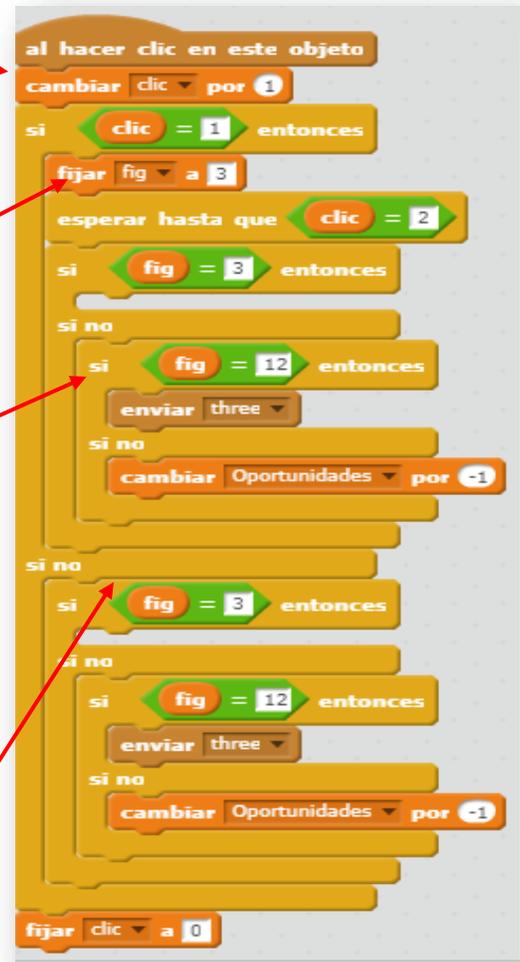
Primero se adiciona un evento “**al presionar**” para que muestre todas las letras y números al iniciar el juego. Ahora se agrega un evento “**al hacer clic en este objeto**” luego se declara una variable “**clic**” y se carga en un dato “**cambiar**” y se asigna valor “**1**” para que este sea denominado como el primer clic para hacer pareja con el segundo clic.

Ahora, se agrega la condición, la cual se activa si se hace clic en la figura y le asigna un valor a la variable “**fig**” dependiendo del número que representa y espera el segundo clic de su pareja.

Se agrega una segunda condición dentro de la primera, la cual dice que, si se hace clic nuevamente sobre la misma figura, no ejecute ninguna acción, pero si hace clic su pareja, ejecute una tercera condición la cual un mensaje con el nombre de su pareja.

Se agrega un evento “**al recibir**” para que el contador sume un punto y esconda las figuras.

Si no se cumple la tercera condición, o sea, si hace clic en una figura diferente a su pareja, el contador descuenta un punto a la variable “**oportunidades**”.



Al no cumplirse la primera condición:

Se agrega una cuarta condición, la cual dice que si la variable “**fig**” tiene un valor de “**3**” no ejecute ninguna acción.

Si no se cumple la cuarta condición, se agrega una quinta condición, la cual dice que, si la variable “**fig**” toma el valor asignado a la figura de su pareja, ejecute un evento con su nombre, el cual activa el evento “**al recibir**”.

Si no se cumple la quinta condición, se agrega un evento que reste un punto a la variable “**oportunidades**”

Al terminar el ciclo de condiciones, el contador de clics se fija a cero.

Esta programación se repite en todas las figuras numéricas y alfabéticas cambiando los valores de las variables al número que corresponde a cada una.

```
al hacer clic en este objeto
cambiar clic por 1
si clic = 1 entonces
  fijar fig a 3
  esperar hasta que clic = 2
  si fig = 3 entonces
    si no
      si fig = 12 entonces
        enviar three
      si no
        cambiar Oportunidades por -1
    si no
      si fig = 3 entonces
        si no
          si fig = 12 entonces
            enviar three
          si no
            cambiar Oportunidades por -1
      si no
        fijar clic a 0
```

## Anexo 6. Manual de usuario

### MANUAL DE USUARIO INSTRUCTIVO PARA JUEGO “NUMBERS”

#### 1. OBJETIVO DEL JUEGO

Permitir al jugador aprender los números en inglés mediante la realización de parejas número-palabra, por medio de los clicks del ratón del ordenador.

#### Módulo de ingreso

No requiere registro para el ingreso al juego

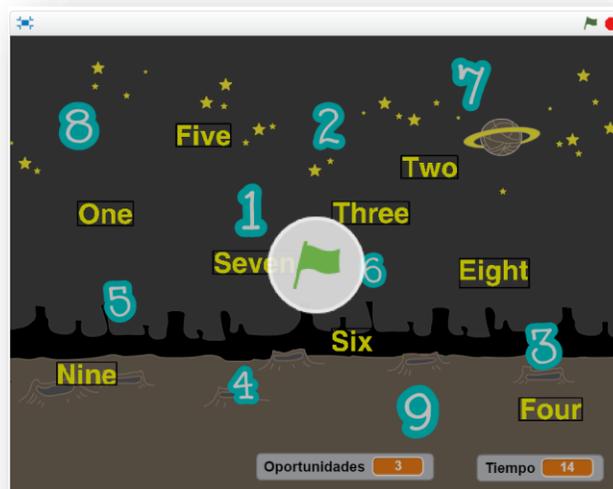
#### 2. INGRESO A LA APLICACIÓN

Para ingresar al juego se debe dar click en el enlace siguiente enlace: <https://scratch.mit.edu/projects/228789278/#fullscreen> y para su ejecución solo se requiere un navegador WEB.

#### Navegadores sugeridos:

- GOOGLE CHROME
- MOZILLA FIREFOX

Al dar click en el enlace se muestra la siguiente pantalla donde encontramos el tablero de juego:



### 3. COMO INICIAR EL JUEGO

Para dar inicio al juego se da un click en la bandera verde que se encuentra en el centro de la pantalla (Ver imagen).

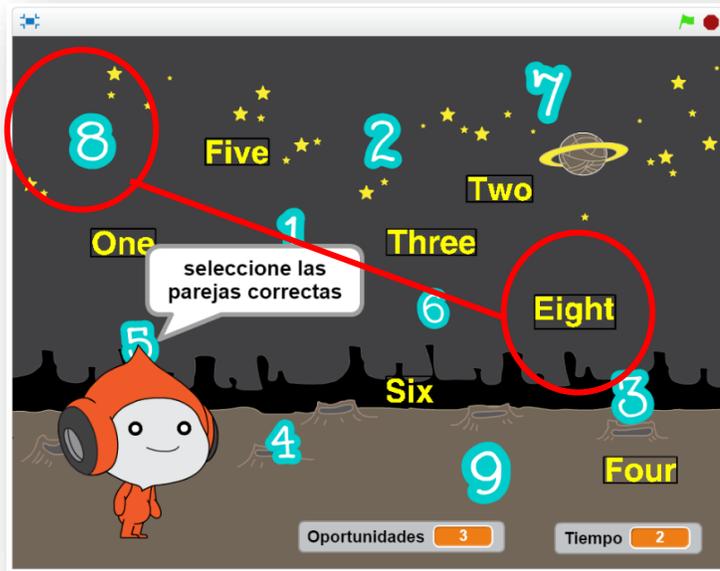


Al dar click en la bandera verde, el juego dará la bienvenida al jugador.



#### 4. COMO JUGAR

Para empezar a jugar se debe observar los números que se encuentran en el entorno, los cuales se encuentran tanto en número como en letras, el jugador debe identificar el número el cual corresponde a su respectiva descripción en letras (Ver imagen).

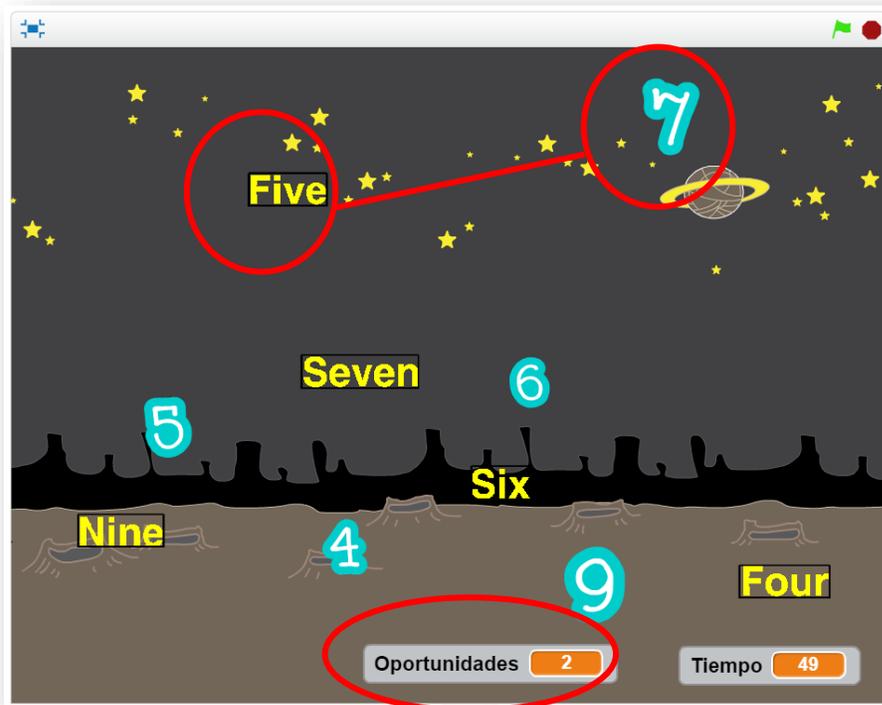


**Aciertos** Al seleccionar la pareja correcta los números van desapareciendo del tablero.



## Fallas

Al seleccionar un número equivocado el sistema disminuye el número de “OPORTUNIDADES”



### Oportunidades

El juego tiene un total de 3 oportunidades de error, al acabarse las oportunidades el juego finaliza.

### El tiempo

El juego cuenta con un contador de tiempo que no depende de ninguna función del juego ni hay límite de tiempo para finalizar el juego. El contador solo marca el tiempo para que el jugador intente romper su propia marca.

## 5. FINALIZAR EL JUEGO

Hay dos maneras de finalizar el juego:

- Ganar: El juego finaliza si se encuentran correctamente todos los pares sin acabar las oportunidades.

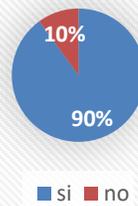


- Perder: El juego finaliza si el jugador agota todas las oportunidades.

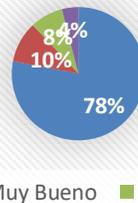


## Anexo 7. Resultados socialización del proyecto

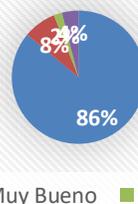
**1. la socializacion de la estrategia fue clara y sencilla, logro sus objetivo?**



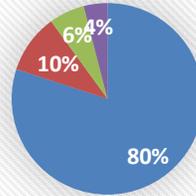
**2. la explicacion de la estrategia para el desarrollo de juegos como la califica?...**



**3. Para el desarrollo del taller el material disponible, ¿como lo califica?...**

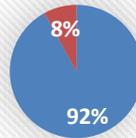


#### 4. ¿como califica el taller?



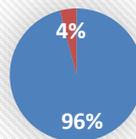
■ Bueno ■ Muy Bueno ■ Regular ■ Malo

#### 5. ¿logro desarrollar la actividad propuesta de forma completa?



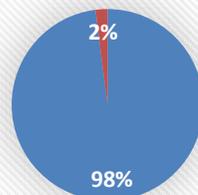
■ si ■ no

#### 6. ¿le gusto el taller?



■ si ■ no

## 7. quiere que se repita el taller?



■ si ■ no

**Anexo 8. Evidencia**





