



Diseño de una propuesta metodológica para el desarrollo de competencias relacionadas con el pensamiento computacional

Maira Isbeth Sarmiento Bolívar

Directora: Cecilia Sanz

Co-directora: Gladys Gorga

**Tesis presentada para obtener el grado de Magister en
Tecnología Informática Aplicada en Educación**

Facultad de Informática - Universidad Nacional de La Plata

2018

DEDICATORIA

A mis padres y a mi hermanita

AGRADECIMIENTOS

Agradezco inmensamente a Dios por guiar mi camino.

A mis amigos y allegados por su apoyo.

A mi directora Cecilia Sanz y codirectora Gladys

Gorga por su ayuda.

A mi colegio IED Pío XII, la profesora Diana Ahumada

y mis estudiantes por hacer parte de este proyecto.

Índice de contenidos

DEDICATORIA.....	I
AGRADECIMIENTOS	III
ÍNDICE DE CONTENIDOS	V
ÍNDICE DE FIGURAS	VIII
CAPÍTULO 1	1
1.1. INTRODUCCIÓN	2
1.2. OBJETIVO.....	2
1.2.1. <i>Objetivo General</i>	2
1.2.2. <i>Objetivos Específicos</i>	3
1.3. MOTIVACIÓN.....	3
1.4. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	5
1.5. METODOLOGÍA DE LA INVESTIGACIÓN	6
1.5.1. <i>Estudio de caso</i>	7
1.5.2. <i>Selección de la bibliografía</i>	7
1.5.2.1. Preguntas de investigación.....	7
1.5.2.2. Estrategia de búsqueda.....	8
1.5.2.3. Criterios de inclusión y exclusión.....	9
1.5.2.4. Proceso preliminar.....	10
1.5.2.5. Proceso de selección final.....	10
CAPÍTULO 2	11
2.1. INTRODUCCIÓN	12
2.2. ESTRATEGIAS METODOLÓGICAS UTILIZADAS EN LAS INICIATIVAS REVISADAS	13
2.3. DISCUSIÓN SOBRE LAS METODOLOGÍAS.....	39
2.4. CONCLUSIONES	44
CAPÍTULO 3.....	45
3.1. INTRODUCCIÓN	46
3.2. DESCRIPCIÓN DE HERRAMIENTAS DE SOFTWARE PARA LA ENSEÑANZA DE LA PROGRAMACIÓN	46
3.2.1. <i>Lenguajes y entornos de programación</i>	46

3.2.2. <i>Lenguajes de programación visual</i>	53
3.3. CONCLUSIONES	63
CAPÍTULO 4	64
4.1. INTRODUCCIÓN	65
4.2. ANTECEDENTES.....	66
4.2.1. <i>El modelo TPACK</i>	67
4.2.2. <i>La propuesta de Manso et al. (2011)</i>	69
4.2.3. <i>Modelo de proceso de desarrollo de software en cascada</i>	70
4.3. METODOLOGÍA PARA EL DISEÑO DE EXPERIENCIAS QUE FOMENTEN EL DESARROLLO DEL PENSAMIENTO COMPUTACIONAL EN NIVEL INICIAL Y MEDIO	71
4.3.1. <i>Descripción detallada de la metodología propuesta</i>	72
4.3.1.1. Definición de requerimientos	74
4.3.1.2. Diseño y aplicación de la experiencia educativa	76
4.3.1.3. Evaluación de la propuesta	80
4.3.1.4. Ajuste.....	82
4.4. CONCLUSIONES	83
CAPÍTULO 5	85
5.1. INTRODUCCIÓN	86
5.2. DESCRIPCIÓN GENERAL DEL ESTUDIO DE CASO.....	86
5.2.1. <i>Diseño del estudio de caso</i>	87
5.2.2. <i>Descripción de los participantes estudiantes</i>	88
5.2.3. <i>Descripción del perfil de la docente interviniente</i>	91
5.2.4. <i>Descripción breve del contexto en el que se desarrolla el caso</i>	92
5.3. APLICACIÓN DE LA METODOLOGÍA EN EL ESTUDIO DE CASO.....	92
5.4. CONCLUSIONES	103
CAPÍTULO 6	104
6.1. INTRODUCCIÓN	105
6.2. RESULTADOS DEL TRABAJO REALIZADO ENTRE DOCENTES CON LA METODOLOGÍA PROPUESTA.....	107
6.2.1. <i>Análisis sobre el nivel de satisfacción</i>	112
6.2.1.1. Nivel de satisfacción respecto a las actividades realizadas en cada una de las sesiones	112

6.2.1.2. Nivel de satisfacción respecto del trabajo realizado en equipo.	114
6.2.1.3. Nivel de satisfacción respecto a los dispositivos utilizados en las actividades.	117
6.2.1.4. Nivel de satisfacción respecto de los entornos de programación utilizados.	119
6.2.2. <i>Análisis sobre el nivel de dificultad encontrado en las actividades y dinámicas propuestas</i>	121
6.2.2.1. Nivel de dificultad en relación a las actividades de lectoescritura.	121
6.2.2.2. Nivel de dificultad en relación a hacer uso de bloques de programación.	123
6.2.2.3. Nivel de dificultad en relación a la solución de los desafíos propuestos.	125
6.2.2.4. Nivel de dificultad en relación a la abstracción de información para la creación literaria.....	127
6.2.2.5. Nivel de dificultad en relación al diseño del cuento en Scratch.	128
6.3. RESULTADOS DEL PRETEST - POSTEST.....	129
6.4. RESULTADOS CON RELACIÓN A LOS OBJETIVOS DEL ESTUDIO DE CASO.....	133
6.5. CONCLUSIONES.....	135
CAPÍTULO 7.....	137
7.1. INTRODUCCIÓN.....	138
7.2. APORTES DE LA INVESTIGACIÓN.....	139
7.3. CONCLUSIONES.....	140
7.4. LÍNEAS DE TRABAJO O DE INVESTIGACIÓN A FUTURO.....	142
<i>REFERENCIAS BIBLIOGRÁFICAS</i>	144
<i>ANEXO I</i>	152
<i>ANEXO II</i>	155

Índice de Figuras

Capítulo 2

Figura 2.1 - Alcances proyecto Program.ar. Tomada del sitio web del proyecto	18
Figura 2.2 - Cobertura PRONIE 2016 tomado de los autores de referencia.....	26
Figura 2.3 - Indicadores de liderazgo tomado del sitio web del proyecto	35
Figura 2.4 - Herramientas utilizadas en las iniciativas. Elaboración propia.	40
Figura 2.5 - Estrategias metodológicas utilizadas. Elaboración propia.....	41
Figura 2.6 - Productos y actividades realizadas. Elaboración propia.....	41
Figura 2.7-Desarrollo de competencias relacionadas con el pensamiento computacional.	42
Figura 2.8 – Técnicas e instrumentos utilizados para la evaluación de las propuestas	43

Capítulo 4

Figura 4.1 - Marco TPACK y sus saberes que lo componen	67
Figura 4.2 - Etapas de la propuesta metodológica	74

Capítulo 5

Figura 5.1 - Edad de los participantes.....	89
Figura 5.2 - Zona de vivienda de los participantes	89
Figura 5.3 - Zona de vivienda de los participantes	90
Figura 5.4 - Conectividad con que cuentan los participantes	90
Figura 5.5 - Reconocimiento de entornos de programación	91

Capítulo 6

Figura 6.1 - Esquema de recorrido del capítulo.....	106
Figura 6.2 – Prueba con escala de Likert 1 y 2. Indica cómo te sentiste con respecto a trabajar en equipos de 2 personas.....	115
Figura 6.3 - Prueba con escala de Likert 1 y 2. Llegar a acuerdos con el compañero de equipo.....	116
Figura 6.4 – Prueba con escala de Likert 1 y 2. Contar con un compañero de trabajo o trabajar en equipo con un compañero.....	116

Figura 6.5 – Prueba con escala de Likert 1. Trabajar en la sala de informática.....	117
Figura 6.6 – Prueba con escala de Likert 1. Poder trabajar con las computadoras.	118
Figura 6.7 - Prueba con escala de Likert 1. Ver las explicaciones con ayuda del Video Beam.	118
Figura 6.8 – Prueba con escala de Likert 1 y 2. Utilizar Pilas Bloques, Minecraft y Scratch.	120
Figura 6.9 – Prueba con escala de Likert 1 y 2. Conocer diferentes entornos de programación (Pilas bloques, Minecraft y Scratch)	121
Figura 6.10 - Prueba con escala de Likert 1 y 2. Realizar una creación literaria	122
Figura 6.11 - Prueba con escala de Likert 1 y 2. Nivel de dificultad al utilizar bloques de comando.....	123
Figura 6.12 - Prueba con escala de Likert 1 y 2. Utilizar bloques de repetición.....	124
Figura 6.13 - Prueba con escala de Likert 1 y 2. Utilizar bloques de alternativa condicional	124
Figura 6.14 - Prueba con escala de Likert 1 y 2. Entender la explicación de las profes .	126
Figura 6.15 - Prueba con escala de Likert 1. Tomar información de las escenas de Pilas Bloques y Minecraft para escribir el cuento.....	127
Figura 6.16 - Prueba con escala de Likert 1. Aprender a escribir una narración con base en las escenas de los desafíos	128
Figura 6.17 - Prueba con escala de Likert 2. Utilizar Scratch para animar el cuento.....	128
Figura 6.18 - Prueba con escala de Likert 2. Realizar animaciones en Scratch	129
Figura 6.19 - Prueba con escala de Likert 2. Mostrar el resultado del trabajo realizado. .	129
Figura 6.20 - Uso de la computadora en la solución de problemas	130
Figura 6.21 - Abstraer la información relevante para reconocer datos principales	131
Figura 6.22 - Diseñar algoritmos para llegar de manera ordenada a la solución	131
Figura 6.23 - Descomponer procesos o problemas en partes pequeñas y manejables..	132
Figura 6.24 - Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común	132
Figura 6.25 – Aprendizajes alcanzados en las sesiones	133

Índice de tablas

Capítulo 2

Tabla 2.1 sesiones de trabajo propuesta programando con robots. Tomada de los autores de referencia.....	16
Tabla 2.2 Descripción de las sesiones de trabajo, tomada de Capot y Espinoza, 2015. ...	27
Tabla 2.3 Resultados. Tomada de autores de referencia	28
Tabla 2.4 Actividades taller con Scratch. Tomada de autores de referencia	29
Tabla 2.5 Unidades del taller y sus actividades asociadas. Tomada de autores de referencia.....	30
Tabla 2.6 Histórico de la cátedra de Fundamentos de Programación en Universidad de Valparaíso. Tomada de autores de referencia.	31
Tabla 2.7 Resultados en la evaluación de la experiencia, tomados de (Muñoz et al., 2015).	31
Tabla 2.8 Medición de calidad, material de estudio. Tomada de autores de referencia....	33
Tabla 2.9 Metodología de trabajo en el proyecto, Tomada de autores de referencia.	36
Tabla 2.10 Experiencias revisadas. Elaboración propia.	44

Capítulo 4

Tabla 4.1 características relacionadas con el pensamiento computacional.....	75
Tabla 4.2 Criterios recomendados en el momento de elegir las herramientas a utilizar ...	79
Tabla 4.3 Herramientas recomendadas	79
Tabla 4.4 Técnicas recomendadas para el seguimiento de las actividades.	81

Capítulo 5

Tabla 5.1 Sesión 1. Trabajo con los estudiantes.....	96
Tabla 5.2 Sesión 2. Trabajo con estudiantes	97
Tabla 5.3 Sesión 3. Trabajo con estudiantes	98
Tabla 5.4 Sesión 4. Trabajo con estudiantes	99
Tabla 5.5 Sesión 5. Trabajo con los estudiantes.....	100
Tabla 5.6 Sesión 6. Trabajo con estudiantes	101
Tabla 5.7 Sesión 7. Trabajo con estudiantes	102

Capítulo 6

Tabla 6.1 Entrevista 1 y 2. ¿Cómo te pareció el trabajo que hicimos hoy con tus compañeros? ¿Por qué?	112
Tabla 6.2 Entrevista 1 y 2. ¿Cuál fue la actividad que más te gustó? ¿Por qué?	113
Tabla 6.3 Entrevista 1 y 2. ¿Cuál fue la actividad que menos te gustó? ¿Por qué?	113
Tabla 6.4 Entrevista 1 y 2. ¿Te gustó trabajar con otro compañero? ¿Por qué?	114
Tabla 6.5 Entrevista 1. ¿Cómo te pareció el entorno de programación? ¿Pudiste solucionar los desafíos que se propusieron?	119
Tabla 6.6 Entrevista 2. ¿Cómo te pareció el entorno de programación Scratch? ¿Fue fácil o difícil utilizarlo?	119
Tabla 6.7 Entrevista 1. ¿Qué fue lo que más te gustó de este entorno de programación? (Pilas Bloques).....	119
Tabla 6.8 Entrevista 2. ¿Qué fue lo que más te gustó y qué no te gustó de este entorno de programación? (Scratch)	120
Tabla 6.9 Entrevista 1. ¿Cómo te quedó el inicio del cuento que escribiste? ¿Fue fácil hacerlo?.....	122
Tabla 6.10 Entrevista 2. ¿Pudiste animar el principio de tu cuento en Scratch? ¿Lo terminaste?.....	122
Tabla 6.11 Entrevista 1 primera sesión. ¿El tiempo que te dieron fue suficiente para terminar los desafíos que debías trabajar?	125
Tabla 6.12 Entrevista. ¿Cuál es el título de tu cuento? ¿Por qué elegiste ese nombre?	127

Resumen

Esta investigación se enfoca en el diseño de una propuesta metodológica para desarrollar competencias vinculadas al pensamiento computacional tales como formular problemas posibles de solución mediante una computadora, abstraer la información relevante para reconocer los datos principales y diseñar algoritmos para llegar de manera ordenada a la solución, entre otros. Estas competencias resultan de utilidad no solo en el ámbito académico sino también en el entorno personal.

Por otra parte, desde hace algunos años los gobiernos alrededor de Iberoamérica interesados en generar cambios pedagógicos en el aula han impulsado mediante programas y políticas públicas proyectos a gran escala en los que se pretende integrar las Tecnologías de la Información y la Comunicación (TIC) en el sistema educativo, como es el caso de los proyectos program.AR (Argentina), Computadores para Educar (Colombia) y Plan Ceibal (Uruguay).

Este documento presenta en un primer momento algunas iniciativas actuales donde se revisan metodologías enfocadas a la enseñanza de la programación y al acercamiento al pensamiento computacional. De esta manera se analizan proyectos con resultados satisfactorios que aportan al núcleo temático de la tesis.

Después de la revisión de las iniciativas, se describen algunas de las herramientas utilizadas en ellas y se investigan otras encontradas posteriormente, abordándolas desde dos categorías: lenguajes de programación y lenguajes de programación basados en bloques de arrastrar y soltar. Para cada uno de estos lenguajes se describen aspectos como: entorno utilizado, facilidad de uso, disponibilidad (libres y gratuitas), requerimientos de software y hardware, entre otras, así como también la información técnica disponible (manuales, videos explicativos y material de apoyo a docentes) que facilitan su aprendizaje.

A partir de la investigación previa, se realiza una propuesta metodológica orientada al diseño, la planificación de estrategias y actividades para fomentar el desarrollo del pensamiento computacional. Esta propuesta guía a los docentes a partir de una serie de preguntas orientadoras en la toma de decisiones para la

planificación de las sesiones que fomenten el desarrollo del pensamiento computacional. La metodología propuesta retoma el modelo TPACK, la propuesta de Manso et al. (2011), y las etapas del modelo de proceso de desarrollo de software en cascada.

Luego, se describe la aplicación de la metodología a un estudio de caso para conocer sus limitaciones y posibilidades desde el punto de vista de los docentes y el logro de los objetivos propuestos para los estudiantes.

La aplicación de la metodología se realizó en un espacio académico llamado Club de lectura “Cuenta un cuento con tecnología”, el cual se desarrolló con la colaboración de una docente de Lengua Castellana de la Institución Educativa Departamental Pío XII, en el municipio de Pacho, Cundinamarca, Colombia. La docente participante tomó las decisiones para el diseño de la experiencia educativa a partir de las etapas planeadas por la metodología propuesta y contó con la asesoría de la tesista en la parte tecnológica.

De esta manera se llevó adelante la implementación durante siete sesiones, donde participaron 16 estudiantes de sexto grado de básica secundaria, en las cuales se pudo recopilar información mediante diversas técnicas e instrumentos. Finalmente se elaboraron conclusiones y se plantearon trabajos a futuro.

Capítulo 1

Introducción

1.1. Introducción

Por el interés de los países en integrar las TIC en la educación se han generado políticas que dotan a las instituciones educativas de computadoras, *netbooks*, *notebooks*, *tablets*, tableros digitales y otras herramientas que han buscado transformar a los entornos de aprendizaje (Lugo, 2015).

Con las herramientas tecnológicas que se encuentran en las instituciones, se han generado iniciativas que mejoran la manera en que los estudiantes interactúan y hacen uso de las potencialidades que les brinda la tecnología para dar solución a problemas de su entorno.

Al mismo tiempo, alrededor de Iberoamérica se implementan proyectos educativos que tienen como fin desarrollar habilidades, conocimientos y actitudes relacionadas con el pensamiento computacional, competencias que son necesarias en el momento de dar solución a problemas haciendo uso de herramientas informáticas. Estas iniciativas en su mayoría han dado resultados satisfactorios, motivo por el cual se continúan implementando, mejorando y replicando (Sarmiento, Gorga, y Sanz, 2016).

Esto motiva en esta tesis a plantear los objetivos que se describen en la siguiente sección.

1.2. Objetivo

1.2.1. Objetivo General

Desarrollar una metodología que permita llevar adelante experiencias que pongan en juego el desarrollo de conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional.

Se realizará un estudio de caso con estudiantes de secundaria de la IED Pío XII en el municipio de Pacho, Cundinamarca, Colombia.

1.2.2. Objetivos Específicos

- Revisar y hacer un estudio comparativo de herramientas que han sido utilizadas en algunas iniciativas para el desarrollo de habilidades relacionadas con el pensamiento computacional.
- Analizar antecedentes de estrategias metodológicas aplicadas para el desarrollo del pensamiento computacional.
- Reconocer qué habilidades se han desarrollado o no con el uso dichas estrategias metodológicas.
- Diseñar una metodología que guie a los docentes en la planificación de experiencias con TIC para el desarrollo del pensamiento computacional.
- Utilizar la metodología diseñada y aplicarla a un estudio de caso para su evaluación.
- Realizar el análisis de los resultados obtenidos.

1.3. Motivación

Desde hace algunos años los gobiernos alrededor de Iberoamérica interesados en generar cambios pedagógicos en el aula han impulsado mediante programas y políticas públicas proyectos a gran escala en los que se pretende integrar las Tecnologías de la Información y la Comunicación (TIC) en el sistema educativo (Lugo, Toranzos, Lopez, y Corbetta, 2014), como es el caso de los proyectos program.AR (Argentina), Computadores para Educar (Colombia) y Plan Ceibal (Uruguay), entre otros. Es así como por medio de estos programas se han dotado a las instituciones educativas y a sus estudiantes de equipos tecnológicos como: *tablets*, computadoras portátiles, *netbooks* y otros.

A partir de estas herramientas tecnológicas, se han generado propuestas particulares que se desprenden de estos proyectos macro promovidos por el estado. El objetivo de algunas de estas iniciativas, además de integrar las TIC en el aula, es promover propuestas educativas de calidad que modifiquen la manera de enseñar y aprender, aportando en el desarrollo de conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional. Se define al

pensamiento computacional como “los procesos de pensamiento que intervienen en la formulación de problemas y sus soluciones, donde las soluciones están representadas en una forma que pueden llevarse a cabo de manera efectiva por medio de un agente de procesamiento de información” (Wing, 2011).

Se ha realizado el análisis de algunas de los proyectos e iniciativas desarrolladas alrededor de Iberoamérica por medio de la revisión y estudio abordado en el trabajo final integrador para alcanzar el grado de Especialista en Tecnología Informática Aplicada en Educación, titulado: “**Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica**”. En este trabajo se da a conocer que algunos de los conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional, no son abordadas o son poco abordadas por las estrategias implementadas en estas experiencias.

Retomando el documento referido, los conocimientos, habilidades y actitudes que según diferentes autores e instituciones reconocidas deben desarrollarse para que un estudiante posea competencias relacionadas al pensamiento computacional son (Barr y Stephenson, 2011; Wing, 2011; ISTE y CSTA, 2011; Kemp, 2014; Sarmiento, Gorga, y Sanz, 2016):

- Formular problemas posibles de solución mediante una computadora.
- Abstractar la información relevante para reconocer los datos principales.
- Diseñar algoritmos para llegar de manera ordenada a la solución.
- Descomponer datos, procesos (modularizar) o problemas en partes pequeñas y manejables.
- Simular modelos de las posibles soluciones antes de ponerlos a prueba.
- Implementar los modelos para saber si la solución propuesta es eficiente y efectiva.
- Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos.
- Persistir en el trabajo con problemas difíciles.

- Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

A partir de la investigación previa, se ha encontrado un área de necesidad donde se visualiza que existen pocas propuestas que promuevan el pensamiento computacional trabajando en equipo para alcanzar una meta o solución común. Por otra parte, se ha visualizado una carencia en la evaluación de las estrategias y experiencias analizadas para saber si éstas dieron respuesta al o los problemas planteados.

Por este motivo, se hace importante diseñar estrategias que desarrollen y a su vez fortalezcan los conocimientos, habilidades y actitudes que no se trataron o se trabajaron someramente en las propuestas y iniciativas revisadas, de modo que puedan ser aplicadas y contribuyan en la formación de estudiantes competentes al momento de dar solución a problemas por medio de herramientas informáticas. Al mismo tiempo, la metodología deberá contar con una etapa de evaluación de las estrategias planteadas para analizar el alcance de los objetivos propuestos.

La metodología será puesta en juego en un estudio de caso en la Institución Educativa Departamental Pío XII de Pacho, Cundinamarca, Colombia.

1.4. Planteamiento del problema y justificación

Las propuestas que se han venido trabajando en pro del desarrollo de habilidades, conocimientos y actitudes relacionadas con el pensamiento computacional alrededor de Iberoamérica han generado iniciativas que se deben tener en cuenta, puesto que según la información dada a conocer en la evaluación de cada proyecto los resultados obtenidos fueron favorables.

La mayoría de estas iniciativas han abordado las características que según varios autores favorecen a los estudiantes para que se conviertan en personas competentes a la hora de dar solución a problemas de su entorno haciendo uso de agentes de procesamiento de información. Aunque la finalidad de las iniciativas realizadas fue desarrollar la totalidad de estas características, en algunos casos estas no se lograron afianzar o se trabajaron someramente, es por esto que se

hace importante diseñar metodologías utilizando herramientas y estrategias que sirvan para fortalecer aquellas competencias que en las propuestas revisadas no se consiguieron desarrollar en su totalidad y de esta manera lograr que los estudiantes sean competentes en el uso de las tecnologías que están al alcance de sus manos, de modo que tengan la capacidad de generar propuestas de solución a necesidades en su entorno.

Para centrar el diseño de la metodología en las habilidades menos abordadas o no abordadas que caracterizan al pensamiento computacional se tiene en cuenta el análisis realizado en el trabajo final integrador “Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica”, donde se dan a conocer las competencias que deben ser fortalecidas.

Teniendo en cuenta que metodologías de este tipo no existen en el contexto escolar donde se aplicará la propuesta y siendo este un campo poco explorado, se hace importante que al diseñar la metodología a ser aplicada se utilicen estrategias contextualizadas, pensadas con base en las características de los estudiantes y aprovechando eficientemente las herramientas que fueron entregadas a la Institución Educativa Departamental Pío XII por medio del programa Computadores Para Educar (CPE).

1.5. Metodología de la investigación

Esta investigación propone realizar un trabajo experimental en el que se diseñará una metodología que pretende desarrollar las habilidades, conocimientos y actitudes que caracterizan al pensamiento computacional. En un primer momento se revisarán las herramientas utilizadas en iniciativas que han trabajado el desarrollo de estas competencias, seguidamente se revisará como fueron utilizadas en las diversas experiencias, reconociendo las posibilidades que tienen al momento de promover el pensamiento computacional. Al mismo tiempo, por medio del uso de estas mismas herramientas se puede generar una nueva metodología que logre desarrollar y fortalecer las competencias que no fueron completamente abordadas en las iniciativas revisadas.

Es por ello que se diseñará una metodología cuyas herramientas y estrategias se adapten al contexto en el cual se va a aplicar y que, por supuesto, se enfoquen al desarrollo de las competencias relacionadas al pensamiento computacional. Luego, se llevará a cabo un estudio de caso en el cual se pondrá en juego la metodología diseñada y de esta manera lograr una descripción del fenómeno objeto de estudio, a partir del cual se busca analizar las potencialidades de la metodología propuesta para el desarrollo del pensamiento computacional.

Por último, se considerarán los resultados obtenidos y se obtendrán las conclusiones y líneas de trabajo futuro.

1.5.1. Estudio de caso

El estudio de caso se llevará a cabo con estudiantes de sexto grado de básica secundaria, cuya franja etaria se encuentra entre 10 y 12 años, alumnos de la Institución Educativa Departamental Pío XII en Pacho, Cundinamarca, Colombia.

1.5.2. Selección de la bibliografía

La selección de la bibliografía se realizará siguiendo pautas de la etapa de planificación de la revisión, según la metodología de Barbara Kitchenham (2004).

1.5.2.1. Preguntas de investigación

Q1: ¿Qué iniciativas para el desarrollo del pensamiento computacional se encuentran alrededor de Iberoamérica?

Q2: ¿Cuáles son las competencias menos consideradas en estas iniciativas en relación al pensamiento computacional?

Q3: ¿Cuáles son las herramientas tecnológicas utilizadas y qué características específicas tienen estas herramientas?

Q4: ¿Cuáles son las estrategias metodológicas usadas en las iniciativas encontradas?

Q5: ¿Qué necesidades o mejoras se requieren en estas estrategias metodológicas?

Q6: ¿De qué manera pueden mejorarse y cuán efectivas son estas mejoras?

1.5.2.2. Estrategia de búsqueda

- Fuentes documentales

Para realizar la selección se hizo una búsqueda en:

- o Librerías digitales

- IEEE Explore Digital Library* (Instituto de Ingeniería Eléctrica y Electrónica)¹

- ACM Digital Library (Association for Computing Machinery)*²

- o Artículos publicados por diversas comunidades científicas

- JENUI (Jornadas sobre la Enseñanza Universitaria de la Informática)³

- RIBIE (Red Iberoamericana de Informática Educativa)⁴

- o Artículos publicados en actas de congresos

- Congreso Iberoamericano de ciencia y tecnología

- Congreso de Tecnología en Educación y Educación en Tecnología

- CACIC (Congreso Argentino de Ciencias de la Computación)⁵

- TISE Congreso Internacional de Informática Educativa.⁶

- o Repositorios académicos y Universitarios

- Bdigital, Repositorio Institucional Universidad Nacional de Colombia.⁷

- DASH (Digital Access to Scholarship at Harvard)*⁸

- SEDICI, repositorio Institucional de la UNLP⁹

- Repositorio Digital Universitario, Reposital UNAM¹⁰

- o Portales académicos

- Dialnet, universidad de la Rioja, España¹¹.

¹ <http://ieeexplore.ieee.org/Xplore/home.jsp?reload=true&>

² <http://dl.acm.org/>

³ <http://upcommons.upc.edu/handle/2099/7607>

⁴ <http://www.ribiecol.org/>

⁵ <http://cacic2015.unnoba.edu.ar/>

⁶ <http://www.tise.cl/>

⁷ <http://www.bdigital.unal.edu.co/>

⁸ <https://dash.harvard.edu/>

⁹ <http://sedici.unlp.edu.ar/>

¹⁰ <http://repositorial.cuaed.unam.mx:8080/jspui/>

¹¹ <https://dialnet.unirioja.es/>

RELPE (Red Latinoamérica Portales Educativos)¹²

Colombia aprende, Portal Educativo del Ministerio Nacional de Colombia¹³

Eduteka¹⁴

- o Búsqueda en el motor de búsqueda *Google Academics*

Los documentos indagados serán: actas de congreso, artículos de revistas, artículos en portales académicos, capítulos de libros, información directa en los sitios web de las herramientas, capítulos de tesis o reportes de investigación. Los documentos se buscarán en idioma español e inglés, publicados entre 2007 y 2017 con el objetivo que la información presentada corresponda a los últimos 10 años.

- Palabras clave

Se utilizarán las siguientes palabras clave en español, en inglés y en portugués: herramientas para el desarrollo del pensamiento computacional, enseñanza de programación, nombres de las herramientas más encontradas (por ejemplo, *Alice*, *Scratch*, *RITA*, *Phyton*, *Pygame*, *Game engine*, *Pilas Engine*, *LightBot*, etc).

Después de realizar la búsqueda teniendo en cuenta las palabras se encontraron un total de 120 documentos los cuales pasarán a un proceso preliminar de selección.

1.5.2.3. Criterios de inclusión y exclusión

Los criterios utilizados para la inclusión de artículos serán:

- Documentos que presentan herramientas que sirvan para promover el desarrollo del pensamiento computacional.
- Documentos que conceptualizan el pensamiento computacional.
- Documentos que describan las herramientas para promover el desarrollo del pensamiento computacional.

¹² <http://www.relpe.org/>

¹³ <http://aprende.colombiaaprende.edu.co/estudiantes2016>

¹⁴ <http://eduteka.icesi.edu.co/>

- Documentos que abordan propuestas para el desarrollo de pensamiento computacional.
- Documentos Publicados entre en año 2007 y 2017.

Los criterios utilizados para la exclusión de artículos fueron:

- Artículos a los que no se pudo acceder al texto completo.
- Documentos cuyo idioma no fuera español, inglés o portugués.
- Literatura informal.

1.5.2.4. Proceso preliminar

En el proceso preliminar se propone confeccionar una lista con los documentos que cumplan con los criterios de inclusión y exclusión ya explicitados. A partir de ellos, se tomarán referencias y bibliografía que podrían servir para este estudio, así como también otros escritos relacionados con la temática de interés de los autores que aparecen en las referencias.

1.5.2.5. Proceso de selección final

Para finalizar el proceso se aplicarán nuevamente los criterios de inclusión y exclusión para descartar aquellos documentos que no cumplieran con los objetivos de la búsqueda.

Por último, se generará una recopilación con el total de documentos que se utilizarán para el desarrollo del presente trabajo.

Después de realizar el proceso final de selección se generó una lista de 48 documentos los cuales sirven como base teórica en la realización de esta investigación.

Capítulo 2

Revisión de las estrategias metodológicas en las iniciativas revisadas

2.1. Introducción

En las iniciativas para el desarrollo de las competencias relacionadas con el pensamiento computacional que se encontraron alrededor de Iberoamérica, se utilizaron diversas estrategias metodológicas para la enseñanza de la programación o temas afines. Retomando las iniciativas analizadas en el Trabajo Final Integrador TFI¹⁵ y algunas encontradas posteriormente, se revisan y se eligen aquellas que abordaron la mayor cantidad de competencias que en ese documento se proponen como habilidades, conocimientos y actitudes importantes para la resolución de problemas por medio de recursos informáticos.

Las competencias referidas son:

- Formular problemas posibles de solución mediante una computadora.
- Abstractar la información relevante para reconocer los datos principales.
- Diseñar algoritmos para llegar de manera ordenada a la solución.
- Descomponer datos, procesos (modularizar) o problemas en partes pequeñas y manejables.
- Simular modelos de las posibles soluciones antes de ponerlos a prueba.
- Implementar los modelos para saber si la solución propuesta es eficiente y efectiva.
- Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos.
- Persistir en el trabajo con problemas difíciles.
- Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

¹⁵ Trabajo Final Integrador, Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica. Sarmiento, Gorga, y Sanz (2016).

2.2. Estrategias metodológicas utilizadas en las iniciativas revisadas

JET: Java en Escuelas Técnicas – Programando con RITA

La iniciativa de la Facultad de Informática de la Universidad Nacional de La Plata, hace uso de la herramienta didáctica-pedagógica RITA (*Robot Inventor to Teach Algorithms*) como medio de integración universidad – escuela, destinada a alumnos de escuelas secundarias de la provincia de Buenos Aires (Argentina), en esta propuesta participaron estudiantes de diferentes ciclos (formación básica y especialización), que fueron seleccionados por docentes de las escuelas cuyos perfiles tenían características y edades diferentes. Los estudiantes de esta iniciativa, alrededor de 116, no contaban con conocimientos previos de JAVA, no habían usado RITA ni tampoco habían programado con herramientas basadas en bloques.

La iniciativa como experiencia piloto inicia en el año 2012, permitiendo no sólo la transferencia sino también la apropiación de contenidos por parte de los alumnos, enmarcada dentro de la política de inclusión y divulgación promovida por la Facultad de Informática de la UNLP, específicamente desde la Secretaria de Extensión, continúa aun siendo implementada, con el objetivo de promover las carreras de grado.

La estrategia metodológica que se utilizó para el año 2014 se viene implementando desde el año 2012 con algunas modificaciones, según se menciona en el artículo de Queiruga, Fava, Gómez, Kimura, y Bartneche (2014), fue la siguiente:

- Explicación introductoria de *Robocode* y uso de RITA con un desafío individual guiado. Duración: 50 minutos.
- Desafío grupal, donde cada grupo (2-3 alumnos) creó la estrategia de su robot. Los alumnos no fueron guiados. Los instructores contestaban las preguntas de los alumnos. Duración: 50 minutos.
- Actividad final: puesta en combate de los robots generados por los alumnos.

Para el 2015 los alumnos participantes de este taller tenían entre 17 y 18 años y estaban cursando 6 y/o 7 año de escuela secundaria, no todos provenían de escuelas con orientación en informática y se realizaron 3 encuentros de 2 horas cada uno, continuando con la metodología de trabajo de taller anteriormente mencionada.

La información que se encontró con respecto a la evaluación de la propuesta únicamente hace referencia a 100 encuestas realizadas a los estudiantes participantes de la experiencia en el año 2012, de las cuales se observó que:

- Los alumnos se sintieron capaces de empezar a usar la herramienta con una explicación básica.
- Los alumnos pudieron construir un robot que se puso a prueba frente a sus compañeros.
- Los alumnos prefieren usar bloques frente a un lenguaje de programación.
- Los alumnos encontraban correspondencia entre la estructura de bloques que ellos crearon y el código Java generado automáticamente por RITA.

Y cuyas conclusiones obtenidas fueron las siguientes:

- En 100 minutos el 93% de los alumnos encuestados, sin experiencia previa con RITA pudieron construir un robot con una estrategia.
- El 91% de alumnos encuestados, encuentran que programar por bloques les resultaba más fácil que hacerlo con un lenguaje de programación ya que se evitaban entender una sintaxis que en general les resulta difícil.
- El 73% de los estudiantes pudieron en un encuentro de 2 horas programar un robot capaz de combatir en un campo de batalla.
- El 76.5% entendieron el código JAVA a partir de los bloques, aun cuando nunca habían visto código JAVA antes.

Con respecto a las implementaciones de los últimos años no se encontró ningún tipo de información en la manera en que se evaluó la experiencia, la anterior información se encuentra en el documento Enseñando a programar con

RITA en escuelas secundarias (Rosales, Queiruga, Kimura, Barnetche, y Gómez, 2015).

En los documentos consultados se indica que la iniciativa favorece el aprendizaje de la programación por medio de la cual estimula la resolución de problemas, la competencia y el trabajo colaborativo, de igual manera promueve la abstracción y modularización de información.

Programando con robots

Fue una iniciativa basada en un proyecto del *Georgia Tech Institute* y promovida por la Facultad de Informática de la Universidad Nacional de La Plata, la cual en un principio utilizó robots *scribblers* y tradujo libros de referencia junto con guías y prácticas propuestas por el instituto para la iniciativa original. En un segundo momento los robots *scribblers* fueron sustituidos por los Múltiplo N6 creación de la Facultad de Informática de la UNLP y la empresa *RobotGroup*. Ambas clases de robots se programaron en *Python* con un API desarrollada para este proyecto.

En esta iniciativa se promovieron dos modalidades de prácticas, en una se trabajó directamente en las escuelas donde se realizaron actividades sociales y lúdicas, motivando a que los participantes diseñen carreras, bailes, batallas, obras de teatro, etc. Al terminar esta instancia se realizaron actividades de mayor complejidad, donde se involucraron conocimientos más avanzados del lenguaje para la realización de actividades interdisciplinarias en colaboración con los docentes.

La otra modalidad se trabajó a través de pasantías académicas donde algunos estudiantes de último año de las escuelas, asistieron a la Facultad para realizar diversas actividades utilizando los robots programados en *Python*, estas pasantías se realizaron en un periodo de 4 a 6 semanas con un encuentro semanal de aproximadamente una hora.

Se describen en la Tabla 2.1 algunos aspectos metodológicos que se realizaron durante las sesiones.

Tabla 2.1 sesiones de trabajo propuesta programando con robots. Tomada de los autores de referencia

Sesión	Tema
Clase 1	El proyecto: Entre Robots, juegos y lenguajes: una experiencia de aprendizaje real en las aulas Conceptos básicos: Introducción al Software Libre - Licencias <i>Creative Commons</i> GNU/Linux Los Robots Introducción a <i>Python</i> Guía práctica
Clase 2	Operadores y expresiones Variables y Tipos Funciones Las Estructuras de Control Guía práctica
Clase 3	Repaso estructuras básicas de <i>Python</i> Sensores del Robot Guía práctica
Clase 4	Tipos de datos. Trabajando con Cadena de caracteres. Manipulación de archivos. Guía práctica

Las modalidades implementadas fueron guiadas por capacitadores que en su mayoría eran estudiantes voluntarios de la Facultad, quienes dictaron talleres teórico prácticos de *Python* para programar los robots. Algunos de estos productos fueron presentados en eventos científico-tecnológicos como congresos o jornadas de software libre, hecho que despertó mayor interés y satisfacción entre los participantes (Tzancoff y Queiruga, 2014; Díaz, Banchoff, Martin, y López, 2012).

Los documentos encontrados no mencionan la manera en que se evaluó la iniciativa, por este motivo se buscó una docente involucrada en el proceso, mediante correo electrónico se contactó a Claudia Banchoff quién por medio de una charla virtual comentó que la evaluación de la experiencia se realizó mediante encuestas informales para reconocer los conceptos de programación aprendidos por los estudiantes, también se indagó por medio de observación directa la actitud con la que los participantes programaban los robots. Según esta evaluación los estudiantes aprendieron conceptos básicos de programación que utilizaron para hacer que los robots realizaran algunas tareas. Este trabajo motivó a los estudiantes, pues pudieron visualizar de manera física los resultados del código escrito. Esta experiencia favorece el desarrollo de habilidades como el

pensamiento analítico o la resolución de problemas mediante el uso de algoritmos y el trabajo grupal.

Program.AR

Es un proyecto de acompañamiento a escuelas que se inició en el 2013 promovida por la Fundación Sadosky del Ministerio de Ciencia, Tecnología e Innovación Productiva de la Presidencia de la Nación en Argentina. En la cual se inicia un trabajo con 20 universidades en el 2012, formando estudiantes de estos planteles para ser divulgadores del programa. Los estudiantes formados brindaron desde ese momento talleres a estudiantes de los últimos años de secundaria para promover carreras informáticas. Hasta el momento el programa tiene un alcance de 20.000 estudiantes.

Continuando con el acompañamiento en el año 2014 se realizó un foro nacional en el cual se trabajó en talleres, debates y *hackatones* contando con más de 2500 asistentes.

De este proyecto, interesa rescatar que en el 2015, en conjunto con la Universidad de Quilmes y la Universidad de Córdoba, se realizó un curso llamado “La programación y su didáctica”, en el que participaron 500 docentes.

En ese marco se desarrollaron manuales para docentes de primaria y secundaria, y también se proporcionaron cursos virtuales y herramientas como: *Scratch, Pilas Bloques, LightBot, Alice, Gobstones, Mumuki, Python, Pygame, Pilas Engine, Ruby, Pharo Smalltalk, PSeInt* y *Haskell* para el aprendizaje de la programación, puestos a disposición mediante su plataforma <http://program.ar/>.

Al revisar el sitio web y los documentos propios de este programa, no se encuentran resultados de la evaluación realizada a este proyecto, únicamente se dan a conocer indicadores tal como se muestran en la Figura 2.1.



Figura 2.1 - Alcances proyecto Program.ar. Tomada del sitio web del proyecto

Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación

Es una iniciativa del Grupo de Investigación y Desarrollo en Innovación Educativa, Departamento de Matemática, Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa, Argentina, dirigido a los ingresantes de las carreras Profesorado en Computación, Profesorado en Matemática y Licenciatura en Matemática. Tiene entre sus objetivos abordar conceptos iniciales de la programación de computadoras, utilizando un lenguaje de programación de alto nivel. En esta propuesta se tratan temas como: la noción

de algoritmos, variables y tipos de datos (simples y estructurados homogéneos), condiciones, estructuras de control y resolución de problemas con computadoras.

La iniciativa es un Taller de Introducción a la Programación (TIP) de carácter optativo con momentos virtuales y presenciales que propone utilizar juegos serios, con un enfoque lúdico y de experimentación, trabajándolos en cuatro niveles: un primer nivel *Lightbot 2.0*, segundo nivel *Code.org*, y tercer y cuarto nivel *Scratch*.

Se utiliza *Moodle* como medio de interacción en el curso, donde se presentan las actividades y los recursos a utilizar. Una vez que el participante completa un nivel obtiene una calificación entre 30 y 100 por ciento en la evaluación propuesta.

En el nivel 1 se utilizó *Lightbot* para construir nociones de algoritmos e instrucciones, por medio de actividades como:

- Foro de presentación y Encuesta inicial.
- Video de presentación de *Lightbot*.
- Realización del primer nivel en *Lightbot*
- Realización de un documento presentando una equivalencia entre instrucciones icónicas de *Lightbot* y su forma textual, como medio para propiciar la transferencia de los aprendizajes.
- Evaluación y paso de nivel usando desafíos con escenarios de *Lightbot*.

En el nivel 2, los estudiantes juegan en *Code.org* realizando actividades de “La hora de código”. Por ejemplo, utilizan el Laberinto clásico, puesto que la metáfora utilizada es similar a *Lightbot*. Los conocimientos adquiridos actúan como base para trabajar en este nuevo nivel, que presenta inicialmente problemas que luego se tornan como tareas rutinarias. A partir del trabajo en sus 20 niveles se abordan conceptos de secuencia, selección y repetición. Este nivel 2 incluye actividades como:

- Video explicativo de *Code.org*.
- Realización de actividades.
- Publicación del certificado que se otorga por la participación en *Code.org*.
- Realización de la actividad evaluativa que permite pasar el nivel.

En el Nivel 3 y 4, se trabaja la noción de variable, estructuras de control y resolución de problemas por medio de *Scratch*, donde los problemas o desafíos se trabajan como actividades dirigidas en los talleres presenciales.

El tercer nivel propone realizar un juego en el que se programa a un mono, por medio de una secuencia de instrucciones para que logre comer las bananas ubicadas en un damero. Para cumplir con este objetivo se cuenta con botones para: agregar las instrucciones (arriba, abajo, derecha e izquierda), borrar todo y ejecutar las instrucciones.

El nivel 3 cuenta con las siguientes actividades y recursos:

- *Polya vs Desarrollo de sistemas*. Video donde se explican y comparan los cuatro pasos básicos del desarrollo de software con los cuatro pasos propuestos por George Polya¹⁶ para la resolución de problemas.
- El gato y el damero. Se presenta el problema de mover un gato (objeto) a través de un damero (escenario) con las flechas del cursor. Este problema involucra el uso de escenarios, eventos e instrucciones. El problema se resuelve en clase con la guía del docente y tiene como objetivo mostrar estrategias y heurísticas para resolver problemas, discutiendo las posibles soluciones y mostrando el razonamiento experto.
- Trabajo práctico 1. Se proponen tres problemas sencillos para resolver con *Scratch*, aplicando lo visto en clase.
- Se presenta el problema del mono come bananas. Este problema involucra el uso de varios objetos (bananas), instrucciones del bloque control (selección y repetición) y se introducen las variables y operadores para ubicar las bananas al azar y para calcular el puntaje. El problema se resuelve en clase con la guía del docente. Los estudiantes deben publicar una solución a la que se le agrega un desafío adicional (problema): un cuadro rojo en el damero que, si es tocado por el mono, éste vuelve al inicio y se le descuentan puntos. Se busca que al realizar esta actividad

¹⁶ George Polya, matemático húngaro, quien diseñó un método para resolver problema de programación en cuatro pasos, Paso 1: Entender el problema, Paso 2: Diseñar el programa, Paso 3: escribir el programa, Paso 4: Examinar la solución obtenida.

puedan transferir lo aprendido, ya que la solución es similar a lo trabajado en clase. La entrega es evaluada por un docente, quien le hace una devolución (puede volver a entregar las veces que sea necesario). Si el programa es correcto acceden al próximo nivel.

- Trabajo práctico 2. Se proponen dos problemas sencillos para resolver con *Scratch*.

El nivel 4 incluye las siguientes actividades y recursos:

- El mono come banana (versión 2). Se agregan, al problema ya planteado, nuevos desafíos. Al finalizar la clase se socializan las soluciones. Los estudiantes deben publicar su solución, un docente las evalúa y hace una devolución. Deben aprobar la tarea de programación para acceder a la próxima actividad del nivel.
- Trabajo práctico 3. Se propone a los estudiantes la programación con *Scratch* de un juego sencillo.
- El mono come banana (versión 3). Aquí se retoma el planteo inicial del problema. Se incorpora el manejo de variables estructuradas (listas) para almacenar las instrucciones.
- Trabajo práctico 4. Se propone la realización de un juego similar al *PAC-MAN*.

Para evaluar esta iniciativa, al terminar el taller, los participantes contestaron una evaluación final donde se indaga sobre los conocimientos abordados en el taller. Esta actividad solamente fue realizada por los estudiantes que completaron exitosamente, al menos, el nivel 3.

Durante 2015 se observaron buenos resultados en la evaluación final, que fue realizada por 25 de los 57 participantes, y de ellos el 70% tuvieron un promedio por encima de 7 en la calificación final. De esta manera se observa que hay un dominio aceptable de los conceptos. Debe tenerse en cuenta que de los 57 inscriptos, 44 contestaron una encuesta inicial y de ellos el 80% no habían utilizado nunca un lenguaje de programación. El 79% de los estudiantes que aprobaron el primer parcial habían completado las actividades del Taller.

En 2016, ingresaron 61 estudiantes para realizar el taller, de ellos 11 no accedieron nunca y 4 accedieron, pero no hicieron ninguna actividad, por esto quedaron un total de 46 participantes, de los cuales 45 completaron el Nivel 1, 38 el Nivel 2, 28 el Nivel 3 y 13 el Nivel 4. Los 28 que completaron el Nivel 3 tenían acceso a la evaluación final, de ellos, sólo 17 la completaron. En promedio, quienes realizaron la evaluación, obtuvieron una calificación de nueve.

Como en el año anterior, del total de participantes, el 80% (37 de los 46) indicaron que no conocían ningún lenguaje de programación.

Esta iniciativa promueve el aprendizaje básico de la programación introduciendo los conceptos de algoritmo, variable, estructuras de control y resolución de problemas (Astudillo, Bast, y Willging, 2016).

Herramientas lúdicas como apoyo a la enseñanza de la programación

La Universidad Nacional del Nordeste en Argentina propone una serie de actividades basadas en las herramientas *Lighthbot* y *Scratch*, con el objetivo de incorporar conceptos básicos de programación mediante herramientas lúdicas que resulten atractivas para los alumnos ingresantes a la carrera Licenciatura en Sistemas de Información.

La iniciativa propone el inicio anticipado de actividades académicas para alumnos ingresantes, realizando un taller de aprestamiento durante tres semanas, con 2 sesiones semanales de 2 horas, en donde se trabajaron los conceptos de “abstracción”, pensando en estrategias antes de intentar la resolución de los ejercicios. También se abordó la “descomposición del problema en partes”, para la creación de bloques o procedimientos en la solución diseñada y en la “legibilidad” de la solución, definiendo nombres representativos de los procedimientos creados.

Para evaluar las actividades desarrolladas en el periodo de aprestamiento, se planteó resolver un problema en el cual se requería utilizar los conceptos de secuencia, alternativas y repetición condicional, a partir del trabajo con la herramienta *Scratch*.

Del mismo modo se buscó por medio de *Scratch* trabajar contenidos específicos, con técnicas similares que luego pudieran ser transferidas en el

momento de hacer uso de C, lenguaje utilizado en la asignatura Introducción a la Programación Estructurada. En esta asignatura se trabaja como contenido las estructuras de control, funciones, parámetros y recorrido en arreglos.

Luego de terminar el curso, se realizó una encuesta a los participantes con el propósito de determinar si el conocimiento adquirido en el periodo de aprestamiento resultó útil para una mejor comprensión de los conceptos. Esta encuesta se realizó de manera online utilizando un formulario en *Google Docs*, que arrojó los resultados que se detallan a continuación.

Respecto a la apreciación sobre las herramientas utilizadas, *Ligthbot* y *Scratch*, se observa una leve preferencia hacia la última, de la cual el 66% piensa que es lúdica, didáctica y entretenida. En ambos casos, sólo el 5% considera que es compleja.

En cuanto a la dificultad para el aprendizaje y uso de las herramientas, si bien *Ligthbot* resulta levemente más fácil de aprender (61%), *Scratch* no representa mayor dificultad.

De las respuestas en relación a las herramientas, se observa en general una alta aceptación hacia las mismas, y en ambos casos resulta difícil sólo para un 7% para *Ligthbot* y un 4% para *Scratch*.

También se preguntó a los alumnos en qué grado las actividades realizadas durante el aprestamiento contribuyeron a incorporar aspectos claves que conforman el método para diseñar soluciones, tales como, abstracción, modularización y legibilidad. Un 68% opina que contribuyeron significativamente a comprender el concepto de abstracción, un 75% a la modularización y 71% a la legibilidad.

De igual manera se preguntó a los alumnos en qué grado las actividades de aprestamiento lo ayudaron a entender los conceptos básicos relacionados con la programación estructurada: secuencia, alternativa (Selección), iteración (Repetición), modularización (Uso de funciones), parámetros (Generalización). En la mayoría de los casos, los alumnos valoraron las actividades de aprestamiento realizadas para mejorar la comprensión de los mencionados conceptos. En un alto grado en el caso de la repetición (91%), y en menor grado en el tema

parametrización, en el que el 50% de los alumnos manifestó que le resultó poco útil. Esto puede deberse al escaso tiempo dedicado a este tema, para la complejidad del mismo.

Finalmente, se preguntó a los alumnos si consideraban que las actividades de aprestamiento sirvieron para ayudar a entender y resolver los prácticos de la asignatura. El 70% respondió que le hubiera costado más, y el resto señaló que hubiera llegado al mismo resultado.

Según la evaluación, esta iniciativa promueve la comprensión de los conceptos propios de la programación en el nivel introductorio universitario, facilitando la incorporación de los conceptos y técnicas formales del área Algoritmos y Estructuras de Datos (Dapozo, Greiner, y Petris, 2016).

Programa Nacional de Informática Educativa (PRONIE)

El Ministerio de Educación Pública y la Fundación Omar Dengo promueven el Programa Nacional de informática Educativa (PRONIE) en Costa Rica. Este proyecto incorpora las tecnologías digitales como herramienta para apoyar el aprendizaje curricular, desarrollando capacidades intelectuales mediante el trabajo por proyectos y en la construcción de productos digitales. Busca fomentar destrezas para el uso de herramientas, programación de computadoras y la arquitectura de redes.

Los estudiantes se involucran activamente en el desarrollo de proyectos que simulan eventos y procesos, al mismo tiempo que estudian los conceptos y principios básicos de programación que les permite crear y poner a funcionar las simulaciones, por medio de programas como: *Construct2*, *Scratch*, *App Inventor*, *Alice*.

Según la información que se muestra en los documentos donde se describe la propuesta, esta iniciativa favorece por medio del aprendizaje por proyectos (Zúñiga y Brenes, 2014):

- El desarrollo del pensamiento lógico matemático
- El desarrollo de habilidades para la resolución de problemas
- El manejo del error como oportunidad de aprendizaje

- La profundización y ampliación de temáticas curriculares
- El desarrollo de la creatividad y expresividad
- El incremento de la autoestima
- La exploración de ambientes tecnológicos
- El aprendizaje en entornos colaborativos

Puesto que este proyecto ejecuta diversas propuestas enfocadas al aprendizaje de la programación, su estrategia metodológica es variada. A continuación, se hace referencia a algunas de las actividades realizadas, ellas son:

a) Actividades *BOOM APPS*: pretende desarrollar competencias digitales y habilidades en programación de aplicaciones para dispositivos móviles. Propone un proceso virtual como preparación para la participación en campamentos tecnológicos donde se reúnen alrededor de 240 estudiantes. Estos campamentos se llevan a cabo durante las vacaciones, también se realiza un encuentro de programación para jóvenes, que se complementa con actividades virtuales (Schroeder, 2014). El trabajo desarrollado se basa en el uso del método de 7 pasos para la resolución de problemas en el diseño de aplicaciones para dispositivos móviles, utilizando el *software Construct2* tratando temas como:

- Manejo de variables;
- Uso de condicionales;
- Creación de animaciones;
- Uso de comportamientos de los objetos;
- Uso de herramientas de multimedia como audio;
- Desplazamientos entre pantallas; y
- Creación de tablas de histórico de datos.

b) Actividades con La hora del código, como medio para la enseñanza de conceptos básicos de programación. PRONIE se suma a esta iniciativa global, de modo que los estudiantes participen de esta actividad que promueve la introducción a las Ciencias de la Computación.

- c) Actividades en Revista Digital, cuenta con una plataforma virtual donde los estudiantes publican reportajes y artículos de opinión haciendo uso de *Scratch*. Los artículos que se quieran publicar son revisados por el Consejo Editorial Estudiantil que se compone de estudiantes pertenecientes a cada uno de los centros educativos participantes de esta actividad.
- d) Actividades de Robótica y aprendizaje por diseño. Se trata de un taller para secundaria llamado “Taller de solución creativa con robótica”, cuyo propósito es involucrar a los jóvenes en proyectos enfocados a la detección, evaluación y solución de problemas de sus comunidades, por medio del diseño, la construcción y programación de prototipos utilizando tecnologías digitales. Los productos se dan a conocer a la comunidad por medio de encuentros organizados en cada una de las instituciones educativas participantes. También se publican los procesos por medio de un blog enlazado al sitio web del proyecto <http://www.fod.ac.cr/robotblogs/intituciones.htm>.

En una estadística encontrada en el sitio web del PRONIE se muestra la cobertura del programa en el año 2016 (ver Figura 2.2).

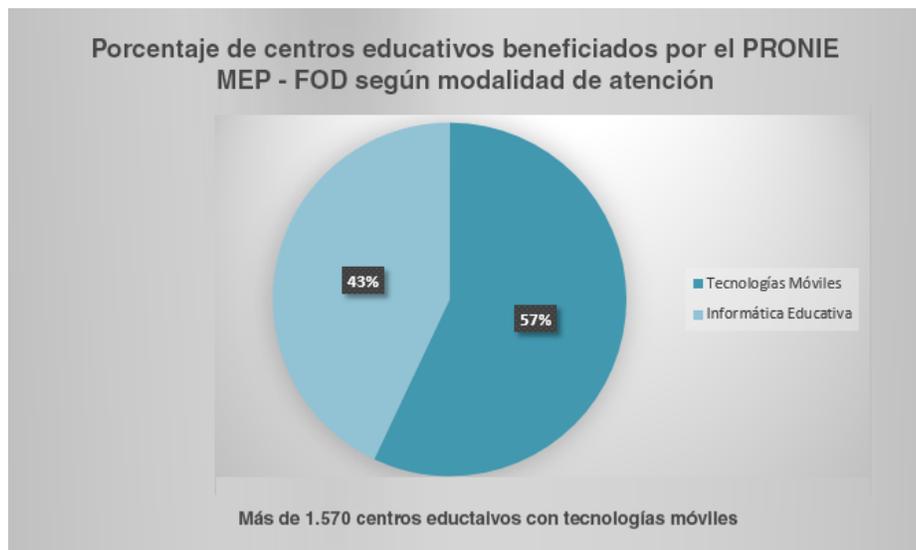


Figura 2.2 - Cobertura PRONIE 2016 tomado de los autores de referencia

Con respecto a la evaluación de este proyecto, no se encuentra información al respecto, pero los procesos de enseñanza y sus actividades se continúan implementando hasta el momento. Esto lleva a pensar que el proyecto ha arrojado resultados favorables al promover y abordar en su totalidad las competencias relacionadas con el pensamiento computacional, por hacer uso de diversas herramientas y propuestas metodológicas para la enseñanza de la programación y temas afines.

Desarrollo del pensamiento computacional con *Scratch*

Es una iniciativa promovida por la Universidad Santiago de Chile, para la cual se diseñaron 11 sesiones (ver Tabla 2.2) con una duración aproximada de hora y media a dos horas, cada sesión tiene un objetivo de lo que debe aprender el estudiante.

La aplicación se realizó en dos grupos, un primer grupo formado por estudiantes de segundo ciclo de enseñanza básica, 9 estudiantes de séptimo y 10 de octavo grado. En el segundo grupo los participantes fueron estudiantes de la Carrera Licenciatura en Ciencia de la Computación, en la asignatura de Computación I. Las sesiones se describen en la tabla 2.2.

Tabla 2.2 Descripción de las sesiones de trabajo, tomada de Capot y Espinoza, 2015.

N	Título	Objetivo	Pensamiento Computacional
1	Conociendo <i>Scratch</i>	Conocer entorno de trabajo y características del software, por medio de juegos y ejemplos en <i>Scratch</i>	Recopilación de Datos
2	Uso de variables simples para generación de animaciones.	Identificar tipos de variables simples que permiten movimientos, en soluciones a problemas simples que se puedan representar computacionalmente.	Recopilación de Datos, Análisis de Datos y Representación de Datos.
3	Creación de animaciones de <i>Scratch</i> en la generación de simulaciones de situaciones de la vida diaria	Realizar simulaciones de situaciones de la vida diaria.	Recopilación de Datos, Análisis de Datos y Simulación

N	Título	Objetivo	Pensamiento computacional
4	Programar en <i>Scratch</i> soluciones a problemas simples matemáticos	Realizar algoritmos y procedimientos utilizando procesos lógicos	Recopilación de Datos, Análisis de Datos, Algoritmos y Procedimientos y Automatización de Procesos
5	Crear una animación basada en una historia, generando así una secuencia lógica.	Organizar información para dividir una tarea en partes pequeñas y manejables para generar una secuencia lógica que resuelva el problema.	Recopilación de Datos, Análisis de Datos, Descomposición de Problemas, Algoritmos y Procedimientos
6	Crear una solución para representar el desplazamiento de una persona.	Reunir y analizar la información para generar desplazamiento de persona y/o objetos.	Recopilación de Datos y Análisis de Datos, Realización de Algoritmos y Procedimientos, Simulaciones de Situaciones
7	Crear juegos usando <i>Scratch</i> interacciones entre distintos elementos.	Realizar un juego en <i>Scratch</i> que integre los elementos vistos en las sesiones anteriores	Recopilación y Análisis de Datos, Descomposición de Problemas.
8	Crear un laberinto interactivo en Scratch	Crear juegos que contemplen en su realización la capacidad de descomposición de problemas, realización de algoritmos y procedimientos.	Descomposición de Problemas, Realización de Algoritmos y Procedimientos
9, 10, 11	Crear un juego en Scratch a elección del estudiante en el que se aplique todo lo enseñado en el curso.	Crear juegos que contemplen en su realización la capacidad de recopilación y análisis de información, descomposición de problemas, realización de algoritmos y procedimientos y automatizar procesos.	Recopilación y análisis de datos, descomposición de problemas, realización de algoritmos y procedimientos y automatizar procesos.

La evaluación de esta iniciativa fue realizada mediante observación directa. En la tabla 2.3 se muestran algunos resultados.

Tabla 2.3 Resultados. Tomada de autores de referencia

Tipo de estudiante	Resultados
Segundo Ciclo de Enseñanza Básica	<ul style="list-style-type: none"> • 100% logra la realización de las actividades. • 20% logra automatizar los procesos. Aunque todos realizan la última actividad, sólo un grupo minoritario incorpora elementos nuevos y realizan interacciones nuevas a la vista durante las sesiones de trabajo.

Tipo de estudiante	Resultados
Primer nivel carrera Licenciatura en Ciencia de la Computación	<ul style="list-style-type: none"> Las primeras actividades se realizan en promedio de una hora, demorando menos del tiempo esperado, por lo que se realizan varias actividades en una sesión de trabajo. 100% logra la realización de las actividades. 85% logra automatizar los procesos. La gran mayoría de los estudiantes incorpora elementos nuevos, interacciones no vistas y complejidad superior a la esperada.

Según esta observación los estudiantes desarrollaron habilidades de orden superior relacionadas con el pensamiento computacional, fortalecieron procesos lógicos que permitieron la modelación correcta para la solución de problemas y fomentaron la creatividad (Capot y Espinoza, 2015).

Uso de *Scratch* y *Legó Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación

En la Universidad de Valparaíso se desarrolló esta iniciativa para la enseñanza de conceptos básicos en el curso de Fundamentos de Programación. Las actividades se organizaron en 12 sesiones presenciales con duración de hora y media, donde se trabajó la construcción de juegos como medio para promover la reflexión y estimulación del aprendizaje. Se hizo uso de entornos interactivos del estilo *drag and drop*, como *Scratch* y el esquema de Usar- Modificar-Crear, en el cual el estudiante explora juegos ya creados modificando su funcionamiento y apariencia (ver Tabla 2.4). Cuando el estudiante comprende el funcionamiento empieza a crear sus propios programas, de esta manera se van explorando progresivamente conceptos que se utilizan en la creación de nuevos juegos.

Tabla 2.4 Actividades taller con *Scratch*. Tomada de autores de referencia

Sesión	Actividades / contenido
1	Familiarización con el ambiente <i>Scratch</i> (conceptos <i>sprite</i> y colisión entre <i>sprites</i>)
2	Variables y estructuras repetitivas
3	Estructuras repetitivas y estructuras condicionales
4	Crear juego Piedra-Papel-Tijera
5-6	Crear el juego Simulación de Guerra
7-8	Crear el juego <i>Breakout</i>
9	<i>Pacman</i> – Crear la mecánica básica de los movimientos de los personajes
10-11	<i>Pacman</i> – Implementar las demás características del juego final
12	Presentación del proyecto Final (<i>Aero Fighters</i>)

En una instancia más avanzada se utilizaron los *Robots Lego Mindstorms* cuyos conceptos de programación adquiridos sirvieron para el diseño y construcción de robots, que fueron abordados por medio de 5 unidades compuestas por un conjunto de tareas cuyo tiempo de desarrollo aproximado fue de 90 minutos. En la Tabla 2.5, se describen las Unidades del taller.

Tabla 2.5 Unidades del taller y sus actividades asociadas. Tomada de autores de referencia

Unidad	Actividad
Unidad 1 Introducción al taller de Robótica	1: Roles en los equipos, introducción a los kits de <i>Lego Mindstorms NXT</i> y su lenguaje de programación (NXC). 2: <i>Lego Digital Designer</i> para la construcción de los robots <i>Lego Mindstorms NXT</i> .
Unidad 2 Servomotores	3: Tipos de datos, constantes, estructuras de control (selectivo y repetitivo), funciones básicas de los servomotores. 4: Programación de movimientos del robot, funciones avanzadas de los servomotores. 5: Ejercicios
Unidad 3 Sensores de luz	6: Funciones básicas y aplicaciones de los sensores de luz. 7: Algoritmos básicos para trabajar con el robot seguidor de luz. 8: Algoritmos básicos para trabajar con el robot seguidor de luz. 9: Ejercicios
Unidad 4 Sensores de Ultrasonido	10: Funciones básicas y aplicaciones del sensor de ultrasonido. 11: Programación del robot explorador (detecta y esquiva obstáculos). 12: Programación del robot explorador 13: Ejercicios y Evaluación
Unidad 5 Sensores táctiles y de sonido	14: Instrucciones básicas y sus aplicaciones en sensores táctiles y de sonido 15: Desarrollo del proyecto final 16: Desarrollo del proyecto final 17: Ejercicios y evaluación

Las actividades se abordaron mediante aprendizaje basado en problemas (ABP) y trabajo colaborativo para dar solución al problema planteado, guiados por el acompañamiento de los docentes.

Esta iniciativa fue evaluada comparando el desempeño de los estudiantes en los años anteriores. Los resultados pueden observarse en la Tabla 2.6, tomada de (Muñoz et al., 2015).

Tabla 2.6 Histórico de la cátedra de Fundamentos de Programación en Universidad de Valparaíso. Tomada de autores de referencia.

Año	N	C1	C2	Porcentaje de Aprobación
2012	75	2,6	2,5	34%
2013	76	1,8	3,2	45%
2014	76	3,7	2,9	50%
N	Número de estudiantes			
C1-C2	Calificación obtenida por cada curso			

Como complemento a la comparación realizada acerca del desempeño de los estudiantes, se hizo una encuesta *online* voluntaria, en la cual debían contestar si estaban Altamente en desacuerdo (AD), Desacuerdo (D), Neutro o Indiferente (N), De acuerdo (A) o Altamente de Acuerdo (AA). Esta encuesta la contestaron 20 estudiantes. La Tabla 2.7 muestra los resultados de la encuesta.

Tabla 2.7 Resultados en la evaluación de la experiencia, tomados de (Muñoz et al., 2015).

Desarrollar los proyectos en Scratch en el curso de Fundamentos de Programación me apoyó a aprender programación				
AD	D	N	A	AA
5%	0%	20%	45%	30%
Creo que la programación con Scratch es fácil de entender				
AD	D	N	A	AA
5%	0%	0%	40%	55%
Me gustó el programar bajo el esquema <i>drag and drop</i> (arrastrar y soltar) que se utiliza en Scratch				
AD	S	N	A	AA
5%	0%	25%	35%	35%
Creo que me ayudó en mi aprendizaje el compartir y/o discutir con mis compañeros estrategias de resolución en el proyecto realizado con Scratch				
AD	D	N	A	AA
5%	5%	25%	45%	20%
Creo que personalizar mi juego en Scratch me apoyó a desarrollar distintas técnicas de programación				
AD	D	N	A	AA
5%	0%	10%	50%	35%

Esta iniciativa desarrolla habilidades para el trabajo en equipo, el reconocimiento de información relevante, el diseño de algoritmos, la simulación de modelos y la implementación de estos, de manera tangible por medio de los robots (Muñoz et al., 2015).

Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje.

Es una iniciativa de la Universidad de Valparaíso, basada en los estilos de aprendizaje de Felder y Silverman¹⁷, en donde se consideran cuatro dimensiones y cada una se extiende entre dos polos opuestos: Sensitivo/Intuitivo, Activo/Reflexivo, Visual/Verbal, Secuencial/Global.

En esta iniciativa se desarrollaron dos clases de talleres uno más avanzado que otro, para ser utilizados en dos asignaturas de la carrera de Ingeniería Civil en informática. Estos talleres tuvieron una duración de 16 semanas con sesiones de hora y media, en las cuales se utilizaron materiales como guías de actividades y video tutorial con el fin de promover los siguientes objetivos:

- Fomentar el trabajo colaborativo entre los alumnos, mediante el desempeño de distintos roles, adecuados según su estilo de aprendizaje.
- Instruir a los alumnos en la Arquitectura de Robots y uso del Kit de Robótica *Lego Mindstorms NXT*.
- Instruir a los alumnos en un lenguaje que permita la programación de robots, como lo es *NXC*.
- Generar material de apoyo para el taller que permita dar instrucción a alumnos con distintos estilos de aprendizaje.
- Mejorar el material a través de actividades de retroalimentación.

Las actividades realizadas se organizaron por medio de experiencias que se describen a continuación.

Experiencia 1: es introductoria al Taller, y presenta a los alumnos tanto el kit de robótica *Lego Mindstorms NXT*, como el lenguaje y entorno de programación.

¹⁷ “Las fuerzas y preferencias características en la forma que tienen los estudiantes para procesar información. Algunos estudiantes pueden centrarse en el manejo de datos y diferentes tipos de algoritmos, otros se sienten mejor con los modelos matemáticos y las teorías. Algunos de ellos responden fuertemente a formas visuales de información como pinturas, cuadros, diagramas y esquemas, y otros más obtienen información de forma verbal mediante escritura y las explicaciones habladas. Algunos discentes prefieren aprender activamente e interactivamente y otros funcionan mejor de manera introspectiva e individual”. R. Felder y L. Silverman. Learning and teaching styles in engineering education. Institute for the Study of Advanced Development. North Carolina State University, 1988.

Luego, en 60 minutos, los alumnos construyen el primer robot a partir de las especificaciones del tutorial de construcción del *Castor Bot*.

Experiencia 2: En esta experiencia, se presentan los conceptos básicos de programación aplicados, tales como: tipo de datos, estructuras repetitivas, selectivas, y funciones. También se presentan las funciones básicas de los servomotores. Por último, los estudiantes analizan, modifican y ejecutan los códigos de ejemplos de “Conceptos básicos de programación”, provistos como material de estudio.

Experiencia 3: en esta experiencia se explica a los estudiantes sobre el sensor de luz y sus funciones básicas. Los alumnos analizan y ejecutan los algoritmos del robot seguidor de línea, y programan aplicaciones para el robot alarma, según lo especificado en la guía de actividades.

Experiencia 4: en esta experiencia se presentan las funciones básicas del sensor de ultrasonido, y sus aplicaciones. Luego los alumnos realizan la guía de actividades, que consiste en programar un robot que debe encontrar la salida de un laberinto sencillo.

Experiencia 5: esta experiencia consiste en incorporar los sensores de tacto y de sonido. Para esto, los alumnos construyen el robot “*Tribot*”. Para esta experiencia se desarrolla un robot capaz de buscar objetos en su camino, utilizando los sensores de tacto, ultrasonido y luz.

Esta iniciativa evaluó en un primer momento la calidad del material pedagógico utilizado (el tutorial de la unidad y la guía de actividades), por medio de una encuesta en la cual los ítems tenían puntajes de 1 a 5, siendo 1 Muy malo, 2 Malo, 3 Mediano, 4 Bueno, y 5 Muy bueno.

Tabla 2.8 Medición de calidad, material de estudio. Tomada de autores de referencia.

Tutorial de la unidad	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5
Las imágenes o diagramas del tutorial son suficiente para explicar el contenido	4,5	4,7	4,6	4,8	4,7
Se explican los conceptos técnicos necesarios para el desarrollo de la experiencia	5,0	4,4	5,0	4,8	4,8
Los códigos de ejemplo ayudan a aclarar el contenido teórico	4,8	4,6	4,9	5,0	4,6

Tutorial de la unidad	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5
Los tutoriales y guías de actividades tienen una estructura adecuada	4,8	4,3	4,8	4,9	5,0
El material utilizado se adecua a las necesidades y expectativas del alumno	4,8	4,3	4,8	4,9	5,0
Promedio Experiencia	4,7	4,5	4,7	4,9	4,9

El material fue considerado de buena calidad por la calificación sobre 4,0 que dieron los estudiantes.

Como medio para evaluar el taller se realizaron encuestas de motivación con los alumnos al inicio de las sesiones y luego de transcurrida la quinta experiencia. Las encuestas fueron aplicadas a 16 estudiantes y contenían 11 preguntas que arrojaron las siguientes afirmaciones:

- El 68,4% está “totalmente de acuerdo en recomendar el taller de robótica a sus compañeros y futuros alumnos de la carrera”.
- El 57,9% está “de acuerdo con que el taller potencia el trabajo en equipo y el traspaso de conocimiento de unos a otros”.
- El 89% dice estar “de acuerdo” o “muy de acuerdo” con que “Aprender programación en el taller de robótica le servirá para entender la materia del curso de Programación”.

El trabajo realizado promueve habilidades para comprender, analizar y diseñar algoritmos, además de desarrollar competencias tales como habilidad analítica y pensamiento creativo, y la aplicación de conceptos para entender y contextualizar problemas, promoviendo el trabajo colaborativo (López, Muñoz, y Barría, 2012).

Desafío STEM – Fundación Telefónica España

Es un concurso interescolar que se realiza en el territorio español, promovido por la Fundación Telefónica, mediante un plan nacional que pretende promover las habilidades STEM (*Science, Technology, Engineering y Mathematics*). Está conformado por una estrategia de formación presencial que se desarrolla durante tres días en la semana STEM, complementado por medio de cursos y recursos virtuales que se encuentran alojados en las plataformas de aprendizaje

*ScolarTIC*¹⁸ y *STEMbyme*¹⁹. Luego de recibir la formación, los estudiantes participan del concurso dando a conocer soluciones que resuelvan necesidades en alguna de las categorías del torneo: internet de las cosas, *ehealth*, industria local y educación digital.

Como parte de la estrategia de formación del Desafío STEM, se encuentran los siguientes cursos: Programando con *processing*, Crea con *AppInventor*, Construye con *Scratch for Arduino* (S4A), HTML5+CSS, Impresión 3D, Programando con *Scratch*²⁰.

Este proyecto que además de ejecutarse en el territorio español se difunde en otros países como Colombia, Chile, Brasil y Perú, en ninguno de los sitios web o documentos encontrados especifica de qué manera han evaluado las propuestas implementadas, solamente dan a conocer algunos indicadores, que según ellos avalan su liderazgo (ver Figura 2.3).



Figura 2.3 - Indicadores de liderazgo tomado del sitio web del proyecto

¹⁸ www.scolartic.com

¹⁹ www.stembyme.com

²⁰ <http://www.desafioSTEM.com/web/guest/inicio>

El uso de *Scratch* en el desarrollo de la programación lógica como un aporte interdisciplinario

Esta iniciativa propone una estrategia metodológica para ser usada en el primer año académico en las clases de computación de las Escuelas Públicas Integradas. Se trata de un programa dirigido por Universidade Federal do Espírito Santo en Brasil, cuya metodología se trabajó en 3 etapas que fueron las siguientes:

Tabla 2.9 Metodología de trabajo en el proyecto, Tomada de autores de referencia.

Etapas	Actividades
Primera etapa	Aprendizaje y exploración de recursos <i>Scratch</i> , en el salón de clase se exploraron las potencialidades del <i>software Scratch</i> , junto con las estrategias para la creación de los juegos, formación de los grupos de trabajo y definición de objetivos y metas por alcanzar.
Segunda etapa	Se desarrollaron los juegos aplicando lógica de programación, diseñando los escenarios y los objetos utilizados para la creación de actividades con temáticas multidisciplinares.
Tercera etapa	Finalización del desarrollo de los juegos, presentación y exposición de los mismos en la feria tecnológica de la escuela.

En la primera etapa se revisaron conceptos en el siguiente orden:

- Comando de la estructura de repetición simple.
- Variables, expresiones lógicas, estructuras condicionales.
- Controles con variables acumulativas.

Para la segunda etapa se les pidió a los estudiantes que crearan juegos relacionados con temáticas multidisciplinares en los cuales se trataran temas de matemáticas, geografía, portugués, historia, física, química, biología, artes, inglés o español.

En la tercera etapa los estudiantes dieron a conocer los productos realizados.

La evaluación de esta propuesta se hace por medio de la observación directa la cual permitió comprobar el desarrollo de habilidades en la organización de la información, utilización efectiva de los recursos asignados y operaciones cognitivas inherentes a la lógica de programación.

Según Ríos y Cury (2016) la realización de juegos interdisciplinarios contribuyó a:

- Relectura y memorización de contenidos de otras materias.
- Promover e integrar grupos de trabajo.
- Estimular la resolución de conflictos y toma de decisiones.
- Poner en juego la creatividad y el desarrollo de ideas.
- Superar retos y frustraciones encontradas en el proceso de diseño y resolución de problemas.
- Desarrollar el pensamiento creativo para la búsqueda de soluciones innovadoras.
- Generar nuevas soluciones al aparecer nuevos desafíos.
- Persistir y practicar.
- Desarrollar pensamiento crítico.
- Aprender a pensar, elegir, crear múltiples medios de información tales como texto, imágenes, animaciones o audio.
- Desarrollar habilidades para comunicar y dar a conocer los resultados.

Trabajar con la robótica educativa en la escuela primaria

Es una iniciativa trabajada en la escuela don Francesco Luppino. SESI, en conjunto con el Instituto Federal de Educación, Ciencia y tecnología de Amazonas IFAM en Brasil, para los estudiantes de primaria en el componente curricular de "Educación Tecnológica" y "proyectos tecnológicos". Está destinada a niños de 10 a 13 años que asisten a grados entre el 5 y el 7 año. El trabajo se realizó con un total de 69 estudiantes, con el fin de promover la enseñanza de robótica de forma lúdica.

El trabajo se realiza mediante conferencias en el aula y prácticas en el laboratorio, utilizando herramientas para introducir conceptos básicos de razonamiento lógico y nociones de programación. Un ejemplo de esto es el sitio Racha Cuca²¹ mediante el cual se desarrolla el pensamiento lógico, puesto que

²¹ <http://www.rachacuca.com.br>

contiene una interfaz de usuario amigable, que brinda instrucciones necesarias para dar solución a problemas planteados por medio de los juegos allí presentes. Se utilizan juegos de razonamiento como una herramienta para mostrar los fundamentos de la lógica.

De igual manera se abordó el concepto de algoritmo y el uso y desarrollo de diagramas de flujo, como medio de organizar y presentar las ideas, continuando la práctica con *Scratch* para crear animaciones y programas en los que se utilizan los conceptos adquiridos en clases teóricas. Uno de los objetivos es motivar el aprendizaje de los conceptos de programación, a través de una experiencia divertida. Después de las etapas de desarrollo del razonamiento lógico y algoritmos, se continuó la experiencia donde los estudiantes trabajaron con el kit de programación y construcción de bloques de *LEGO MINDSTORMS EV3*. Se plantearon diferentes retos con el uso de una interfaz de programación, que permitió codificar los movimientos y acciones de un robot, a partir del desarrollo de programas simples por medio de bloques que se arrastran y se sueltan, para conectar los componentes más básicos, tales como bucles, tiempos de espera, la conmutación e incluso los componentes más avanzados. Esto con el fin de adquirir experiencia en el uso de la herramienta LEGO y poder participar en el torneo de robótica *FLL (First Lego League)*.

En los documentos encontrados relacionados con esta iniciativa no se da a conocer la manera en que fue evaluada. Únicamente se hace referencia a observaciones con respecto al trabajo realizado con *Scratch* donde se dice que los estudiantes adquieren gran capacidad de abstracción y resolución de problema, utilizando los conceptos de programación tratados en las conferencias que luego son aplicados en la práctica.

También hacen referencia a la parte motivacional de los estudiantes, comentando que en el momento de hacer uso del kit de *LEGO MINDSTORMS EV3*, los estudiantes se mostraron emocionados cada vez que el robot llevaba a cabo la tarea propuesta.

Según la observación realizada esta iniciativa promueve habilidades como la abstracción, transferencia y resolución de problemas, trabajando conceptos de

lógica de programación (da Silva, da Silva Martins, Ramos Nascimento, y do Santos Martins, 2016).

2.3. Discusión sobre las metodologías

Las iniciativas revisadas presentan una estrategia metodológica dirigida a estudiantes de escuelas y universidades por medio de talleres y sesiones, las cuales o pueden formar parte del currículo o pueden realizarse como clases optativas que se llevaron a cabo tanto de manera virtual como presencial. Las iniciativas presentan una duración promedio entre 3 y 16 días, con sesiones de 1 hora a 1 hora y media de trabajo. La extensión de estas sesiones está relacionada con la profundidad con que se trataron los conceptos de programación trabajados con los estudiantes, y es así, que las iniciativas que cuentan con mayor cantidad de sesiones son aquellas dirigidas a los estudiantes de los primeros años de universidad.

Cada una de las iniciativas plantea en su estrategia metodológica utilizar diversas herramientas en su mayoría libres y gratuitas, aprovechando los elementos con que cuenta cada una de las instituciones en las cuales se realizó.

Entre los lenguajes de programación utilizados con mayor frecuencia, está *Scratch* (Programa Nacional de Informática Educativa (PRONIE), Desarrollo del pensamiento computacional con Scratch, Desafío STEM – Fundación Telefónica España, El uso de Scratch en el desarrollo de la programación lógica como un aporte interdisciplinario), kit de robótica *LEGO* (Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje) y *Lighthbot - Scratch* (Program.AR, Herramientas lúdicas como apoyo a la enseñanza de la programación, Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación) o la combinación de *Scratch - Lego* (Uso de *Scratch* y *Lego Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación, Trabajar con la robótica educativa en la escuela primaria). Posiblemente esto se deba a que estas herramientas son creadas y acreditadas por instituciones científicas y tecnológicas que promueven la integración de las TIC en el aula, como es el caso del *MIT*, *Code.org* y *Lego Education*. En la Figura

2.3. Se puede visualizar del total de iniciativas, qué porcentaje ha utilizado cada una de estas herramientas.

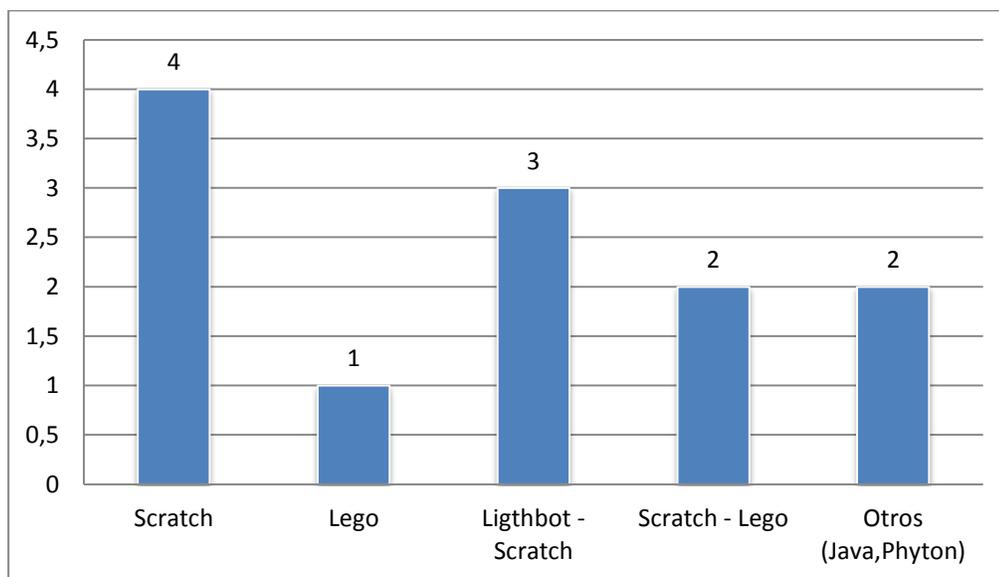


Figura 2.4 - Herramientas utilizadas en las iniciativas. Elaboración propia.

Con respecto a las estrategias metodológicas utilizadas en cada iniciativa, la mayor parte de ellas (8) trabajaron con actividades mediante sesiones y talleres guiados por docentes o instructores. Algunas iniciativas se complementaron con cursos virtuales, encuentros, campamentos, prácticas en laboratorios y torneos (Program.AR, PRONIE, Desafío STEM, Trabajar con la robótica educativa en la escuela primaria).

Estas actividades desarrolladas dieron como resultado: actividades basadas en juegos (RITA, Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación, Herramientas lúdicas como apoyo a la enseñanza de la programación, Desarrollo del Pensamiento computacional con *Scratch*, Uso de *Scratch* y *Lego Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación, El uso de *Scratch* en el desarrollo de la programación lógica como un aporte interdisciplinario), diseño y programación de robots (Programando con robots, Uso de *Scratch* y *Lego Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación, Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje,

Trabajar con la robótica educativa en la escuela primaria), desarrollo de productos digitales diversos (Programar.AR, PRONIE, Desafío STEM) y participación en concursos de robótica y tecnología (Trabajar con la robótica educativa en la escuela primaria, Programando con robots).

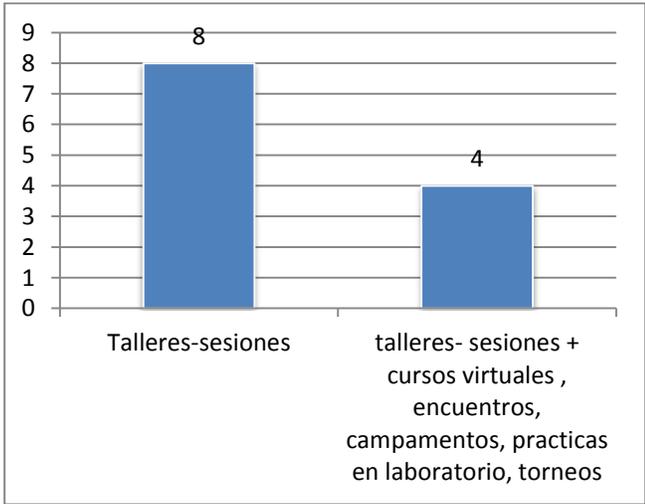


Figura 2.5 - Estrategias metodológicas utilizadas. Elaboración propia.

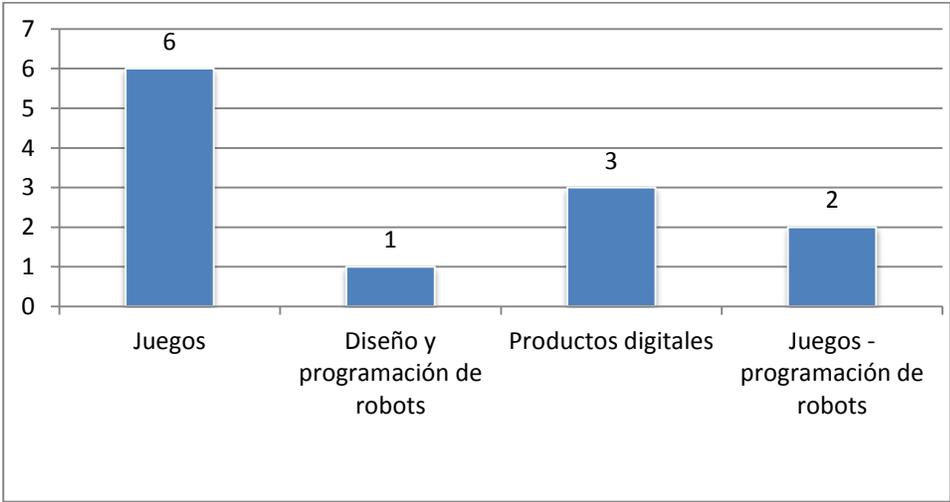


Figura 2.6 - Productos y actividades realizadas. Elaboración propia.

Al considerar las estrategias metodologías utilizadas en las iniciativas indagadas, se puede decir que según la información encontrada y analizadas en el

TFI²², la mayoría de estas iniciativas promueven varios de las competencias relacionadas con el pensamiento computacional que se nombran al inicio del capítulo, aunque no todas en igual medida y es por ello que resulta importante fortalecer algunas como es el caso del trabajo en equipo, la transferencia, la simulación de modelos antes de ponerlos a prueba y su implementación.

Las iniciativas que se caracterizan por abordar la totalidad de las competencias relacionadas con el pensamiento computacional son: Programando con *robots*, PRONIE, Uso de *Scratch* y Lego *Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación, Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje, Desafío STEM, El uso de *Scratch* en el desarrollo de la programación lógica como un aporte interdisciplinario, Trabajar con la robótica educativa en la escuela primaria.

Aquellas iniciativas que en la información recopilada no dan a conocer de qué forma desarrollan habilidades como el trabajo en equipo, la simulación de modelos y su implementación son: program.AR, Herramientas lúdicas como apoyo a la enseñanza de la programación, Desarrollo del Pensamiento computacional con *Scratch*.



Figura 2.7 - Desarrollo de competencias relacionadas con el pensamiento computacional. Elaboración propia.

²² Trabajo Final integrador, Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica. Sarmiento, Gorga, y Sanz (2016).

Las estrategias metodológicas utilizadas en estas propuestas según los documentos que las describen han dado resultados positivos al momento de ser evaluadas, de modo que se continúan aplicando año tras año en las instituciones para las cuales fueron diseñadas.

Cada una de las iniciativas analizadas son dispares en el momento de describir la estrategia metodológica utilizada, algunas cuentan detalladamente el proceso que siguieron en la implementación de la propuesta, mientras en otras, sobre todo en los proyectos a gran escala, tales como PRONIE, Program.ar, Desafío STEM, no es clara la estrategia metodológica y pedagógica utilizada para la implementación de las iniciativas. Tampoco se presenta una descripción de cómo se llevaron a cabo las evaluaciones para analizar el alcance de sus objetivos.

Entre las técnicas utilizadas para la evaluación de las iniciativas, se aplicaron encuestas, observación directa, comparación e indicador de desempeño y algunas combinaciones como: encuesta + evaluación Final, encuesta + observación y encuesta + comparación de desempeño (ver figura 2.8)

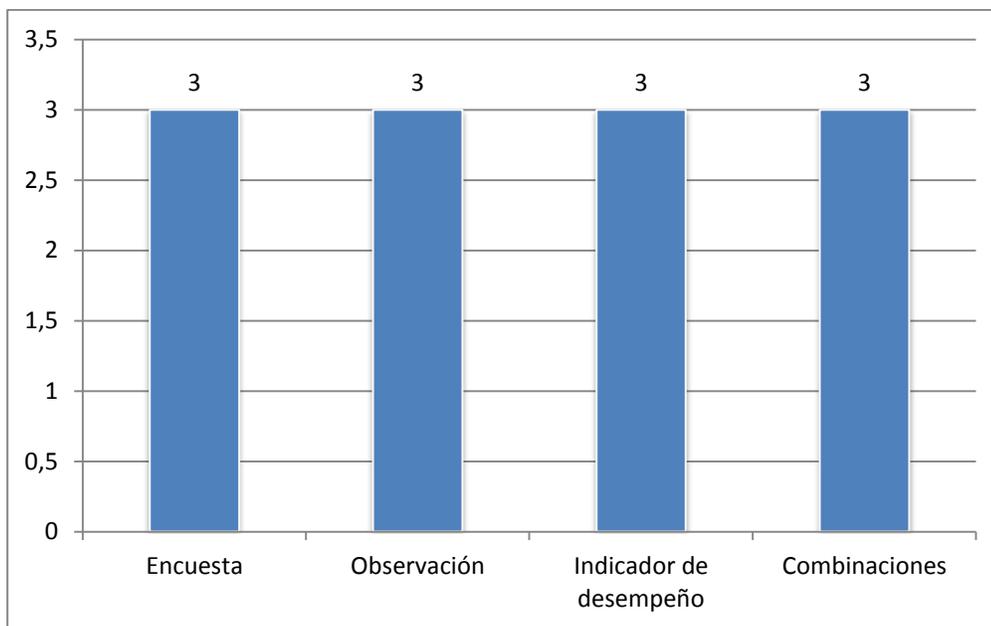


Figura 2.8 – Técnicas e instrumentos utilizados para la evaluación de las propuestas. Elaboración propia.

2.4. Conclusiones

En este capítulo se concluye que las iniciativas encontradas alrededor de Iberoamérica que surgieron entre los años 2009 y 2016 para el desarrollo de las competencias relacionadas con el pensamiento computacional y vinculadas a la enseñanza de la programación, proponen diversas estrategias metodológicas, que buscan de manera lúdica abordar conceptos básicos y desarrollar habilidades, conocimientos y actitudes que le sirvan a los estudiantes para dar solución a problemas de su entorno haciendo uso de herramientas computacionales. En este capítulo se describieron 12 experiencias de diversos países como se muestra en la Tabla 2.10.

Tabla 2.10 Iniciativas revisadas. Elaboración propia.

País	Propuesta
Argentina	Programando con robots
	Program. AR
	Herramientas lúdicas como apoyo a la enseñanza de la programación
	JET: Java en Escuelas Técnicas – Programando con RITA
	Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación
Brasil	El uso de <i>Scratch</i> en el desarrollo de la programación lógica como un aporte interdisciplinario
	Trabajar con la robótica educativa en la escuela primaria
Chile	Uso de <i>Scratch</i> y <i>Legó Mindstorms</i> como Apoyo a la Docencia en Fundamentos de Programación
	Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje
	Desarrollo del Pensamiento computacional con <i>Scratch</i>
Costa Rica	Programa Nacional de Informática Educativa - PRONIE
España	Desafío STEM – Fundación Telefónica España

Estas iniciativas resultan ser representativas entre las propuestas que se vienen trabajando alrededor de Iberoamérica, en ellas se muestra el uso de estrategias y herramientas que permiten trabajar conceptos básicos de programación acorde al tipo de participantes. Se trabaja por medio de sesiones y talleres presenciales o virtuales. Estas prácticas se retomarán luego para la propuesta metodológica a elaborar en esta tesis.

Capítulo 3

Descripción de herramientas para la enseñanza de la programación

3.1. Introducción

En este capítulo se presenta una descripción detallada de una serie de herramientas que acompañan una diversidad de iniciativas vinculadas a la enseñanza de la Programación. Se detallan cuestiones relacionadas con la calidad del entorno utilizado, su facilidad de uso, disponibilidad (libres y gratuitas), los requerimientos de *software* y *hardware*, entre otras, así como también la información técnica disponible (manuales, videos explicativos y material de apoyo a docentes) que facilitan su aprendizaje. Las herramientas seleccionadas han sido mencionadas en varias de las iniciativas revisadas en el capítulo anterior, y sirven para un primer acercamiento a la enseñanza de la programación.

3.2. Descripción de herramientas de software para la enseñanza de la Programación

En esta sección se describen las principales herramientas de *software* que han sido utilizadas en las iniciativas educativas mencionadas en el capítulo anterior, así como también otras revisadas posteriormente que se consideran de importancia para la enseñanza de la Programación.

Las herramientas se abordarán desde dos categorías: lenguajes de programación y lenguajes de programación basados en bloques de arrastrar y soltar.

3.2.1. Lenguajes y entornos de programación

Greenfoot

Greenfoot es un entorno de desarrollo interactivo *Java*, educativo, gratuito y multiplataforma, creado para ayudar a los estudiantes a aprender. Es un proyecto del grupo de investigación de la escuela de computación de la Universidad de *Kent* en *Canterbury*, Reino Unido. En este entorno, los estudiantes adquieren experiencia en la terminología empleada y los conceptos detallados de programación orientada a objetos, mientras desarrollan aplicaciones graficas bidimensionales tales como juegos y simulaciones en un ambiente divertido,

interesante y visualmente atractivo, que hace que el entorno sea fácil de usar para personas sin experiencia en este tipo de lenguajes. Por esta razón es adecuado para estudiantes de secundaria y primeros años de universidad.

Cuenta con una comunidad llamada *Greenroom* integrada por docentes, que pueden compartir recursos de enseñanza y discutir sobre la mejor manera de usar *Greenfoot*, como herramienta para el aprendizaje.

Greenfoot cuenta con un sitio web²³ en donde se encuentran tutoriales y videos para guiar el aprendizaje. Allí también se pueden publicar los proyectos para ser compartidos, comentados y utilizados por las personas interesadas, con el fin de aprender de los proyectos de otros, obtener ideas, ayudar y solucionar problemas de la programación.

Este *software* no se menciona en el capítulo anterior, pero es una opción a tener en cuenta para el diseño de la estrategia metodológica.

Gobstones

Es una secuencia didáctica que utiliza un lenguaje de programación como parte de una propuesta para la enseñanza y el aprendizaje de ideas básicas en programación. Fue creado en el 2009 por Pablo E. Martínez López y Eduardo A. Bonelli en la Universidad de Quilmes. Su diseño tiene en cuenta las características que se buscan fortalecer en los estudiantes de un curso inicial en la carrera de Tecnicatura Universitaria en Programación Informática. Por medio de la resolución de problemas de programación se intenta motivarlos y captar su atención utilizando elementos simples como un tablero, bolitas de colores y un dispositivo que posee un cabezal por medio del cual se puede recorrer el tablero utilizando instrucciones sencillas a través de comandos.

Gobstones posee conceptos fundamentales comunes a todos los lenguajes. Su sintaxis es similar a la de lenguajes como C o Java. Se basa en tres elementos fundamentales, el desarrollo de la abstracción, el aprendizaje de la división de tareas y la obtención de pautas de estilo que favorezcan la posterior interpretación de los programas generados. Todo ello se lleva a cabo mediante conceptos bien

²³ <https://www.greenfoot.org/door>

simples para promover el aprendizaje inicial, ya que trabaja con tipos básicos de datos, y no agrega nociones complejas de alcance de variables o elementos globales (Martínez López, Bonelli, y Sawady O'Connor, 2007).

En 2013 se gesta la idea de *XGobstones*, un lenguaje basado en el original *gobstones*, que incorpora estructuras de datos básicas y otros elementos avanzados. Actualmente posee una implementación en *Python*, denominada *PyGobstones* que implementa un ambiente de desarrollo integrado para *Gobstones*, proveyendo edición del programa, validación de sintaxis, edición de tableros y algunas otras características adicionales.

*Gobstones*²⁴ es un software libre y disponible para *Linux* y *Windows*, para su buen funcionamiento es necesario instalar Python 2.7 y la librería PyQt4.

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Program.AR.

Pilas Engine

Este proyecto se lanzó el 1 de agosto del 2010 y está a cargo de Hugo Ruscitti. Está inspirado en las ideas de Seymour Papert, en el cual se propone una herramienta para construir video juegos de manera sencilla y llamativa. Está enfocada, principalmente, a jóvenes interesados en aprender a programar que no cuenten con conocimientos previos en esta área y a la vez proporcionar un entorno para desarrolladores casuales y novatos. Ofrece una colección de actores, escenas prediseñadas y rutinas que facilitan el desarrollo y que pueden ser modificadas, añadiendo comportamientos y haciéndolos interactivos, con la posibilidad de visualizar el resultado de las instrucciones escritas.

Entre sus características se puede mencionar que es una herramienta libre y gratuita bajo la licencia LGPL, multiplataforma (funciona sobre GNU/*Linux* y *Windows*). Se puede descargar del sitio oficial²⁵, utiliza el idioma español y cuenta con material de apoyo, ejemplos y tutoriales que facilitan su aprendizaje. Por tratarse de un entorno basado en *Python*, la programación se realiza mediante

²⁴ <http://www.gobstones.org/>

²⁵ <http://pilas-engine.com.ar/>

instrucciones simplificadas que permiten al usuario abstraerse de los detalles de dicho lenguaje.

Es un entorno que cambia constantemente por la participación de una comunidad abierta de programadores, docentes y estudiantes interesados en colaborar en el desarrollo del entorno.

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Program.AR.

PSeInt

*PseInt*²⁶ (Pseudo Intérprete) es un Entorno de Desarrollo Integrado (IDE) para Pseudocódigo que permite editar e interpretar programas escritos. Es una herramienta para aprender lógica de programación. Es simple de usar y está en idioma español. Fue pensada para estudiantes sin experiencia en dicha área, donde crean pseudocódigos para introducir conceptos básicos como el uso de estructuras de control, expresiones, variables, etc. Este programa pretende facilitar la escritura de algoritmos en pseudolenguaje permitiendo crear, almacenar, ejecutar y corregir fácilmente programas por medio de herramientas que brindan asistencia y ayudan en la tarea de detectar errores y comprender la lógica del algoritmo.

Entre sus características y funcionalidades se pueden mencionar las siguientes:

- Presenta herramientas de edición para escribir algoritmos en pseudocódigo en español
- Permite generar y editar el diagrama de flujo del algoritmo
- Permite la edición simultánea de múltiples algoritmos
- El pseudo - lenguaje utilizado es configurable
- Puede interpretar (ejecutar) los algoritmos escritos
- Determina y marca claramente los errores

²⁶ <http://pseint.sourceforge.net/>

- Permite convertir el algoritmo de pseudocódigo a código en numerosos lenguajes de programación
- Ofrece un sistema de ayuda integrado acerca del pseudocódigo y el uso del programa
- Es multiplataforma (probado en *Microsoft Windows*, *GNU/Linux* y *Mac OS X*)
- Es totalmente libre y gratuito (licencia GPLv2)

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Program.AR.

Python

Python fue presentado, en los años '90, por el investigador holandés Guido Van Rossum. Es un lenguaje de programación interpretado, con tipado dinámico, orientado a objetos, multiplataforma, de sintaxis y semántica sencilla y consistente, que permite escribir programas con muy pocas líneas de código logrando diversas funcionalidades. Cuenta con una gran cantidad de librerías con varios propósitos posibilitando al alumno explorar distintas funciones que se pueden llevar a cabo con este lenguaje, aplicaciones web, serialización de objetos, interfaces gráficas, etc (Lovos, Gibelli, y Bertone, 2014).

Es un programa multiplataforma, que posee una licencia de código abierto, denominada “*Python Software Foundation License*” y se puede descargar del sitio oficial²⁷.

En el documento “*Python como primer lenguaje de programación*”, Ortiz (2010) describe las características principales de este entorno.

- Es un lenguaje orientado a objetos pero también soporta estilos de programación procedural y funcional.
- Corre en múltiples plataformas.

²⁷ <https://www.python.org/>

- Su sintaxis es sencilla y consistente, comparada con otros lenguajes como Java o C.
- Utiliza tipos dinámicos.
- Es adecuado para programar scripts y aplicaciones de gran tamaño.
- Es modular.
- Administra automáticamente la memoria a través de recolección de basura.
- Incluye múltiples bibliotecas de clases.
- Cuenta con una comunidad interesada en promover su desarrollo y adopción.
- Es ideal para llevar a cabo programación experimental y educativa.

Python es un lenguaje ideal para la enseñanza y aprendizaje en un primer curso de programación, pues ofrece simplicidad sintáctica y semántica y su modo interactivo permite a los estudiantes ver el resultado de forma inmediata al probar nuevos conceptos. Cuenta con recursos bibliográficos y materiales de apoyo que surgen de las diversas experiencias generadas en diversos entornos educativos.

Este *software* se mencionó en el capítulo anterior pues forma parte de las herramientas utilizadas en la iniciativa Programando con robots.

Pygame

*Pygame*²⁸ es una librería gráfica para *Python*, que surge en el año 2000. Es multiplataforma, libre y de código abierto, se distribuye bajo licencia GNU LGPL que permite desarrollar aplicaciones multimedia, dibujar gráficos, manejar imágenes, interactuar con el usuario, controlar los escenarios, realizar animaciones, interactuar con el teclado, *mouse*, *joystick* y reproducir archivos de sonidos y audio, entre otros.

Para utilizar *Pygame* es necesario conocer algunos conceptos simples y algunas funciones básicas predefinidas, tales como creación de ventanas y funciones de dibujo. Entre las ventajas de *PyGame* se pueden mencionar la facilidad de instalación y el manejo de diferentes formatos de imágenes, y como

²⁸ <http://www.pygame.org/download.shtml>

desventaja se puede mencionar que la velocidad de procesamiento disminuye considerablemente al contar con mucha información almacenada, tanto en las estructuras de datos como en la memoria, (Harari y Banchoff, 2011).

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Program.AR.

Ruby

*Ruby*²⁹ es un lenguaje de programación creado en Japón en 1995 por Yukihiro Matsumoto. Es un *software* libre, multiprograma que se puede, incluso, ejecutar en dispositivos móviles. Está basado en características similares a la de otros lenguajes como *Python* o *Perl* y se dio a conocer en occidente, tiempo después de su creación debido a que la documentación estaba escrita en japonés.

En el año 2004 aumentó el interés por *Ruby*, debido a *Ruby on Rails*, que es un entorno para desarrollo *web* creado por David Heinemeier Hansson. *Ruby* es un lenguaje de programación orientado a objetos, con sintaxis relativamente simple, que hace de las variables simplemente instancias de estos objetos, y las funciones son tratadas como métodos de estas clases. Es un lenguaje dinámico e interpretado que no tiene compilador, sino que traduce y ejecuta el programa al mismo tiempo, mediante una consola interactiva. Esto permite que sea posible modificarlo en tiempo de ejecución, añadiendo código, editando variables o extendiendo algunas clases. Su funcionamiento se puede extender mediante bibliotecas escritas en *Ruby* o utilizando el lenguaje C.

El sistema de tipado que utiliza *Ruby* es el *Duck Typing*, donde los métodos y propiedades de un objeto determinan su validez semántica, en vez de su herencia de clases o la implementación de una interfaz específica. Por estas características *Ruby* tiene un amplio alcance y se utiliza en aplicaciones diversas desde desarrollo *web* hasta simulaciones en entornos complejos (García, Sánchez, y Guerra, 2015; Rodríguez, León, Miranda, Segredo, y Segura, 2013).

Este *software* se mencionó en el capítulo anterior pues forma parte de las herramientas utilizadas en el proyecto Program.AR.

²⁹ <https://www.ruby-lang.org/es/>

3.2.2. Lenguajes de programación visual

Alice

Es un entorno de programación 3D desarrollado en *Java*, de libre disposición con código abierto, creado por la *Carnegie Mellon University* en el 2004, para la creación de juegos, historias y videos que pueden ser compartidos en internet.

Por medio de este entorno intuitivo y de fácil uso los estudiantes aprenden conceptos fundamentales de programación orientada a objetos, arrastrando y soltando cuadros gráficos para realizar la animación, construyendo algoritmos y visualizándolos de forma gráfica. De esta manera se motiva a los estudiantes a crear historias animadas que ejecutan paso a paso, permitiéndoles ver inmediatamente el resultado de su trabajo.

El uso de *Alice* permite que el acercamiento a la programación sea agradable, aprendiendo conceptos fundamentales de una manera divertida por medio de un aprendizaje visual donde se puede lograr una animación sin errores. En general en los métodos tradicionales para enseñar POO el alumno ve su trabajo en líneas de código y solo se muestra un resultado final cuando el programa no genera ningún error de compilación. Esta situación puede frustrar al alumno y lo desmotivan (Ramírez, Lucio, Garza, García, Vargas, 2011).

El entorno resulta ser una herramienta de desarrollo atractiva debido a su tridimensionalidad, su interfaz lúdica y colorida, motivando a los estudiantes a ejercer creativamente las tareas planteadas. De modo que esta herramienta demuestra un potencial para promover aspectos como el pensamiento algorítmico, la programación, el modelado y la abstracción, habilidades fundamentales del pensamiento computacional (Werner, Denner, Bliesner y Rex, 2009).

Rebeca es la versión en español de *Alice 2.2*, en este momento se cuenta con la versión 2.4 que se puede configurar para trabajar en español, y la última versión 3.1 que se encuentra en etapa de desarrollo y traducción en curso.

Los requerimientos para el funcionamiento del programa son:

Alice 2.4 funciona en Windows 7 y 8, Vista, XP y 2000, así como en Linux (Huayra Linux, Ubuntu, RedHat). Los requerimientos de hardware son procesador Pentium II (o equivalente) y 1G de memoria RAM.

Para *Alice 3.1* es recomendable contar con 2GB de memoria RAM, aunque con 1 GB es suficiente. *Alice 3.1* requiere tener instalada *Java JDK*. El programa puede ser descargado de su sitio oficial³⁰.

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Program.AR.

App Inventor

Es una aplicación web basada en *Java* desarrollada por *Google*. En 2011, finalizado su desarrollo, fue publicado como *Software Libre* bajo la licencia Apache 2.0, actualmente el Instituto Tecnológico de *Massachusetts* (MIT) está a cargo de su desarrollo. Esta herramienta permite realizar aplicaciones para móviles y *tablets* que funcionan con el sistema operativo *Android*. Cuenta con un entorno visual e intuitivo que permite arrastrar y soltar objetos, de manera similar a *Scratch*, lo cual permite programar sin escribir ningún tipo de código.

El programa se ejecuta vía web³¹ mediante el editor de bloques y el emulador que permite probar las aplicaciones que se desarrollan, luego de ser probadas estas se podrán compilar para ser utilizadas en dispositivos *Android*. También se guardan los proyectos en la web y la pc, de modo que puedan ser compartidas y revisadas desde cualquier lugar que cuente con conexión a Internet.

App Inventor es parte de un movimiento en el ámbito de la educación, que se inició con Seymour Papert y el grupo de Logo del MIT y también con el trabajo de Mitchel Resnick, *Lego Mindstorms* y *StarLogo*. Este entorno permite pensar lógicamente y solucionar problemas de manera metódica, usando bloques de programación que tienen elementos comunes a la mayoría de los lenguajes de programación existentes, con los que se pueden construir bucles, condiciones, variables, etc (Rederjo, 2013).

³⁰ <https://www.alice.org/>

³¹ <http://appinventor.mit.edu/explore/>

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Desafío STEM – Telefónica Fundación.

AstroCódigo

AstroCódigo es un juego serio presentado en el año 2017 como trabajo de tesina de grado en la Facultad de Informática de la Universidad Nacional de La Plata, Argentina.

Tiene como objetivo introducir por medio del juego conceptos básicos de programación, tales como algoritmo, secuencias y estructuras de control como la decisión, la repetición y la iteración. Basado en una temática de ciencia ficción donde el participante cumple el rol de un astronauta que cumple una serie de misiones en el espacio.

Por medio de bloques de arrastrar y soltar el jugador soluciona distintas situaciones que se presentan, además de estos desafíos el juego tiene la oportunidad de crear escenarios personalizados, a través de una herramienta web, de modo que los docentes puedan crear sus propios escenarios, de acuerdo a los objetivos que se plantee enseñar.

El juego se puede descargar de la página web <http://www.astrocodigo.com/>, para utilizar el juego se debe contar con conexión a internet y una computadora con sistema operativo *Windows, Linux o MacOS*.

E-toys

Etoys es un entorno multimedia y un ambiente de programación basado en *Squeak*, que funciona como herramienta educativa para niños. Es multiplataforma y de código abierto y gratuito. Su desarrollo es acompañado por una importante comunidad académica. En un principio participó Alan Kay de *Disney Engineering Research*, influenciado por Seymour Papert y el lenguaje de programación Logo. Luego su desarrollo fue continuado por Scott Wallace, Ted Kaehler, John Maloney, Dan Ingalls.

Cuenta con una importante comunidad alrededor del mundo, como por ejemplo EEUU, Europa, Sudamérica, Japón, Corea, India, Nepal y otras que comparte sus

creaciones en el sitio web. Es un entorno multimedia que permite creaciones de diferentes proyectos.

Al utilizar *Etoys*, los estudiantes aprenden a resolver problemas creativamente, desarrollan el pensamiento crítico y aumentan su autoestima al observar los resultados logrados.

Desde 2006 *Etoys* es utilizado en los proyectos de educación OLPC, como herramienta educativa para la simulación de situaciones de la vida real a través de la programación. Dado que es un lenguaje versátil y muy amigable, puede ser instalado en equipos con sistemas operativos *Windows*, *Mac* y *Linux*. Puede descargarse de su sitio oficial <http://www.squeakland.org/download/>.

Game Engine de Blender

Game Engine es el motor de juegos de *Blender*³² que permite crear aplicaciones 3D interactivas y simulaciones. En este entorno se puede modelar cualquier tipo de objeto. El usuario debe estar familiarizado con la programación de alto nivel para diseñar los comportamientos por medio de *Logic Bricks* sin necesidad de escribir código. De esta manera, ofrece un entorno visual sencillo para el diseño de las aplicaciones.

Kodu

Kodu es un lenguaje de programación visual e interpretado de alto nivel. Fue creado en 2009, por los laboratorios *Future Social Experiences (FUSE)* de *Microsoft*, y ha sido pensado específicamente para la creación de juegos. De manera visual, este lenguaje permite que niños y adultos, logren programar utilizando la PC o el *Xbox*, sin necesidad de escribir código, promoviendo la creatividad, la resolución de problemas y el desarrollo de narraciones.

La interfaz de *Kodu* es simple y está basada en íconos. Proporciona objetos y herramientas para la creación de escenarios en distintos mundos. Los programas se componen de páginas, que se dividen en líneas de programación que son llamadas reglas y estas a su vez contienen condiciones y acciones que se

³² <https://www.blender.org/download/>

representan por medio de baldosas. La programación de cada objeto se define por, al menos, una página y cada página tiene una o más reglas, de modo que se pueden expresar conceptos avanzados de diseño de juego de manera simple, directa e intuitiva.

El programa para la PC se puede descargar de forma gratuita desde el sitio web de *Microsoft*³³. Existe una versión en inglés y otra en español.

Los requerimientos para su instalación son los siguientes: *Windows 10, Windows 7, Windows 8, Windows Vista o Windows XP*, una tarjeta gráfica compatible con *DirectX 9.0c* y *Shader Model 2.0* o superior, también *.NET Framework 4.0* o superior.

Este entorno se propone enseñar a programar por medio de ensayo y error, desarrollando el pensamiento computacional. Promueve en los niños cualidades tales como predecir el comportamiento del programa desde el código, realizar simulaciones mentales ejecutando el código en su cabeza y reconocer patrones en el código entre otros. Además, cuenta con materiales de apoyo en inglés que se encuentran en el sitio web de *Kodu*.

LightBot

Es un *software* enfocado a la enseñanza de conceptos de programación que, por medio de pequeños retos, hace que el usuario inconscientemente realice algoritmos que permitan superar el nivel en el que se encuentra.

Se basa en un juego de casillas con un escenario tridimensional, de interfaz amigable con colores planos y un mínimo de texto, en el cual se mueve un agente (robot) que debe ser programado. Por medio de un conjunto de instrucciones que se escriben utilizando botones se representan las acciones que puede realizar el robot. El objetivo es encontrar la casilla de salida. Se pueden utilizar movimientos en distintas direcciones a partir de los botones que se encuentran en el recuadro de acción y cuenta con movimientos para avanzar, girar 90° en un sentido y en el otro, saltar y encender una bombilla, entre otros.

³³ <https://www.kodugamelab.com/>

Al programar las acciones se pulsa un botón para ver su ejecución. En caso de que el procedimiento sea correcto y que el robot alcance la casilla de salida se pasa al siguiente nivel.

Se encuentra disponible en dos versiones, una paga para *Windows*, *Mac*, *Android* y *Iphone* con varios niveles y otra versión gratuita y disponible en la web <https://lightbot.com/hocflash.html> que cuenta solo con algunos niveles (Dapozo et al., 2016; Saturio, García, y Hernández, 2015).

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Program.AR.

Minecraft Hora del Código

Es un video de juego de construcción tipo mundo abierto, creado por la empresa *Mojang AB*. Lanzado en el año 2011, en el año 2014 esta empresa fue adquirida en su totalidad por *Microsoft*, en el año 2015 la intención de *Microsoft* por impulsar la integración de las tecnologías en el aula, lo lleva a unirse al movimiento La Hora del Código, proyecto que promueve el aprendizaje de la programación alrededor del mundo. Para que el juego *Minecraft* se pudiera integrar a esta experiencia *Microsoft*, *Mojang AB*, y *Code.org* presentaron unas versiones especiales gratuitas y libres llamadas *Minecraft Hora del Código* la cual contiene actividades como: *Minecraft: Hero's Journey* y Diseñador de *Minecraft* con versión online y también Aventurero de *Minecraft*, que cuenta con una versión descargable.

Además de la creación de este material diseñado para el aprendizaje de la programación, *Microsoft* continúa adaptando versiones de este famoso juego que puedan ser utilizadas como herramientas educativas, es así que en el año 2016 se lanza *Minecraft: Education Edition*, que tiene un costo de 5 dólares por usuario al año y cuenta con tutoriales, planes de estudio, y mundos ya creados. Según la información encontrada en la página de *Microsoft* se continúan desarrollando versiones actualizadas donde aparecerán nuevas características.

Para utilizar la herramienta *Minecraft: Education Edition* se debe contar con las siguientes especificaciones: *Windows 10* o *MacOS*, Registrar una cuenta *Office*

365 *Education*. La versión descargable puede ser utilizada en *Windows 7* o *Windows 10*.

Pilas Bloques

Pilas bloques es un desarrollo del equipo del proyecto Program.AR. Es un entorno para la enseñanza de la programación basado en Pilas *Engine*, que por medio de bloques trabaja pequeños desafíos utilizando estrategias para la solución de problemas, modularizando este en partes para poder darle respuesta.

Al igual que Pilas *Engine*, bloques es una herramienta libre y gratuita, en idioma español y cuenta con material de apoyo que facilita el trabajo con estudiantes que no tienen conocimientos previos en programación, se puede descargar del sitio oficial <http://pilasbloques.program.ar/>.

Este *software* se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en el proyecto Program.AR.

Proyecto ICARO

Es un proyecto educativo que nace en Córdoba, Argentina, con el objetivo de desarrollar *software* y *hardware* libre para la enseñanza de la programación y la robótica en instituciones educativas. Su objetivo es explicar y simplificar el complejo contenido técnico de los fundamentos de la programación (iteraciones, repeticiones, recursividad, si condicional) aplicándolos a un contexto físico como robots o sistemas automatizados. Para ello, se aprovechan los equipos de cómputo con que cuentan los colegios, facilitando el acceso y la apropiación de la tecnología por parte de los alumnos.

Teniendo en cuenta que es un proyecto multidisciplinar con *hardware* y *software* libre que puede ser modificado continuamente, existen comunidades dentro y fuera del país que colaboran con el perfeccionamiento del código entre otras acciones. Algunas de esas comunidades se encuentran en Venezuela, Panamá y Nicaragua, y están conformadas por profesionales en distintas ramas como la electrónica, mecánica, programación, diseño y pedagogía, que resultan una valiosa fuente de conocimiento en el momento de compartir experiencias.

Ícaro, está compuesto por paquetes de *software* que trabajan con *hardware* de bajo costo y fabricación, cuyas versiones pueden ser ejecutadas en *Linux*, permitiendo diseñar robots de forma sencilla. El módulo principal de *software* es Icaro Bloques³⁴ que brinda un entorno fácil de utilizar, enfocado a la programación orientada a objetos que por medio de los bloques permite hacer programas complejos. Cuenta con la opción de ver en el mismo instante el código generado escrito en el entorno C, brindando un acercamiento a este lenguaje de programación que tiene un amplio uso en la industria. Además de visualizarlo, es posible escribirlo o modificarlo según se requiera. Este código luego será cargado en el microcontrolador que acciona el robot.

R.I.T.A (Robot Inventor To Teach Algorithms)

Es un juego de programación presentado como tesina para la licenciatura en informática en la Universidad Nacional de La Plata año 2012. R.I.T.A. (*Robot Inventor to Teach Algorithms*) integra los *frameworks* *OpenBlocks* y *Robocode*, con ella los estudiantes estimulan las habilidades en la resolución de problemas contribuyendo al pensamiento computacional, a la transferencia de saberes en otros contextos, promoviendo el pensamiento analítico, sistemático, fomentando la creatividad y el trabajo colaborativo. Esto se puede lograr por medio del diseño de estrategias de supervivencia para la batalla de robots virtuales programados mediante la metáfora visual de arrastre de bloques y conectores que deben encastrarse, y que puede ser visualizada en código *Java*.

Por medio del entorno se aprenden conceptos básicos introduciendo algunas nociones de programación orientada a objetos tales como estado y comportamiento, herencia y sobrescritura, familiarizándose de esta manera con la sintaxis *Java*, al diseñar estrategias de batalla de un robot llamado *JuniorRobot* que es el tipo de robot más simple construido dentro las categorías de esta aplicación, recomendada para programadores novatos.

R.I.T.A como herramienta didáctica de carácter lúdico motiva el interés hacia el aprendizaje de programación, pues permite compilar el código *Java* generado y

³⁴ <http://roboticaro.org/software/>

ponerlo en ejecución en el escenario brindado por *Robocode*, comprobando las acciones programadas de manera visual en el mismo instante. Cuando los estudiantes terminan de escribir su estrategia, la prueban combatiendo con el robot que programaron y modificando la programación, en el caso que su táctica no resulte efectiva. La motivación se da, también, porque la programación por bloques resulta ser más agradable, puesto que elimina la dificultad que tienen los estudiantes en el momento de enfrentarse a un lenguaje de programación por primera vez. De igual manera, para los estudiantes resulta interesante poder competir con sus compañeros en el diseño y perfeccionamiento de las estrategias de ataque y defensa del robot con el que compiten (Aybar, Queiruga, y Banchoff, 2012).

RITA es una herramienta gratuita y de código abierto, está disponible para su uso y modificación bajo licencia GNU GPL en el repositorio gitHub. RITA en gitHub: <https://github.com/vaybar/RITA>. Cuenta con versiones para *Windows* y *Linux*.

Este *software* se mencionó en el capítulo anterior pues forma parte de las herramientas utilizadas en la iniciativa JET: Java en Escuelas Técnicas – Programando con RITA.

Scratch

Es un entorno de programación multimedia basado en “*Squeak*” desarrollado en 2007 por “*Lifelong Kindergarten*” del *Media Laboratory MIT* de la Universidad de California en Los Ángeles.

Scratch cuenta con una interfaz atractiva y accesible para cualquier persona interesada en aprender conceptos básicos de programación en un nivel inicial. Trabaja con bloques gráficos de colores que deben utilizarse según el tipo de instrucción que se quiera dar. De esta manera se facilita el uso porque plantea un trabajo visual donde no hay necesidad de escribir ningún tipo de código, sino que arrastrando, soltando y encastrando se logra programar.

El entorno cuenta con fondos, personajes y objetos, pero también brinda la posibilidad de utilizar o editar escenarios propios u objetos, importándolos desde

otros entornos. Ello tiene la finalidad de crear juegos, animaciones, cuentos, con los que además de aprender a programar, también pueden tratar contenidos de diversos temas. Se accede al entorno de manera *on line* o descargando el programa que es libre y gratuito desde su sitio oficial³⁵, multiplataforma y multilinguaje. Puede ser utilizado en dispositivos digitales móviles.

Al utilizar *Scratch* los estudiantes involucran un conjunto de conceptos que se manejan en otros entornos de programación como por ejemplo secuencias, bucles, paralelismo, eventos, condicionales, operadores y datos. Otra característica sobresaliente de *Scratch* es que se ha convertido en una comunidad que, por medio de la web se comunica y comparte proyectos, scripts y personajes. Esta característica permite realizar un trabajo colaborativo en el cual todos pueden aportar, rediseñando y mejorando los proyectos que se han compartido, posibilitando también, crear aplicaciones más complejas de lo que una sola persona habría podido crear.

Por otro lado, *Scratch* ofrece la oportunidad de crear aplicaciones robóticas haciendo uso de las tecnologías de hardware libre, como por ejemplo controlando una tarjeta Arduino mediante instrucciones *Scratch* en el programa S4A (*Scratch* para Arduino) que es una variante del entorno creado por *Citilab* (Brennan y Resnick, 2012; Vázquez-cano y Delgado, 2015).

Este software se mencionó en el capítulo anterior pues hace parte de las herramientas utilizadas en las iniciativas: Desarrollo del Pensamiento computacional con *Scratch*, Uso de *Scratch* y *Lego Mindstorms* como Apoyo a la Docencia en Fundamentos de Programación, Desafío STEM – Telefónica Fundación, Program.AR, Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación, Herramientas lúdicas como apoyo a la enseñanza de la programación, El uso de *Scratch* en el desarrollo de la programación lógica como un aporte interdisciplinario, Trabajar con la robótica educativa en la escuela primaria.

³⁵ <https://scratch.mit.edu/>

3.3. Conclusiones

En este capítulo se describen 19 lenguajes y entornos de programación que han sido desarrolladas por programadores pertenecientes a diversos grupos de investigación de instituciones educativas y tecnológicas interesadas en la enseñanza, cuyo objetivo es lograr que, a través de esas herramientas, las personas se acerquen al pensamiento computacional o en otros casos que aprendan conceptos básicos de programación. Están principalmente orientadas a chicos en edad escolar y programadores novatos. En algunos casos, requieren tener algún tipo de conocimiento o experiencia básica relacionada con la temática.

En su mayoría, estas herramientas cuentan con interfaces atractivas, intuitivas, simples y de fácil manejo, donde se abordan conceptos básicos de programación, codificando por medio de objetos que se arrastran y sueltan y que suelen ser bloques gráficos o botones de acción, y también por medio de instrucciones escritas de sintaxis simple. En general, se dispone de versiones en español o inglés, cuentan con materiales de apoyo accesibles desde sus páginas oficiales, así como también desde diversos sitios web. Sus licencias son libres y/o gratuitas, multiplataforma y con requisitos básicos de *hardware* y *software*.

A partir del análisis de las diversas iniciativas presentadas en el capítulo 2 vinculadas al desarrollo de conocimientos, actitudes y habilidades que promueven el pensamiento computacional y del conjunto de herramientas de programación descritas en este capítulo, se presenta a continuación una metodología propia que propone llevar adelante experiencias de abordaje del pensamiento computacional en el nivel inicial y medio.

Capítulo 4

Metodología para el diseño de experiencias que fomenten el desarrollo del pensamiento computacional

4.1. Introducción

En los capítulos anteriores, se han revisado una serie de estrategias metodológicas implementadas alrededor de Iberoamérica, que desarrollan competencias relacionadas con el pensamiento computacional, haciendo uso de herramientas para la enseñanza de la programación y temas afines.

Del mismo modo se han explorado diversos lenguajes y entornos de programación, algunos de ellos utilizados en las propuestas examinadas y otros encontrados mediante la búsqueda en sitios educativos enfocados a esta temática.

Después de analizar las iniciativas encontradas y reconocer las fortalezas y debilidades en cada una, se hace foco como parte de esta tesis en generar una nueva metodología con el objetivo de fortalecer el desarrollo de los conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional.

Para generar la propuesta se tiene en cuenta el concepto de metodología enunciado por la RAE (Real Academia Española), donde se define que una metodología es un conjunto de métodos que se siguen en una investigación científica. De igual manera se conceptualiza la palabra competencias como las habilidades, conocimientos, actitudes, capacidades, valores, comportamientos y en general atributos personales, que se relacionan más directamente con un desempeño exitoso de las personas en sus trabajos, funciones y relaciones (Cataldi et al., 2010). Con base en estos conceptos, el capítulo 4 propone y describe una serie de etapas propias de una metodología para la generación de una propuesta educativa, cuyo objetivo es el desarrollo de competencias relacionadas con el pensamiento computacional.

El motivo por el cual se hace importante desarrollar una propuesta metodológica, es el interés en el apoyo que se puede brindar a los docentes, para la planificación de experiencias orientadas al desarrollo del pensamiento computacional mediante el uso de lenguajes de programación, puesto que estos son el vehículo perfecto para desarrollar este tipo de habilidades, ya que implica la resolución de problemas haciendo uso de conceptos informáticos (Compañ-Rosique, Satorre-Cuerda, Llorens-Largo, y Molina-Carmona, 2015).

En los antecedentes revisados se han encontrado ejemplos de iniciativas que abordan y buscan innovar en esta temática. Sin embargo, varias de ellas no presentaban procesos de evaluación y otras no explicitaban la forma en que se alcanzaron los resultados. También se observó que algunas no especificaban cuestiones como el contexto con el que se trabajó. Esto puede deberse o bien a una falta de explicitación en los artículos revisados, o que los grupos que llevan adelante el trabajo no hayan tenido en cuenta estos aspectos relevantes en la planificación de la experiencia.

La metodología que se presenta en este capítulo propone seguir una serie de etapas que buscan facilitar el diseño, la planificación de estrategias y actividades que se puedan utilizar en el aula. Todo esto sin perder de vista que la tecnología se vinculará a la propuesta, teniendo en cuenta los objetivos que se persiguen junto con las necesidades pedagógicas y curriculares. Así como lo plantean Harris y Hofer:

“En primer lugar, la integración satisfactoria de la tecnología se basa en el contenido curricular y en los procesos de aprendizaje relacionados con el contenido; en segundo lugar, en el uso inteligente de las tecnologías educativas.

Al integrar las tecnologías educativas en la enseñanza, la planificación de los docentes debe organizarse en torno a los requisitos de los diseños curriculares, a prácticas pedagógicas eficaces y a las posibilidades y limitaciones de las tecnologías disponibles” (Harris y Hofer, 2009 citados en Magadán, 2012, p4).

A continuación, se revisan una serie de antecedentes que servirán de base y fundamento para la propuesta metodológica que se presenta.

4.2. Antecedentes

La integración de TIC en procesos educativos ha dado lugar a diferentes estrategias, sugerencias, modelos, y metodologías, que son propuestos por diversidad de autores para guiar el diseño de las experiencias educativas con TIC (Sarmiento, Gorga, y Sanz, 2016). Si bien no es el objetivo de este trabajo realizar

una revisión completa de esta temática, se retoman algunos antecedentes que permiten fundamentar y orientar la propuesta metodológica que aquí se realiza.

4.2.1. El modelo *TPACK*

Para lograr que la propuesta educativa diseñada por un docente, desde cualquier disciplina, a partir de las etapas que se presentan en este capítulo, sea consistente se hace importante que el docente o los docentes que lleven a cabo esta planificación tengan habilidades relacionadas al conocimiento Tecnológico Pedagógico del Contenido acorde a lo que propone el modelo *TPACK*³⁶. Además, se busca que su propuesta posea un equilibrio entre estos componentes (Koehler y Mishra, 2015).

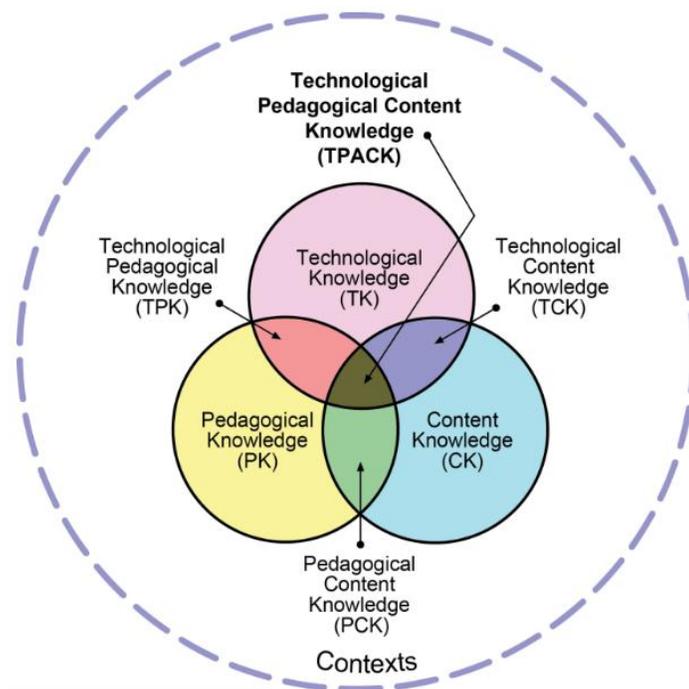


Figura 4.1 - Marco *TPACK* y sus saberes que lo componen³⁷

El modelo *TPACK* se enfoca en tres componentes como se muestra en la Figura 4.1, los cuales son (Koehler y Mishra, 2015):

³⁶ Habilidades en el conocimiento Tecnológico Pedagógico del Contenido *TPACK*, término desarrollado por Mishra y Koehler en el año 2008.

³⁷ Fuente: Tomado del sitio web *TPACK.org*.

Conocimiento sobre el Contenido (CK): Es el saber del docente acerca del contenido que enseña. Conoce, comprende, teorías, conceptos y procedimientos de un área determinada.

Contenido pedagógico (PK): Es el relacionado con el conocimiento de los docentes acerca de la enseñanza y el aprendizaje, sobre comprensión de cómo aprenden los estudiantes, estrategias de manejo de clase, planificación de clases y evaluación.

Conocimiento sobre la tecnología (TK): Es el conocimiento relacionado con la comprensión amplia de la tecnología de la información, de modo que se pueda aplicar productivamente al mundo y a la vida cotidiana, es un tipo de conocimiento que se encuentra en estado de fluidez y para el cual se debe estar en la capacidad de aprender y adaptarse.

Conocimiento pedagógico del contenido (PCK): Está relacionado con el conocimiento pedagógico aplicable a la enseñanza de contenido específico, es cuando el docente transforma el conocimiento de la disciplina para su enseñanza, trabaja sobre la enseñanza, aprendizaje, currículum, evaluación e informes y las condiciones que promueven el aprendizaje y los vínculos entre el currículum, la evaluación y la pedagogía.

Conocimiento sobre el contenido tecnológico (TCK): Los docentes necesitan saber los contenidos de su área y comprender cómo el uso de la tecnología cambia la manera de aprender estos contenidos. De esta manera es importante entender qué tecnologías son adecuadas para el aprendizaje en su disciplina.

Conocimientos tecnológicos pedagógicos (TPK): Se relaciona con conocer las posibilidades y limitaciones de las tecnologías y cómo éstas funcionan en diversos contextos disciplinares, este tipo de conocimientos busca el uso de la tecnología para mejorar los aprendizajes y la comprensión de los estudiantes.

Saberes tecnológicos y pedagógicos del contenido (TPACK): Se refiere a la interacción entre los saberes de contenido, pedagogía y tecnología. Es la base de la enseñanza efectiva con tecnología, es la manera de enseñar contenidos usando la tecnología pedagógicamente, teniendo en cuenta el contexto. Los docentes deben tener habilidades relacionadas con estos tipos de conocimientos, pero además conocer la manera en que estos dominios se interrelacionan de modo que puedan construir propuestas efectivas.

4.2.2. La propuesta de Manso et al. (2011)

Diversos autores vinculados a la integración de TIC en espacios escolares o educativos, en general, presentan diferentes guías, etapas o pasos para ayudar a los docentes a planificar dichos procesos de integración de tecnología, de modo tal que estén articuladas con la enseñanza de los contenidos curriculares y el propio contexto en el que se llevará adelante. En (Manso, Perez, Libedinsky, Light y Garzón, 2011) se plantean cinco pasos que orientan este desarrollo:

- Paso 1: Desarrollar los objetivos de aprendizaje.
- Paso 2: Decidir las estrategias de enseñanza que se van a utilizar.
- Paso 3: Diseñar o seleccionar las actividades de aprendizaje.
- Paso 4: Seleccionar las estrategias de evaluación.
- Paso 5: Seleccionar y articular las herramientas o los recursos TIC para las actividades.

Los pasos propuestos por estos autores coinciden con los tipos de decisiones que se deben tener en cuenta según *TPACK*.

Según en el modelo *TPACK* cuando se diseña una propuesta de trabajo es necesario tomar tres tipos de decisiones en este orden (Magadán, 2012):

Decisiones curriculares:

- Definir el tema o bloque de contenidos seleccionado de acuerdo con el diseño curricular.

- Especificar los objetivos de aprendizaje.

Decisiones pedagógicas:

- Plantear los tipos de actividades que se van a proponer y el producto final que esperamos alcanzar.
- Establecer el rol que cumplirán los docentes y el rol que se espera de los alumnos para llevar adelante la propuesta.
- Contemplar las estrategias de evaluación que se implementarán.

Decisiones tecnológicas:

- Tener en cuenta las necesidades pedagógicas para elegir los recursos digitales.
- Buscar los recursos digitales
- Pautar y prever la utilización de estos recursos

Estos antecedentes han sido considerados para la propuesta de una metodología que permita a partir del uso de herramientas digitales llevar adelante experiencias que promuevan el pensamiento computacional

4.2.3. Modelo de proceso de desarrollo de *software* en cascada

Se han encontrados ciertas similitudes en el ordenamiento en pasos propuesto por Manso et al. (2011), y algunos modelos de procesos de desarrollo de *software*, cuya intención también es planificar y ordenar las tareas involucradas en un proceso de diseño.

Tal es el caso del modelo de proceso para el desarrollo de *software* en cascada, que también sigue una secuencia de pasos, en los cuales se propone la ejecución de un conjunto de actividades. Las actividades a su vez generan salidas que son utilizadas como entradas para el paso subsiguiente, lo cual supone que una actividad debe terminarse en algún grado para empezar la siguiente (Roa, 1985).

Según el autor este modelo tiene las siguientes características:

- Es un modelo de proceso clásico.
- Es el modelo más simple conocido.
- Es sencillo y fácil de entender.
- Para poder pasar a la siguiente fase se tienen que haber conseguido ciertos objetivos de la fase anterior.
- Es un proceso con progresos ordenados y coherentes.

4.3. Metodología para el diseño de experiencias que fomenten el desarrollo del pensamiento computacional en nivel inicial y medio

La metodología que se propone a continuación retoma las propuestas del modelo *TPACK*, los pasos presentados por Manso et al. (2011) y, al mismo tiempo, dado que lo que se propone diseñar a partir de la metodología es un proceso educativo que involucra tecnología, se han considerado algunos aspectos también del modelo de proceso para el desarrollo de *software* en cascada. Estos antecedentes permiten ordenar la metodología que aquí se presenta. La metodología guía la planificación de una iniciativa que integra las TIC en el aula para desarrollar el pensamiento computacional con herramientas digitales. Al mismo tiempo, considera la planificación de la evaluación de la iniciativa diseñada, a partir de la revisión de sus resultados.

Se retoma la idea de etapas y/o pasos tanto del modelo en cascada como del trabajo de Manso et al. (2011). Finalmente, se consideran los aspectos curriculares, pedagógicos y tecnológicos del modelo *TPACK* presentado anteriormente.

Además, la metodología que se presenta busca rescatar las buenas prácticas encontradas a lo largo de las iniciativas analizadas y organizarlas, de manera que ofrezca un acompañamiento para aquellos docentes que deseen planificar una propuesta de tal tipo. La misma se genera a partir de una serie de recomendaciones que guían la toma de decisiones con respecto a las estrategias, actividades y herramientas a utilizar según las necesidades presentes en el entorno educativo en el que se espera aplicar. A continuación, se describe la metodología propuesta.

4.3.1. Descripción detallada de la metodología propuesta

Se describe cada una de las etapas que conforman la metodología junto con preguntas orientadoras que servirán de guía para su desarrollo.

Las primeras etapas que se proponen en esta metodología se basan en los pasos descritos por Manso et al. (2011) y se realizan algunas modificaciones pues en este caso se hace importante además de plantear los objetivos de aprendizaje reconocer el contexto, los destinatarios y las posibilidades tecnológicas para llevar adelante la experiencia. En la Figura 4.2, se presenta todas las etapas de la metodología y las diferentes decisiones involucradas en cada caso, de manera tal de dar al lector un panorama general de la propuesta. En las siguientes secciones se desarrollarán cada una de las etapas en forma detallada.

Etapas 1
Según Manso se plantean los objetivos

Definición de requerimientos
¿Qué voy a hacer?

- ¿Cuáles son las competencias, habilidades y actitudes relacionadas con el pensamiento computacional que se desean desarrollar en la experiencia educativa?
- ¿Cuál es el tema y los contenidos que se esperan abordar?
- ¿De qué tiempo se dispone para llevar adelante la experiencia educativa?
- ¿Qué características generales tiene el grupo de estudiantes que participarán de la experiencia educativa?
- ¿Con qué dispositivos tecnológicos se cuenta en la institución o en el contexto de aplicación para su utilización en la experiencia a desarrollar?
- ¿Qué conocimientos previos debe tener los docentes y estudiantes en relación a la temática abordada y la tecnología a utilizar?

Etapas 2

Según Manso se retoma el tema de estrategias y el diseño de las actividades, junto con la selección y articulación de herramientas TIC

Diseño y aplicación de la experiencia educativa

¿Cómo lo voy a hacer?

- ¿Qué actividades haciendo uso de herramientas informáticas o tecnológicas, se planificarán en la propuesta a trabajar?
- ¿Qué otras actividades complementarias se desarrollarán en la propuesta a trabajar y de qué manera se organizará la secuencia de trabajo?
- ¿En qué momentos y tiempos se realizará cada actividad y la propuesta en su conjunto?
- ¿Cuáles de los conocimientos, habilidades y actitudes planeadas como objetivos se desarrollarán a partir de cada actividad?
- ¿De qué manera se trabajará cada actividad (grupal/ individual/ mixta)?
- ¿Cuál será el rol de los docentes durante cada actividad?
- ¿Cuáles herramientas de software podrían ser pertinentes para la propuesta a implementar y qué características deberían cumplir?
- ¿Qué espacios y equipos tecnológicos se requieren para la implementación de la estrategia pedagógica diseñada?

Etapas 3

Según Manso se retoma el tema de seleccionar las estrategias de evaluación

Evaluación de la propuesta

¿Cómo resultó el trabajo en aula?

- ¿Qué aspectos de la experiencia se evaluarán (alcance de los objetivos, motivación, satisfacción, uso de recursos, tiempos, barreras encontradas, aspectos exitosos)? En caso de decidir medir el alcance de los objetivos, ¿se realizará por cada actividad, en relación a los objetivos que se propone? ¿O en conjunto? ¿Se considerará la secuenciación de las actividades? ¿Las herramientas utilizadas en la actividad fueron aceptadas por los estudiantes y docentes? ¿Los motivaron? ¿Cuáles fueron las actitudes y reacciones?
- ¿Cuáles actividades resultaron de mayor aceptación dentro del grupo de participantes en la experiencia educativa? ¿Qué oportunidades y barreras se encontraron en cada una?

Etapas 4

No se encuentra contemplada en la propuesta Manso, pero se hace importante en esta investigación.

Ajuste
¿Qué debo modificar?

- ¿Qué correcciones cree pertinente realizar en cuanto a la planificación de las actividades a trabajar? (Tiempos, orden en la que se realizaron, cantidad)
- ¿Cuáles serán los ajustes a realizar para cumplir con los objetivos de la propuesta?
- ¿Cómo se diagramará el plan de mejoras?

Figura 4.2 - Etapas de la propuesta metodológica³⁸

4.3.1.1. Definición de requerimientos

En esta etapa se analizan las necesidades y se definen aspectos específicos acerca de factores didácticos, pedagógicos, tecnológicos, y contextuales.

Es importante definir todos los aspectos a tener en cuenta según el entorno en el cual se aplicará la propuesta, de esta manera se reconocerán los recursos con los que se cuentan para poder generar la experiencia educativa.

Para ello se propone tener en cuenta los siguientes interrogantes:

- a) ¿Cuáles son las competencias, habilidades y actitudes relacionadas con el pensamiento computacional que se desean desarrollar en la experiencia educativa?

La iniciativa a diseñar pretende desarrollar competencias relacionadas con el pensamiento computacional, las cuales se hacen importantes en el momento de dar solución a problemas o satisfacer necesidades haciendo uso de herramientas informáticas. Es importante aquí tener en cuenta los conocimientos previos de los estudiantes al respecto. Esta serie de conocimientos, habilidades y actitudes ha sido definida por una variedad de grupos de investigación e investigadores, de las

³⁸ Fuente: elaboración propia, basada en un proceso en cascada para el diseño de software.

cuales se toman algunas que se perciben como las de mayor importancia y se sintetizan en la tabla 4.1

Tabla 4.1 características relacionadas con el pensamiento computacional

Conocimiento, habilidades y competencias relacionadas con el pensamiento computacional
<ul style="list-style-type: none">● Formular problemas posibles de solución mediante una computadora.● Abstractar la información relevante para reconocer los datos principales.● Diseñar algoritmos para llegar de manera ordenada a la solución.● Descomponer datos, procesos (modularizar) o problemas en partes pequeñas y manejables.● Simular modelos de las posibles soluciones antes de ponerlos a prueba.● Implementar los modelos para saber si la solución propuesta es eficiente y efectiva.● Generalizar y transferir ese proceso de solución de problemas a una gran diversidad de estos.● Persistir en el trabajo con problemas difíciles.● Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

Se sugiere tener en cuenta las habilidades³⁹ señaladas en la tabla 4.1 para dar respuesta al interrogante a) planteado en esta etapa.

b) ¿Cuál es el tema y los contenidos que se esperan abordar?

A partir de la indagación de los conocimientos previos sugeridos en el interrogante anterior, se reconoce ahora, desde donde es posible iniciar la temática y los contenidos a tratar, considerando si es necesario abordar el tema desde los conceptos básicos o si es posible iniciar desde un nivel más avanzado. No se debe olvidar que esta metodología plantea el trabajo con estudiantes de nivel educativo inicial y medio. En función de los conocimientos, actitudes y habilidades a desarrollar se seleccionan los contenidos y las actividades que ayudarán a alcanzar los objetivos propuestos.

³⁹ Estas habilidades se definieron después de un estudio concienzudo realizado a diversos documentos de autores sobresalientes en el tema del desarrollo del pensamiento computacional, este estudio se encuentra en el documento: Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica (trabajo final). Universidad Nacional de La Plata, La Plata, Buenos Aires. Sarmiento, M., Gorga, G., y Sanz, C. (2016).

Una vez definidos los objetivos de aprendizaje se puede continuar con el análisis de los demás requerimientos.

c) ¿De qué tiempo se dispone para llevar adelante la experiencia educativa?

Es importante definir el tiempo, días, meses, horas, con el que se cuenta para aplicar la propuesta diseñada y de este modo poder plantear las sesiones de trabajo.

d) ¿Qué características generales tiene el grupo de estudiantes que participarán de la experiencia educativa?

Para poder avanzar en el diseño de la estrategia a trabajar, es importante conocer las características sociales, económicas, educativas, culturales de los estudiantes y del entorno en que se desenvuelven.

Se debe indagar sobre aspectos como la franja etaria, el contexto en el cual conviven, las tecnologías con las que suelen trabajar/entretenerse los estudiantes, entre otros.

e) ¿Con qué dispositivos tecnológicos se cuenta en la institución o en el contexto de aplicación para su utilización en la iniciativa a desarrollar?

Para definir, posteriormente en la etapa de diseño, las herramientas de software posibles de utilizar y su vinculación con las actividades a realizar es importante saber con qué tipos de dispositivos tecnológicos (*PC, Tablet, celulares, etc.*) se cuenta en el aula/institución en la cual se realizará la implementación.

f) ¿Qué conocimientos previos debe tener los docentes en relación a la temática a abordar y la tecnología utilizada?

Es importante identificar los conocimientos previos relacionados con la temática a tratar, de manera tal de planificar si fuera necesario instancias de capacitación previas.

4.3.1.2. Diseño y aplicación de la experiencia educativa

Habiendo pasado por la primera etapa de análisis y teniendo un panorama más detallado de los puntos de partida, se continúa con la siguiente etapa en donde se

definirá la manera en que se organizará la propuesta a implementar. Para ello se propone responder a los siguientes interrogantes:

a) ¿Qué actividades haciendo uso de herramientas informáticas, se planificarán en la propuesta a trabajar?

Se debe planear la estrategia indicando qué actividades se realizarán (actividades de diagnóstico, motivación, autoevaluación, otras), qué recursos digitales se necesitan para realizar estas tareas, con qué habilidades, conocimientos y actitudes del pensamiento computacional se relacionan. También es importante definir si el trabajo es individual o grupal y de qué manera se organizarán las tareas y responsabilidades, en qué momento se utilizará la/s herramienta/s, cuánto duran las actividades, qué resultados se esperan y de qué manera se le hará seguimiento y devolución al proceso del estudiante.

Esto puede ser reflejado en un instrumento del estilo cronograma de la experiencia a desarrollar que detalle las sesiones de trabajo, sus objetivos, las actividades y los materiales y herramientas involucrados en cada una.

b) ¿Qué otras actividades complementarias se desarrollarán en la propuesta a trabajar y de qué manera se organizará la secuencia de trabajo?

Se describirá de manera similar al punto anterior cada una de las actividades a efectuar en las que no se involucra el uso herramientas informáticas. Por ejemplo, ejercicios con el uso de papel y lápiz, revisión de algún tipo de información escrita, observación de material videográfico, entre otras. Se hace necesario enumerar los diferentes materiales educativos a utilizar, su objetivo y su justificación dentro de la propuesta.

Estas también pueden ser aunadas en un cronograma de la experiencia para mostrar la secuencia de actividades en el tiempo.

c) ¿En qué momentos y tiempos se realizará cada actividad y la propuesta en su conjunto?

Teniendo como referencia el tiempo con el que se cuenta para la aplicación de la propuesta educativa, se deben organizar las actividades de manera tal que se

tenga planeado qué sucederá en cada una de las sesiones, e indicar el tiempo que llevará cada una de ellas.

d) ¿Cuáles de los conocimientos, habilidades y actitudes planeadas como objetivos se desarrollarán a partir de cada actividad?

Basados en los conocimientos, habilidades y actitudes que se piensan abordar, es necesario desarrollar actividades enfocadas a lograr dichos objetivos.

e) ¿De qué manera se trabajará cada actividad (grupala/ individual/ mixta)?

De igual manera que en el interrogante anterior, teniendo en cuenta las características a desarrollar se debe planear la manera en que trabajarán los estudiantes, por ejemplo, si las actividades serán grupales o individuales, en clase o no, en qué contexto, etc.

f) ¿Cuál será el rol de los docentes durante cada actividad?

Al definir las actividades que se desarrollarán en cada una de las sesiones, es necesario precisar qué rol tendrá cada uno de los docentes participantes en la implementación.

g) ¿Cuáles herramientas informáticas podrían ser pertinentes para la propuesta a implementar y qué características deberían cumplir?

Es importante tener en cuenta aspectos como el nivel académico, la franja etaria, los contenidos a desarrollar, los conocimientos previos de los estudiantes y docentes que participan del proceso, y los recursos disponibles en la institución o en el contexto de aplicación. Otros aspectos que se podrían considerar están relacionados con la disponibilidad de material de apoyo como pueden ser tutoriales, manuales, foros de ayuda, etc. que orienten en la utilización de esas herramientas.

Como sugerencia de la metodología, en la tabla 4.2, se lista una serie de criterios a considerar para acompañar la decisión de la opción de herramientas más convenientes.

Tabla 4.2 Criterios recomendados en el momento de elegir las herramientas a utilizar

CRITERIOS RECOMENDADOS
C1: Adaptables al contexto de aplicación.
C2: Aptas para el trabajo con estudiantes del grupo objetivo.
C3: Dirigidas al nivel inicial y medio en el desarrollo del pensamiento computacional.
C4: Con interfaz llamativa que motive a los estudiantes
C5: Intuitivas y de fácil comprensión.
C6: Libres y gratuitas.
C7: Con versiones en español o el idioma predominante del grupo objetivo.
C8: Con acceso a material de apoyo preferiblemente de sitios oficiales (manuales, videos, guías, etc.).
C9: Con reconocimiento positivo desde el punto de vista pedagógico y didáctico en el ámbito educativo, en experiencias similares.
C10: Con posibilidades de uso offline (pensado para las instituciones en donde no se cuenta con conectividad).
C11: Adaptables a los requisitos técnicos de los equipos con que se cuentan para la implementación.

Para orientar la selección de las herramientas a utilizar, esta metodología sintetiza en la Tabla 4.3 una selección de herramientas encontradas previamente en la literatura, que han sido utilizadas en diversas experiencias que arrojaron resultados interesantes en término del desarrollo del pensamiento computacional (Sarmiento, Gorga, y Sanz, 2016)

Esta síntesis puede ser un aporte para facilitar la selección antes mencionada.

Tabla 4.3 Herramientas recomendadas

Herramientas recomendadas	
Lenguajes y entornos de programación	Lenguajes de programación visual
<i>Greenfoot</i>	<i>Alice</i>
<i>Gobstones</i>	<i>App Inventor</i>
<i>Pilas Engine</i>	<i>AstroCódigo</i>
<i>PSeInt</i>	<i>E-toys</i>
<i>Python</i>	<i>Game Engine de Blender</i>
<i>Pygame</i>	<i>Kodu</i>
<i>Ruby</i>	<i>LightBot</i>
	<i>Minecraft</i>
	<i>Pilas Bloques</i>
	<i>Proyecto ICARO</i>
	<i>R.I.T.A (Robot Inventor To Teach Algorithms)</i>
	<i>Scratch</i>

h) ¿Qué espacios y equipos tecnológicos se requieren para la implementación de la iniciativa pedagógica diseñada?

Teniendo en cuenta la cantidad de estudiantes y las necesidades previamente indagadas, se deben seleccionar aquellos espacios físicos o virtuales que permitan llevar adelante las actividades planificadas. Al mismo tiempo se debe verificar que estos espacios cumplan los requerimientos técnicos necesarios y permitan trabajar con las herramientas seleccionadas, de no ser así se revisará si deben adecuarse para la aplicación de la propuesta.

4.3.1.3. Evaluación de la propuesta

En esta etapa se planifican las estrategias que se llevarán adelante para medir el alcance de los objetivos que se especificaron inicialmente. Para ello la metodología avanza con las siguientes preguntas orientadoras.

- a) ¿Qué aspectos de la experiencia se evaluarán (alcance de los objetivos, motivación, satisfacción, uso de recursos, tiempos, barreras encontradas, aspectos exitosos)? En caso de decidir medir el alcance de los objetivos, ¿se realizará por cada actividad, en relación a los objetivos que se propone? ¿En su conjunto? ¿Se considerará la secuenciación de las actividades también?

En caso de considerar evaluar la experiencia en cuanto a la motivación y/o satisfacción de los participantes, se pueden considerar test estándares de motivación o adecuarlos, y analizar algunos de los siguientes aspectos: los temas tratados, las actividades realizadas, los programas utilizados, el trabajo realizado con los compañeros, el horario y los tiempos de las sesiones, entre otros.

A partir de estas decisiones se pueden tomar algunas/todas de las siguientes preguntas.

- b) ¿Las herramientas utilizadas en la actividad fueron aceptadas por los estudiantes y docentes? ¿los motivaron? ¿Cuáles fueron las actitudes y reacciones?

Si se hace importante considerar si las herramientas utilizadas cumplieron las expectativas de los participantes, tanto de los estudiantes como de los docentes se puede incluir preguntas o indagaciones al respecto.

c) ¿Cuáles actividades resultaron de mayor aceptación dentro del grupo de participantes en la experiencia educativa? ¿Qué oportunidades y barreras se encontraron en cada una?

Como se dijo anteriormente, es indispensable hacer un seguimiento a los estudiantes, observando qué tipo de actividades resultaron motivadoras y despertaron interés en el trabajo a realizar, revisando en cuáles actividades se presentaron dificultades y cuáles dieron los mejores resultados, de manera que éstas se continúen desarrollando. Aquellas actividades que no tuvieron aceptación o presentaron dificultad deben ser modificadas para una próxima aplicación.

d) En el caso que se desee medir el impacto en el aprendizaje ¿Qué técnicas se planifican para medir el impacto?

Para guiar esta labor se recuperan aquí una serie de técnicas utilizadas ampliamente en el campo de la investigación educativa que puede orientar a los docentes que llevan adelante la experiencia.

Tabla 4.4 Técnicas recomendadas para el seguimiento de las actividades.

Técnicas recomendadas
Preprueba - posprueba y grupo de control
Posprueba únicamente y grupo de control
Cuatro grupos de Solomon

Según Sampieri et al. (2010) las características de estas técnicas son las siguientes:

Preprueba - posprueba y grupo de control: en este diseño se aplican prepruebas a los grupos que componen el experimento. Los participantes se asignan al azar a los grupos, después a éstos se les aplica simultáneamente la preprueba. Luego un grupo recibe el tratamiento experimental y otro no (grupo de control); por último, se vuelve a aplicar simultáneamente, una posprueba.

Posprueba únicamente con grupo de control: Este diseño incluye dos grupos, uno recibe el tratamiento experimental y el otro no (grupo de control). Los

participantes se asignan a los grupos de manera aleatoria. Cuando concluye el tratamiento experimental, a ambos grupos se les aplica la posprueba. La posprueba debe administrarse inmediatamente después de que concluya el experimento y debe aplicarse de manera simultánea a ambos grupos.

Cuatro grupos de Solomon: Es una combinación de las propuestas dos anteriores. Se puede trabajar con cuatro grupos: dos experimentales y dos de control. Los primeros reciben el mismo tratamiento experimental y los segundos no reciben tratamiento. Sólo a uno de los grupos experimentales y a uno de los grupos de control se les administra la preprueba, luego del tratamiento experimental, a los cuatro grupos se les aplica la posprueba. Los participantes se asignan en forma aleatoria.

Estas son algunos ejemplos, pero estas sugerencias no resultan para nada exhaustivas, sino más bien ejemplos de técnicas a utilizar.

Para la recopilación de datos también podrían ser utilizadas técnicas tales como: encuestas, entrevistas, cuestionarios, y observación directa.

4.3.1.4. Ajuste

Es una etapa centrada en los cambios que se puedan realizar para mejorar el proceso y resultados de la propuesta diseñada.

Teniendo en cuenta la fase anterior y después de revisar los resultados en la aplicación de la propuesta, es importante decidir qué cambios serían posibles de realizar, de manera que la propuesta pueda evolucionar para una próxima implementación. Es decir, se planifica la elaboración de acciones de mejoras de manera tal que la experiencia pase a integrarse de manera más efectiva en el contexto educativo para la que fue diseñada.

Cuando se elabore el plan de mejoras se revisarán algunas de las decisiones abordadas en las etapas de análisis y diseño previamente descritas en función de los resultados obtenidos. Por ejemplo, se podrían considerar:

- a. ¿Qué correcciones cree pertinente realizar en cuanto a la planificación de las actividades a trabajar? (Tiempos, orden en la que se realizaron, cantidad)

- b. ¿Cuáles serían los ajustes a realizar para cumplir con los objetivos de la propuesta?
- c. ¿Qué cambios en el contexto, herramientas y decisiones en general se requieren?

4.4. Conclusiones

El capítulo 4 hace referencia a la metodología propuesta, que busca promover el diseño de una experiencia educativa orientada al desarrollo de pensamiento computacional. La metodología involucra una serie de preguntas orientadoras, agrupadas en etapas que ayudan a ordenar las decisiones propias de una planificación educativa con tecnología digital.

Se pretende que la metodología propuesta ofrezca una alternativa apropiada para el diseño de prácticas educativas, donde se organicen las actividades a trabajar, se seleccionen las herramientas a utilizar, se planee la manera en que se evaluará y se hará seguimiento a los participantes. De este modo se presenta una alternativa que puede ser utilizada por docentes interesados en desarrollar habilidades, conocimientos y actitudes relacionadas con el pensamiento computacional, en nivel primario y secundario.

Como aspectos particulares de la metodología se realizan recomendaciones y orientaciones *ad-hoc* tanto de los tipos de habilidades, conocimientos y actitudes vinculadas al pensamiento computacional relevadas en la literatura, así como de las herramientas mayormente utilizadas en el marco de este tipo de experiencias.

En el próximo capítulo se presenta un estudio de caso en el que utiliza la metodología propuesta aquí de manera tal de responder a los siguientes interrogantes:

- a. ¿Qué preguntas orientadoras no fueron contempladas y se hacen necesarias en el momento de la planificación?
- b. ¿En qué aspectos ayudó la metodología para llevar adelante una experiencia para el desarrollo del pensamiento computacional y su evaluación?

- c. ¿Qué barreras se encontraron a lo largo del proceso de diseño e implementación de la experiencia que merezcan ser atendidas desde la propuesta metodológica?

Estas preguntas serán retomadas como parte de la aplicación y evaluación de la propuesta metodológica que se pondrá en juego en un estudio de caso realizado en un contexto educativo específico.

Capítulo 5

Estudio de caso

5.1. Introducción

En este capítulo se presenta un estudio de caso en el que se pone en juego la metodología propuesta en el capítulo previo, siguiendo las etapas para la planificación de experiencias en el aula que pretenden desarrollar conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional.

El estudio de caso aborda la realización de una iniciativa educativa en la Institución Educativa Departamental Pío XII, del municipio de Pacho, Departamento de Cundinamarca, Colombia. En particular, se trabaja con estudiantes de sexto grado de básica secundaria, pertenecientes al Club de lectura llamado “Cuenta un cuento con tecnología” dirigido por una docente del área de Lengua Castellana, con el apoyo de la tesista en la parte tecnológica.

El estudio de caso propone definir juicios de valor que sirvan para tomar decisiones relacionados con los aspectos que se deben mejorar o mantener con respecto a la metodología, y al mismo tiempo, permita fortalecer el desarrollo de habilidades computacionales de los estudiantes de la institución seleccionada.

5.2. Descripción general del estudio de caso

El caso que se estudia es la aplicación de la metodología descrita en el capítulo anterior para el diseño de actividades áulicas orientadas al desarrollo del pensamiento computacional. Este estudio pretende conocer los alcances y limitaciones de la metodología propuesta en el momento de ser utilizada y aplicada por un docente. Asimismo, se examina si la metodología orienta a que las experiencias diseñadas y realizadas a partir de ésta, fomenten el desarrollo del pensamiento computacional en los estudiantes participantes de la iniciativa.

Al evaluar el estudio de caso se analizan estos dos aspectos: limitaciones y posibilidades de la metodología desde el punto de vista de los docentes y logro de los objetivos propuestos para los estudiantes.

5.2.1. Diseño del estudio de caso

Para iniciar el proceso de diseño del estudio de caso se seleccionó una docente que estuviera interesada en hacer un trabajo transversal entre su área y el área de tecnología, también en hacer uso de las TIC en pro del desarrollo de características tanto de su área como del área tecnológica y que contara con tiempo para llevar a cabo sesiones de trabajo para el diseño de las actividades y su implementación. Al contar con la docente interesada, que en este caso es la docente de Lengua Castellana, se realizaron dos sesiones de dos horas cada una en las cuales se acordó la temática a trabajar, se dialogó y se explicó la metodología propuesta para el diseño de actividades con tecnología, llegando a acuerdos con respecto a las competencias por desarrollar, relacionadas tanto con el pensamiento computacional como con el área de Lengua Castellana.

Para poder analizar la metodología propuesta se diseñaron instrumentos de recolección de datos como la entrevista semi estructurada que fue aplicada a la docente tanto en el momento del diseño de actividades como al finalizar la implementación de éstas. Además, se realizó la observación y el registro de las sesiones trabajadas con la docente para el diseño de las actividades.

El seguimiento a los avances de la investigación se hizo a través de un diseño pretest – postest de un solo grupo, para lo cual se utilizó instrumentos de recolección de información con el fin de analizar los alcances en el desarrollo de las habilidades, conocimientos y actitudes relacionadas con el pensamiento computacional que se propusieron abordar en las sesiones trabajadas en aula con los estudiantes. Estos instrumentos son: Preprueba, Entrevistas 1 y 2, Prueba Likert 1 y 2. *Focus group* 1 y 2, registro audiovisual y postprueba al trabajo de las sesiones (Ver Anexo II).

En las dos sesiones trabajadas con la docente de Lengua Castellana, se tomaron las preguntas orientadoras propuestas por la metodología como base para el diseño de la experiencia, centrándose en la toma de decisiones con respecto a la etapa 1: definición de requerimientos y etapa 2: diseño y aplicación de la experiencia educativa, de modo que las decisiones tomadas fueran pertinentes para el trabajo con el grupo de estudiantes en el aula (Ver 5.3).

Después de diseñar la experiencia educativa, se trabajó en su implementación con los estudiantes del Club de lectura en siete sesiones de dos horas cada una, durante los meses de mayo, junio y julio de 2018.

Para finalizar, se realizó un análisis de los resultados del estudio de caso tomando la información recopilada por medio de los instrumentos diseñados para este fin, de este modo se logra dar los resultados y las conclusiones obtenidas.

5.2.2. Descripción de los participantes estudiantes

Con respecto a los estudiantes que participan en la experiencia, se definieron mediante un muestreo intencional⁴⁰, seleccionando algunos niños del grado sexto de básica secundaria, los cuales cuentan con permiso de participación firmada por sus acudientes⁴¹ y habitan cerca de la Institución de manera que puedan cumplir con el horario en que se realizan las sesiones.

La experiencia se implementa en horario extraescolar, con reuniones de dos horas semanales mediante un espacio académico llamado Club de lectura “Cuenta un cuento con tecnología”. El club está conformado por 16 estudiantes.

Como se dijo anteriormente este estudio de caso se realiza con la participación de una docente en el área de Lengua Castellana, interesada en implementar la metodología propuesta para el diseño de actividades áulicas, en las cuales los estudiantes utilizan entornos de programación para la creación y muestra de textos literarios, con el fin de promover el desarrollo de competencias, habilidades y actitudes que se referirán más adelante. A continuación, se describen detalladamente algunas características que son importantes para reconocer aspectos relacionados con el entorno en el cual los estudiantes conviven.

La franja etaria de los estudiantes está entre 10 y 12 años, 2 estudiantes de 10 años, 13 de 11 años y 1 de 12 años, el total de estudiantes que forman parte del club de lectura son 16 estudiantes (ver figura 5.1).

⁴⁰ Es el muestreo realizado de forma deliberada con un propósito (Punch 1998 citados en Izcara, 2007)

⁴¹ El formato de consentimiento informado para la participación de los estudiantes y formato de ficha personal se puede ver en el Anexo I.

La mayoría de los estudiantes (14 estudiantes) viven en zona urbana, los demás (2 estudiantes) viven en zona rural (ver figura 5.2).

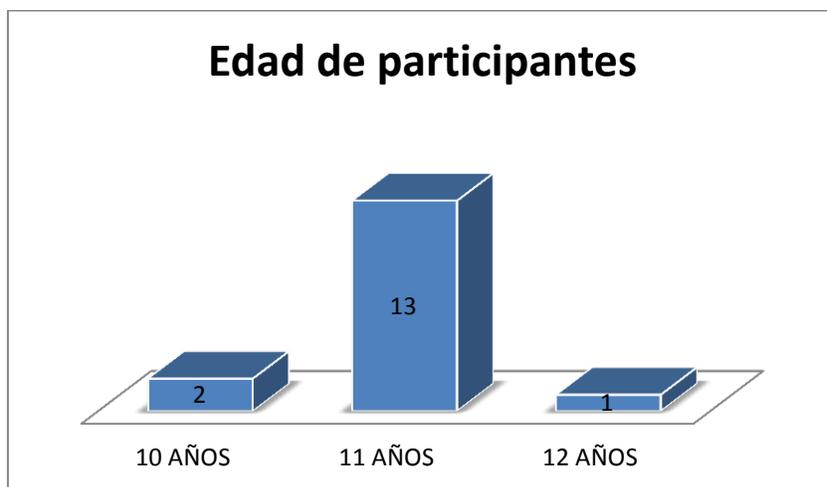


Figura 5.1 - Edad de los participantes

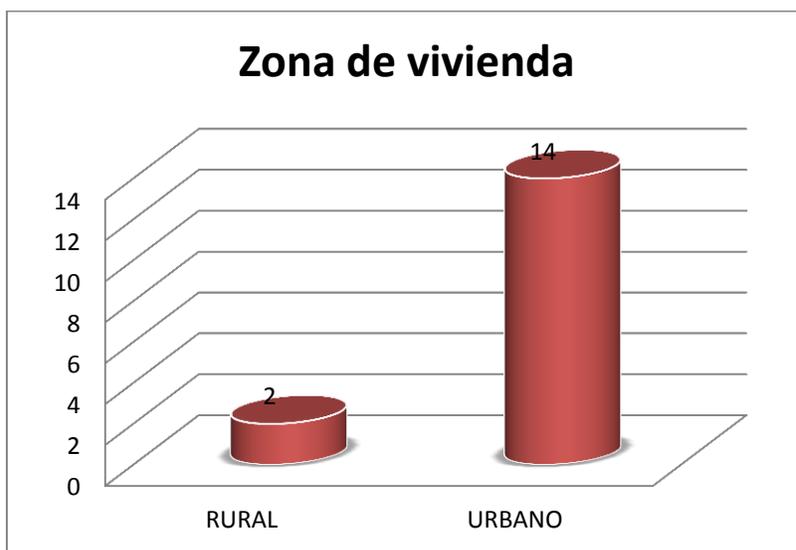


Figura 5.2 - Zona de vivienda de los participantes

Dentro de la Institución Educativa los estudiantes cuentan con una sala de informática dotada con 20 computadoras portátiles y un video *beam*. No se dispone de servicio de Internet.

Aunque la iniciativa se realiza dentro de la institución utilizando los equipos de la sala de informática, es importante conocer los medios tecnológicos que tienen

los estudiantes a su alcance, de esta manera saber el grado de familiaridad con que hacen uso de algunos dispositivos como el celular, la *Tablet* y las PC (ver figura 5.3).

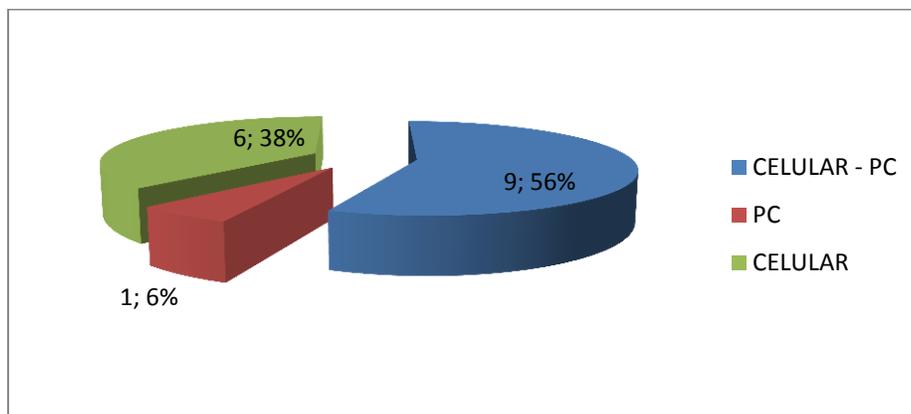


Figura 5.3 - Zona de vivienda de los participantes

Con respecto a los dispositivos con que cuentan los participantes, 9 estudiantes tienen celular y PC, 6 estudiantes únicamente tienen celular y 1 estudiante tiene sólo PC, todos cuentan con algún tipo de dispositivo. En cuanto a conectividad, 1 de ellos no cuentan con Internet en su casa, 9 de ellos tienen Internet móvil y los otros 6 Internet domiciliario (ver figura 5.4).

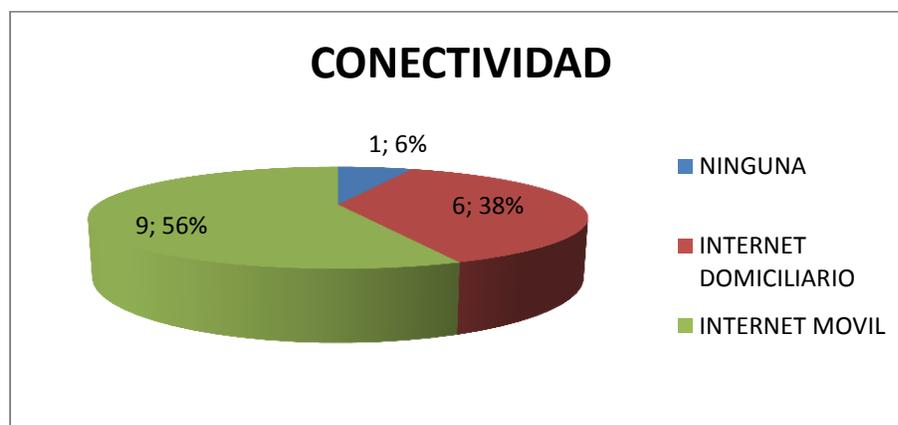


Figura 5.4 - Conectividad con que cuentan los participantes

Por otro lado, se hace importante saber qué conocimientos previos tienen los estudiantes con respecto al uso de entornos de programación, con el objetivo de

poder planear las actividades teniendo en cuenta el nivel de conocimiento de los estudiantes con respecto a estos (ver figura 5.5).

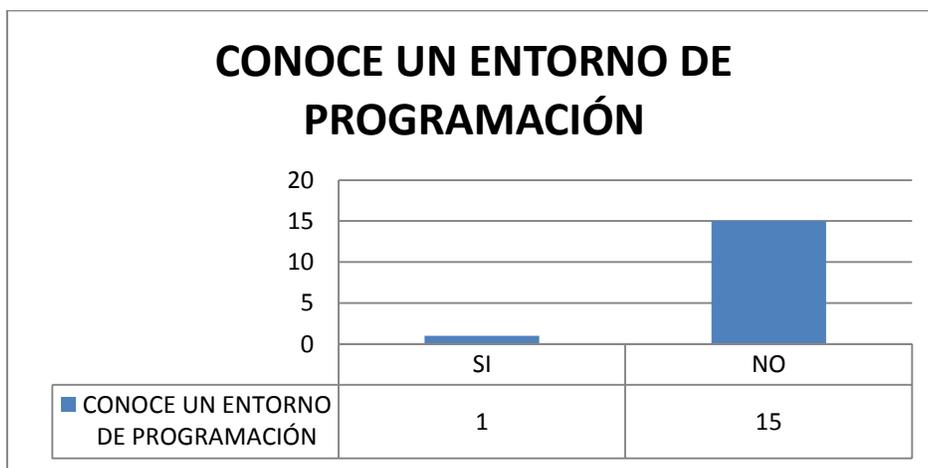


Figura 5.5 - Reconocimiento de entornos de programación

La mayoría de los estudiantes (15 estudiantes) no conocen ningún entorno de programación, 1 de ellos reconocen entornos como *Kodu*, *Pilas Bloques* y *Scratch*.

5.2.3. Descripción del perfil de la docente interviniente

La profesora Diana Ahumada nació en el municipio de Pacho, tiene 30 años, es docente de Lengua Castellana en la Institución Educativa Departamental Pío XII, Licenciada en Educación Básica con énfasis en Lengua Castellana de la Universidad Pedagógica y Tecnológica, con alrededor de 9 años de experiencia, aunque no ha realizado estudios específicos en el área tecnológica es una persona interesada por las TIC y cercana a los dispositivos, utiliza con facilidad el celular, la *Tablet* y la *PC*, conoce programas como el paquete de *Office* y algunos *software* educativos. Como experiencia en el diseño de actividades con tecnología, la profesora propone el uso de *Tablet* en sus clases dentro del horario escolar usando aplicaciones como *JClic* y *Hot Potatoes* para trabajar temáticas específicas en su área.

5.2.4. Descripción breve del contexto en el que se desarrolla el caso

La experiencia educativa se desarrolla en el municipio de Pacho, Cundinamarca, Colombia, municipio distante a 88 Km de Bogotá D.C, el municipio se encuentra en la región del Rionegro a una altitud de 1905 msnv, con un clima promedio de 18° C, el nivel socio económico de los estudiantes es variado, la economía de región está basada en la agricultura de cítricos (naranja, mandarina, entre otros) y ganadería en pequeña escala.

La Institución Educativa Departamental Pío XII, es la más grande y antigua de la región con alrededor de 100 años de existencia, cuenta con diversas sedes tanto de primaria como de secundaria a la cual asisten en el año 2018 cerca de 1.300 estudiantes.

5.3. Aplicación de la metodología en el estudio de caso

A continuación, se dan a conocer las decisiones tomadas para el diseño de la experiencia educativa por la docente de Lengua Castellana Diana Ahumada a partir de los pasos planteados por la metodología.

Etapas 1. Definición de requerimientos

- ¿Cuáles son los conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional que se desean desarrollar en la experiencia educativa?

Los conocimientos, habilidades y actitudes a desarrollar en las sesiones son:

- Formular problemas posibles de solución mediante una computadora.
- Abstracter la información relevante para reconocer los datos principales.
- Diseñar algoritmos para llegar de manera ordenada a la solución.
- Descomponer datos, procesos (modularizar) o problemas en partes pequeñas y manejables.
- Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

Además de estas competencias también se propone trabajar otras relacionadas con el área de Lengua Castellana como son las habilidades lectoescritoras de los estudiantes.

- ¿Cuál es el tema y los contenidos que se esperan abordar?

Los temas a abordar serán:

Creación literaria: el cuento, cohesión, párrafo, coherencia, lectura de imágenes, expresión oral.

Programación con bloques de arrastrar y soltar: lógica de programación, bloques de comando, repetición y alternativa condicional.

- ¿De qué tiempo se dispone para llevar adelante la experiencia educativa?

Se dispone de dos horas semanales fuera del horario escolar, durante el tiempo que se requiera. La experiencia propone trabajar siete sesiones, una sesión semanalmente.

- ¿Qué características generales tiene el grupo de estudiantes que participarán de la experiencia educativa?

El grupo de estudiantes con el cual se va a trabajar son estudiantes de grado sexto de la Institución Educativa Departamental Pío XII, pertenecientes al Club “Cuenta un cuento con tecnología” cuya edad esta entre 10 y 12 años. En la sección 5.2.2 puede verse un análisis detallado llevado a cabo por la docente y la tesista.

- ¿Con qué dispositivos tecnológicos se cuenta en la institución o en el contexto de aplicación para su utilización en la experiencia a desarrollar?

En la sala de informática de la Institución Educativa Departamental Pío XII, sitio donde se realizará la implementación se cuenta con 20 computadoras portátiles y un Video *Beam*.

- ¿Qué conocimientos previos debe tener los docentes y estudiantes en relación a la temática abordada y la tecnología a utilizar?

En este caso donde la experiencia educativa es trabajada transversalmente, se necesita contar con un experto en el área de Lengua castellana y otro en el área de tecnología, además los dos docentes deben tener conocimientos básicos del área contraria.

Con relación a los conocimientos previos de los estudiantes, en el área de Lengua Castellana deben manejar conceptos básicos de escritura y lectura, junto con un uso básico de las computadoras.

Etapa 2. Diseño y aplicación de la experiencia educativa

- ¿Qué actividades haciendo uso de herramientas informáticas o tecnológicas, se planificarán en la propuesta a trabajar?

Se realizarán actividades en equipo de dos estudiantes que harán uso de las computadoras y el entorno de programación elegido, utilizando bloques de comandos, repeticiones y condicionales. Para realizar las creaciones literarias los participantes podrán escribirlas haciendo uso de procesadores de texto, esta información se podrá revisar detalladamente en las tablas 5.1 a 5.7.

- ¿Qué otras actividades complementarias se desarrollarán en la propuesta a trabajar y de qué manera se organizará la secuencia de trabajo?

La actividad complementaria realizada por los estudiantes está relacionada con la creación literaria escrita a lápiz en papel, donde por medio de la lectura de imágenes y la abstracción de información de las escenas trabajadas en los entornos de programación se crea un cuento que luego será compartido a partir de su lectura en voz alta.

- ¿En qué momentos y tiempos se realizará cada actividad y la propuesta en su conjunto?

Las actividades están organizadas por medio de sesiones de dos horas en las cuales se trabajan conjuntamente las competencias relacionadas con las dos áreas del conocimiento, esta información se podrá revisar detalladamente en las tablas 5.1 a 5.7.

- ¿Cuáles de los conocimientos, habilidades y actitudes planeadas como objetivos se desarrollarán a partir de cada actividad?

Cada una de las actividades busca cumplir con el objetivo de desarrollar los conocimientos, habilidades y actitudes que se propone abordar desde las dos áreas del conocimiento, esta descripción se encuentra detallada en las tablas 5.1 a 5.7.

- ¿De qué manera se trabajará cada actividad (grupal/ individual/ mixta)?

Las actividades se realizarán grupalmente, desarrollando los ejercicios y los desafíos en equipo de dos estudiantes que compartirán sus creaciones literarias con el grupo en general.

- ¿Cuál será el rol de los docentes durante cada actividad?

El rol del docente será el de explicar, asesorar y ofrecer pautas para realizar y mejorar las soluciones dadas a las actividades planeadas en cada sesión.

La docente de tecnología se hará cargo de guiar las actividades relacionadas con los entornos de programación y apoyada por la docente de Lengua Castellana, asimismo la docente de Lengua Castellana se hará cargo de las actividades de lectura, escritura y creación de textos literarios, apoyada por la docente de tecnología.

- ¿Cuáles herramientas de *software* podrían ser pertinentes para la propuesta a implementar y qué características deberían cumplir?

Después de revisar los criterios y las herramientas propuesta en la metodología, se eligen los entornos de programación Pilas Bloques, *Minecraft* hora del código y *Scratch*, puesto que cumplen con los criterios recomendados y pueden ser trabajados en los dispositivos con que cuenta la Institución. Algunas características por las que fueron elegidas estas herramientas son:

Scratch se encuentra instalado en las computadoras como parte de las herramientas educativas dispuestas en estos dispositivos que fueron donados por el Ministerio de Educación Nacional. Las tres herramientas son libres y gratuitas, intuitivas y de fácil comprensión, con versiones en español y con requerimientos técnicos que cumplen las computadoras con que se cuentan. Además, son reconocidas en el ámbito educativo y académico.

- ¿Qué espacios y equipos tecnológicos se requieren para la implementación de la estrategia pedagógica diseñada?

Se requiere un aula con espacio para aproximadamente 20 personas y un mínimo de 10 computadoras, un TV o Video *Beam*, y un juego de parlantes.

A continuación, se presentan de manera organizada y detallada, las siete sesiones trabajadas con los estudiantes.

Tabla 5.1 Sesión 1. Trabajo con los estudiantes

SESIÓN 1						
TEMA	ESTRATEGIA	TIEMPO	COMPETENCIAS A TRABAJAR	ROL DOCENTE	DEL	FORMAS DE TRABAJO
Introducción a la sesión	Dialogo charla introductoria	10 min		Introducir el tema a tratar en la sesión		
Exploración entorno de programación	Los estudiantes interactúan con el entorno de programación (Pilas Bloques) dando respuesta a los desafíos presentes, utilizando comandos.	40 min	Abstraer la información relevante. Diseñar algoritmos.	Asesorar y ofrecer pautas para llevar acabo la actividad.		Equipo de dos estudiantes
Creación literaria	Teniendo en cuenta una escena de los desafíos propuestos anteriormente, los estudiantes crean colectivamente el inicio de un cuento.	30 min	Leer a partir de imágenes. Escribir con párrafos, apelar a la cohesión y coherencia. Abstraer la información relevante. Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.	Hacer seguimiento, dar sugerencias y orientar a cada uno de los grupos.		Equipo de dos estudiantes
Expresión oral	Cada uno de los equipos lee su creación literaria en voz alta y al finalizar la docente realiza la retroalimentación respectiva.	30 min	Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común Expresarse en forma oral.	Retroalimentar y promover la comprensión y retención del conocimiento		Equipo de dos estudiantes
Entrevista	Se evaluará la sesión realizando una entrevista a 3 participantes	10 min		Charla a la cual se le hará una grabación de voz		Prueba individual

Tabla 5.2 Sesión 2. Trabajo con estudiantes

SESIÓN 2					
TEMA	ESTRATEGIA	TIEMPO	COMPETENCIAS A TRABAJAR	ROL DEL DOCENTE	FORMAS DE TRABAJO
Introducción a la sesión	Dialogo charla introductoria	5 min		Introducir el tema a tratar en la sesión	
Trabajo con el entorno de programación, uso de bloques de repetición	Los estudiantes interactúan con el entorno de programación (Pilas Bloques) solucionando los desafíos presentes donde se utilizan bloques de repetición.	40 min	Abstraer la información relevante. Diseñar algoritmos.	Asesorar, ofrecer pautas y guiar la actividad	Equipo de dos estudiantes
Creación literaria	Teniendo en cuenta una escena de los desafíos trabajados, los estudiantes continúan con la creación colectiva redactando el nudo del cuento.	40 min	Leer a partir de imágenes. Escribir párrafos, apelar a la cohesión y coherencia. Abstraer la información relevante. Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.	Hacer seguimiento, realizar sugerencias	Equipo de dos estudiantes
Expresión oral	Cada uno de los equipos lee su creación literaria de inicio y nudo en voz alta, al finalizar la docente realizará la retroalimentación respectiva	30 min	Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común. Expresarse en forma oral.	Retroalimentar y promover la comprensión y retención del conocimiento	Equipo de dos estudiantes
Cierre de actividad	Se realiza un <i>Focus Group</i> de 5 minutos para hablar sobre el trabajo realizado en la sesión	5 min			Todos los participantes

Tabla 5.3 Sesión 3. Trabajo con estudiantes

SESIÓN 3					
TEMA	ESTRATEGIA	TIEMPO	COMPETENCIAS A TRABAJAR	ROL DEL DOCENTE	FORMAS DE TRABAJO
Introducción a la sesión	Diálogo charla introductoria	5 min		Introducir el tema a tratar en la sesión	
Trabajo con bloques de alternativa condicional	Los estudiantes interactúan con el entorno de programación (<i>Minecraft</i> hora del código) dando respuesta a los desafíos que utilicen bloques de alternativa condicional	35 min	Abstraer la información relevante. Diseñar algoritmos.	Ofrecer pautas y guiar la actividad que se lleva a cabo.	Equipo de dos estudiantes
Creación literaria	Teniendo en cuenta una escena de los desafíos trabajados, los estudiantes escriben en creación colectiva el desenlace del cuento.	35 min	Leer a partir de imágenes. Escribir párrafos, apelar a la cohesión y coherencia. Abstraer la información relevante Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común	Asesorar, hacer seguimiento, sugerir y orientar a cada uno de los grupos	Equipo de dos estudiantes
Expresión oral	Cada uno de los equipos cuenta su cuento y lee el desenlace en voz alta, al finalizar la docente realiza la retroalimentación	35 min	Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común. Expresarse en forma oral.	Retroalimentar y promover la comprensión y retención del conocimiento.	Equipo de dos estudiantes
Aplicación de prueba tipo Likert	Para evaluar la sesión los estudiantes contestan una prueba tipo Likert	10 min		Ofrecer pautas para resolver correctamente la prueba.	Prueba individual

Tabla 5.4 Sesión 4. Trabajo con estudiantes

SESIÓN 4					
TEMA	ESTRATEGIA	TIEMPO	COMPETENCIAS A TRABAJAR	ROL DEL DOCENTE	FORMAS DE TRABAJO
Introducción a la sesión	Dialogo charla introductoria.	10 min		Introducir el tema a tratar en la sesión.	
Creación literaria	Basado en los ejercicios de las sesiones anteriores, y pensando en hacer uso de bloques de comando, repeticiones y condicional los estudiantes escriben la primera parte de un cuento con temática libre.	50 min	<p>Escribir párrafos, apelar a la cohesión y coherencia.</p> <p>Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.</p> <p>Descomponer datos, procesos o problemas en partes pequeñas y manejables.</p>	Asesoría, seguimiento, sugerencias y orientaciones.	Equipo de dos estudiantes
Trabajo en el entorno de programación <i>Scratch</i>	Los estudiantes inician la animación de la primera parte del escrito, haciendo uso del entorno de programación <i>Scratch</i> .	50 min	<p>Abstraer la información relevante.</p> <p>Formular problemas posibles de solución mediante una computadora.</p> <p>Abstraer la información relevante.</p> <p>Diseñar algoritmos.</p> <p>Descomponer datos, procesos o problemas en partes pequeñas y manejables.</p>	Ofrecer pautas y guiar la actividad que se lleva a cabo.	Equipo de dos estudiantes
Entrevista	Se evaluará la sesión realizando una entrevista a 3 participantes	10 min		Charla a la cual se le hará una grabación de voz	Prueba individual

Tabla 5.5 Sesión 5. Trabajo con los estudiantes

SESIÓN 5					
TEMA	ESTRATEGIA	TIEMPO	COMPETENCIAS A TRABAJAR	ROL DEL DOCENTE	FORMAS DE TRABAJO
Introducción a la sesión	Dialogo charla introductoria.	5 min		Introducir el tema a tratar en la sesión.	
Creación literaria	En esta sesión, los estudiantes escriben el nudo del cuento con temática libre, pensando en hacer uso de bloques de comando, repeticiones y condicional	55 min	<p>Escribir párrafos, apelar a la cohesión y coherencia.</p> <p>Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común</p> <p>Descomponer datos, procesos o problemas en partes pequeñas y manejables.</p>	Hacer seguimiento, sugerir y orientar	Equipo de dos estudiantes
Trabajo en el entorno de programación <i>Scratch</i>	Los estudiantes continúan el diseño de su animación, tomando el escrito realizado.	55 min	<p>Abstraer la información relevante</p> <p>Formular problemas posibles de solución mediante una computadora.</p> <p>Abstraer la información relevante.</p> <p>Diseñar algoritmos.</p> <p>Descomponer datos, procesos y problemas en partes pequeñas y manejables.</p>	Ofrecer pautas y guía la actividad que se lleva a cabo.	Equipo de dos estudiantes
Cierre de actividad	Se realiza un <i>Focus Group</i> de 5 minutos para hablar sobre el trabajo realizado en la sesión	5 min			Todos los participantes

Tabla 5.6 Sesión 6. Trabajo con estudiantes

SESIÓN 6					
TEMA	ESTRATEGIA	TIEMPO	COMPETENCIAS A TRABAJAR	ROL DEL DOCENTE	FORMAS DE TRABAJO
Introducción a la sesión	Dialogo charla introductoria.	5 min		Introducir el tema a tratar en la sesión.	
Creación literaria	En esta sesión, los estudiantes escriben el desenlace del cuento con temática libre. pensando en hacer uso de bloques de comando, repeticiones y condiciona	55 min	<p>Escribir párrafos, apelar a la cohesión y coherencia.</p> <p>Descomponer datos, procesos o problemas en partes pequeñas y manejables.</p> <p>Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común</p>	seguimiento sugerir, orientar y cuestionar	Equipo de dos estudiantes
Trabajo en el entorno de programación <i>Scratch</i>	Los estudiantes continúan el diseño de su animación, trabajando sobre el desenlace del cuento.	55 min	<p>Abstraer la información relevante.</p> <p>Formular problemas posibles de solución mediante una computadora.</p> <p>Diseñar algoritmos.</p> <p>Descomponer datos, procesos y problemas en partes pequeñas y manejables.</p>	Ofrecer pautas y guía la actividad que se lleva a cabo.	Equipo de dos estudiantes
Cierre de actividad	Para evaluar la sesión los estudiantes contestan una prueba tipo Likert	5 min			Todos los participantes

Tabla 5.7 Sesión 7. Trabajo con estudiantes

SESIÓN 7					
TEMA	ESTRATEGIA	TIEMPO	COMPETENCIAS A TRABAJAR	ROL DEL DOCENTE	FORMAS DE TRABAJO
Introducción a la sesión	Dialogo charla introductoria.	5 min		Introducir el tema a tratar en la sesión.	
Revisión de la animación	Las docentes revisan la animación realizada por los estudiantes en las sesiones anteriores y orientan sus avances y las correcciones necesarias	45 min	Formular problemas posibles de solución mediante una computadora. Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común	Revisar y orientar los cambios necesarios en la animación realizada	Equipo de dos estudiantes
Trabajo en el entorno de programación <i>Scratch</i>	Los estudiantes terminarán el diseño de su animación, mostrando la creación literaria de principio a fin.	60 min	Formular problemas posibles de solución mediante una computadora. Abstraer la información relevante. Diseñar algoritmos. Descomponer datos, procesos y problemas en partes pequeñas y manejables.	Ofrecer pautas y guiar la actividad que se lleva a cabo.	Equipo de dos estudiantes
Cierre de la actividad	Se reflexiona sobre el trabajo realizado en las sesiones, se evalúa por medio de una postprueba el avance del grupo de estudiantes	10 min		Hacer preguntas reflexivas que le permitan al estudiante reconocer y evaluar sus nuevos aprendizajes	Grupo completo

De esta manera se organizaron las actividades que forman parte de la experiencia educativa diseñada para el trabajo con estudiantes en el aula. Algunos detalles de las actividades realizadas se comparten por medio del blog <https://cuentacuentotecno.blogspot.com/>, igualmente la creaciones literarias plasmadas por medio del entorno de programación Scratch se dejan disponibles en la comunidad Scratch <https://scratch.mit.edu/> de manera que la comunidad educativa conozca los trabajos realizados por los estudiantes que participan en el Club de lectura “Cuenta un cuento con tecnología”.

5.4. Conclusiones

El capítulo 5 presenta un estudio de caso en donde se pone en juego la metodología diseñada y descrita en el capítulo 4, por medio de la cual se diseña y se realiza una experiencia educativa aplicada en la Institución Educativa Departamental Pío XII en el municipio de Pacho, Cundinamarca, Colombia. Los participantes son un grupo de estudiantes que conforman el Club de lectura “Cuenta un cuento con tecnología”. Esta implementación busca poner a prueba la metodología diseñada con el fin de mejorar y/o mantener acciones que promuevan el desarrollo de conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional.

De igual manera, en este capítulo se describe el perfil de la docente participante y la manera en que interactuó con la metodología propuesta para llegar a acuerdos con relación a la temática y las actividades áulicas a implementar. También se describen aspectos relacionados con los estudiantes participantes y el contexto en donde se realiza la experiencia.

Por último, el capítulo presenta las actividades a partir del trabajo con la metodología propuesta. Se especifican así las competencias a trabajar, los temas, roles, tiempos y objetivos que se buscan con cada una de las sesiones planificadas, junto con los instrumentos y técnicas de recolección de información que servirán para analizar los alcances y limitaciones de la metodología.

Capítulo 6

Análisis y resultados

6.1. Introducción

Esta tesis llevó adelante un estudio de caso que se encuentra descrito en el capítulo 5, el cual puso a prueba la aplicación de una metodología para el diseño de actividades de aula que pretende promover el desarrollo del pensamiento computacional.

Al hacer uso de la metodología se diseñaron siete sesiones de trabajo con estudiantes, en las cuales se abordaron las características que se pretendían desarrollar en los participantes de un espacio académico denominado Club de lectura “Cuenta un cuento con tecnología”.

Como se comentó en el capítulo 5, el seguimiento de esta investigación se llevó a cabo a través de un diseño pretest – postest de un solo grupo. Para recolectar la información se utilizaron los siguientes instrumentos:

- Preprueba: prueba con escala de *Likert* con afirmaciones concernientes al uso de conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional. Aplicada antes de iniciar las siete sesiones de trabajo.
- Entrevista 1 y 2: entrevistas semiestructurada donde se indagó sobre las opiniones del trabajo realizado después de las sesiones 1 y 4.
- Prueba con escala de *Likert* 1 y 2: pruebas de opinión con respecto al trabajo realizado en las sesiones 3 y 6.
- *Focus group* 1 y 2: diálogo de opinión con respecto al trabajo realizado en las sesiones 2 y 5.
- Registro audiovisual: videos registrados en cada una de las sesiones.
- Postprueba: prueba con escala de *Likert* con afirmaciones concernientes al uso de conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional. Aplicada al finalizar las siete sesiones de trabajo.

El capítulo 6 se encuentra organizado de la siguiente manera (Fig.6.1), en la primera parte, se analizan los resultados con relación al trabajo de los docentes

con la metodología propuesta. En este sentido se recuperan sus opiniones, destacando puntos fuertes y débiles encontrados.

En la segunda parte, se analizan los resultados de la experiencia con alumnos, y se organizan según su nivel de satisfacción con relación a las actividades realizadas en cada una de las sesiones, el trabajo realizado en equipo, los dispositivos utilizados en las actividades y los entornos de programación involucrados.

De igual manera, se tiene en cuenta el nivel de dificultad encontrado en las actividades y dinámica propuestas, analizando aspectos como las actividades de lectoescritura, hacer uso de bloques de programación, la solución de los desafíos propuestos, la abstracción de información para la creación literaria y el diseño del cuento en *Scratch*.

Estos últimos resultados aportan a uno de los objetivos del estudio de caso relacionado con el alcance de las competencias del pensamiento computacional y las dinámicas para su logro.

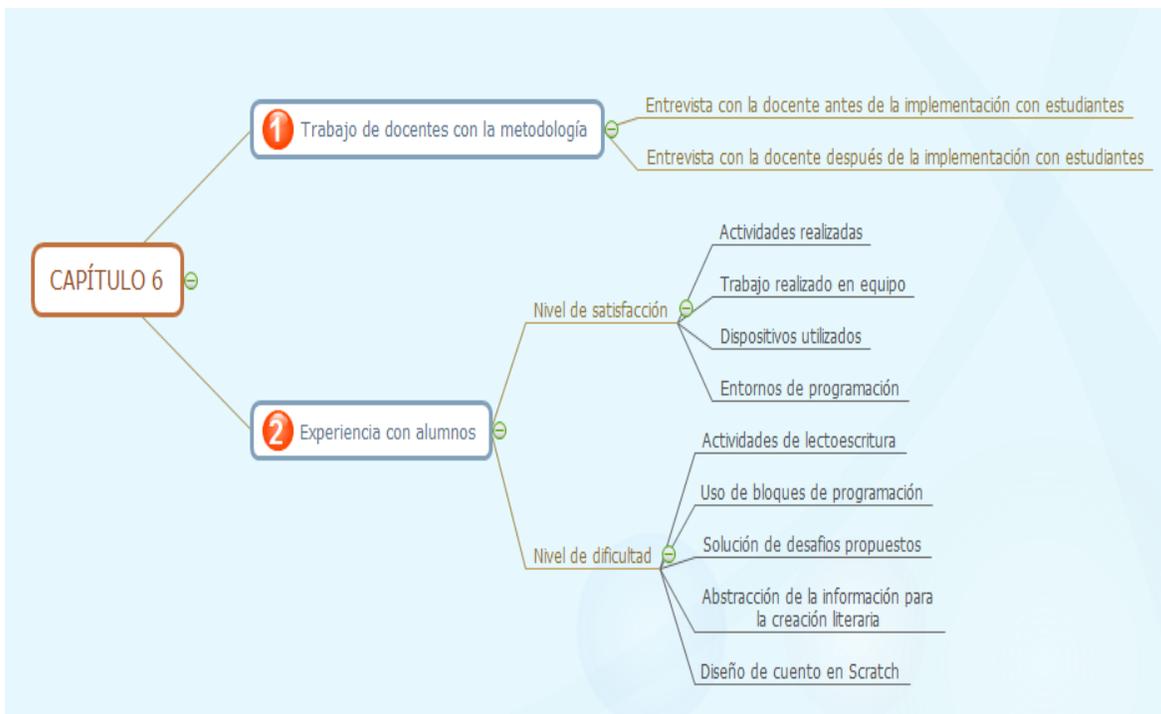


Figura 6.1 - Esquema de recorrido del capítulo⁴²

⁴² Fuente: elaboración propia

6.2. Resultados del trabajo realizado entre docentes con la metodología propuesta

Como ya se mencionó anteriormente, la metodología propuesta se aplicó en el marco de un trabajo conjunto entre la tesista y la docente de Lengua Castellana de la Institución Educativa Departamental Pío XII del municipio de Pacho, Cundinamarca, Colombia. A partir de la aplicación de la metodología, se diseñaron siete sesiones de trabajo áulico con los participantes del Club “Cuenta un Cuento con Tecnología”. Luego, se analizó la información recopilada a partir de las técnicas descritas en la sección anterior.

Para reconocer la opinión que la docente de Lengua Castellana, de ahora en más Diana, tiene con respecto a la metodología propuesta, se realizó una entrevista semiestructurada luego de las sesiones de trabajo. En la entrevista, se analizaron aspectos relacionados con las dos etapas trabajadas para el diseño de las actividades a implementar en el aula.

Las preguntas que se hicieron fueron las siguientes:

1. Luego de la presentación que se ha realizado de la metodología ¿Se pudo comprender la intencionalidad, fundamentación y la forma de trabajo con la metodología?
2. ¿Las etapas de la metodología son fáciles de seguir y de entender?
3. ¿Te parece que el orden de las etapas y sus preguntas son adecuadas?
4. ¿Te parece que las sugerencias dadas por la metodología sirven para diseñar el trabajo en el aula?
5. ¿Por qué parecen importantes las características del pensamiento computacional y las habilidades lectoescritoras que esperan abarcar?
6. ¿Piensas que las decisiones tomadas gracias a la metodología apuntan a que se puedan desarrollar las habilidades, actitudes y conocimientos elegidos?
7. ¿Piensas que hacen falta algunas etapas o algunas preguntas orientadoras en la metodología?

8. ¿Cómo te parece que quedó la propuesta para el trabajo en aula?
9. ¿Te parece importante tratar de integrar el área de Lengua Castellana con el área de tecnología? ¿Por qué?

Ante estas preguntas la docente expresa conocer la propuesta metodológica desde su base por medio de la explicación de la tesista, con respecto a las etapas de la metodología trabajada opina que están bien y son las necesarias para diseñar las actividades. Expresa que en cada etapa se analizan los requisitos precisos para organizar el trabajo a realizar con los alumnos. Con respecto a los interrogantes y sugerencias propuestas por la metodología opina que son suficientes para responder a las necesidades del grupo con el cual se va a trabajar.

Acerca de las competencias elegidas que se esperan desarrollar, la docente opina que son importantes porque le sirven al estudiante para desenvolverse en diversos entornos y no solamente en el ambiente académico. Diana considera que este tipo de competencias se deberían trabajar y desarrollar a diario en el aula con cada asignatura, pero no resulta posible porque el trabajo con 40 estudiantes por curso limita esta posibilidad. Asimismo, espera que gracias a la integración de las dos áreas y el trabajo planeado con este grupo de estudiantes se desarrollen las competencias abordadas.

En cuanto a las decisiones tomadas la docente comenta que en las preguntas orientadoras de cada etapa se cubrieron los requerimientos y los aspectos necesarios para diseñar las actividades de cada una de las sesiones. Esto es porque se abordaron los contenidos y se tuvo en cuenta el contexto y las características de los chicos que conforman la muestra.

Con respecto a las dos etapas trabajadas, definición de requerimientos y diseño y aplicación de la experiencia educativa, Diana opina que son suficientes y cubren los aspectos necesarios para desarrollar la propuesta. Considera que el trabajo no resultó extenso ni confuso al momento de diseñar las actividades.

Con relación a las siete sesiones diseñadas, la docente indica que las preguntas orientadoras en cada etapa guiaron el trabajo a realizar para lograr integrar las áreas, en el momento de planear actividades como: utilizar algoritmos,

solucionar desafíos por medio de bloques, diseñar cuentos teniendo en cuenta las imágenes del desafío, dar a conocer estos cuentos y reconocer que cada actividad tiene un proceso determinado para lograr su ejecución.

De este modo espera que el trabajo resulte llamativo para los estudiantes y así se logre el objetivo que se persigue. En lo que se refiere a la integración de las áreas, la docente comenta que la transversalidad es importante para que los estudiantes vinculen el conocimiento y puedan aplicarlos en otros entornos, tanto académicos como personales.

Para concluir, en esta primera entrevista Diana señala que la metodología resulta organizada y cuenta con las preguntas orientadoras y sugerencias adecuadas para guiar la labor en el diseño de actividades de aula. Al mismo tiempo, menciona que despierta interés para realizar la implementación y verificar los resultados logrados.

En un segundo momento, al finalizar las 7 sesiones de trabajo con el grupo de muestra, se realizó una segunda entrevista donde se dialogó con Diana, respecto al trabajo realizado. Nuevamente se hizo una entrevista semiestructurada en la cual se preguntó lo siguiente:

1. ¿Cómo te ha parecido el trabajo realizado en las sesiones con los chicos?
¿Qué actitudes viste en ellos?
2. ¿Recuerdas la metodología utilizada para el diseño de esta experiencia, etapa 1 y 2? Después de utilizarla y aplicar lo diseñado ¿Cómo te parece?
¿piensas que le hizo falta algo?
3. ¿Te parece que lo planeado en las sesiones para trabajar con los chicos estuvo bien, fue adecuado? ¿Te parece que la propuesta para esas sesiones funcionó?
4. ¿Qué aspectos positivos y negativos observaste en el trabajo realizado?
5. Teniendo en cuenta las habilidades y conocimientos que se propusieron abordar ¿crees que se lograron desarrollar?

A las preguntas, Diana responde de la siguiente manera: *“me parece muy bueno el trabajo que se hizo en las sesiones, observo que los chicos se mostraron*

inquietos al querer conocer el trabajo que se haría en cada sesión, motivados ante el trabajo y persistentes en el momento de solucionar los desafíos y actividades propuestas”.

Con referencia a la metodología, indica que la estructura guio el trabajo, fue acorde al proceso y los chicos lograron la integración de las dos áreas de conocimiento. Además, opina que se abordó la temática a tratar y se fortalecieron las habilidades seleccionadas, con un buen resultado.

Diana comenta que: *“la organización de las sesiones estuvo bien pues apuntó a desarrollar las habilidades y con respecto al trabajo de los chicos también resultó favorable porque en cada sesión trabajaron con diferentes compañeros, de esta manera se conocieron y pudieron llegar a acuerdos para realizar el trabajo asignado”.* También, destaca que las actividades propuestas en cada sesión resultaron motivantes para los participantes, aunque al comienzo fue difícil lograr que los chicos trabajaran en equipo porque no se conocían, y en algunos casos, no tenían afinidad, pero luego se mostraron interesados en el trabajo porque además podían conocerse y lograr fortalecer su parte social.

Con respecto a los aspectos negativos, Diana observó que: *“al inicio de las sesiones fue difícil lograr que todos los chicos convocados asistieran a los encuentros, porque la mayoría de ellos realizan otras actividades del colegio, y por la edad que tienen y la etapa que están viviendo, no se organizan de la mejor manera para poder cumplir con las jornadas”.* En la parte positiva la docente percibió el mejoramiento de las habilidades de los chicos, la motivación con que asistieron, el interés, la agilidad y persistencia con que desarrollaron las actividades. Pero sobre todo lo que le pareció sobresaliente es que el trabajo realizado con este pequeño grupo fue reconocido por la institución y otros estudiantes manifestaron estar interesados en trabajar proyectos de este tipo. Así, la idea de la docente es trabajar nuevamente con más estudiantes en una próxima oportunidad.

Con relación a las habilidades abordadas Diana afirma que efectivamente se lograron fortalecer con el trabajo realizado. Principalmente, las habilidades que

tienen que ver con la resolución de problemas, la descomposición, el reconocimiento de datos principales, el diseño de algoritmos y el trabajo en grupo.

Para concluir, la entrevista realizada después de las 7 sesiones con los chicos, muestra que la docente se siente satisfecha con el trabajo realizado, comenta sobre las fortalezas que percibió tanto de la metodología como del trabajo en el aula y manifiesta que fueron pocas las debilidades encontradas a los largo de las sesiones, ninguna de ellas relacionada con la metodología, sino con el horario en contra jornada escolar planeado para la implementación, en un principio se citaron 20 estudiantes de los cuales 16 estudiantes participaron permanentemente.

Resultados posteriores a las sesiones en el aula

A continuación, se analizan los resultados de la experiencia con alumnos, organizados según nivel de satisfacción y de dificultad encontrados en las actividades y dinámicas propuestas. Los criterios a tener en cuenta son las siguientes:

- Nivel de satisfacción con relación a:
 - Las actividades realizadas en cada una de las sesiones.
 - Trabajo realizado en equipo.
 - Los dispositivos utilizados en las actividades.
 - Los entornos de programación con los que se trabajó.

- Nivel de dificultad encontrado en las actividades y dinámica propuestas con relación a:
 - Las actividades de lectoescritura.
 - Hacer uso de bloques de programación.
 - La solución de los desafíos propuestos.
 - La abstracción de información para la creación literaria.
 - El diseño del cuento en *Scratch*.

6.2.1. Análisis sobre el nivel de satisfacción

6.2.1.1. Nivel de satisfacción respecto a las actividades realizadas en cada una de las sesiones

Al finalizar las sesiones uno y cuatro, se realizaron entrevistas semiestructuradas, del total de 16 estudiantes en cada una de estas sesiones se entrevistaron a 3 de ellos. Para referir a la opinión de estos alumnos se los mencionará como Participante 1 (P1), Participante 2 (P2), Participantes 3 (P3), Participante 4 (P4), Participante 5 (P5), Participantes 6 (P6).

En las entrevistas realizadas se indagó sobre el trabajo realizado con preguntas como: ¿Cómo te pareció el trabajo que hicimos hoy con tus compañeros? ¿Por qué?, ¿Cuál fue la actividad que más te gustó? ¿Por qué? ¿Cuál fue la actividad que menos te gustó? ¿Por qué?

Los estudiantes en su mayoría se mostraron satisfechos con las actividades realizadas. Únicamente el participante 2 y el participante 5 mostraron cierta desconformidad con relación a la solución de un desafío, en el que tuvieron dificultad para lograr su resolución. P2, indicó que debió utilizar muchos bloques para la solución y el segundo, P5, manifestó que no pudo cambiar el idioma inglés en que estaba el entorno *Scratch*.

A continuación, en las tablas 6.1 – 6.2 – 6.3, se presentan extractos de la transcripción realizada a partir del registro audiovisual en la entrevista 1 de la sesión uno y la entrevista 2 de la sesión cuatro.

Tabla 6.1 Entrevista 1 y 2. ¿Cómo te pareció el trabajo que hicimos hoy con tus compañeros? ¿Por qué?

Entrevista 1 (después de la sesión 1)	
P1	Bien, pensé que iba a ser difícil, pero fue fácil
P2	Bien, me sentí bien
P3	Bien, bonito
Entrevista 2 (después de la sesión 4)	
P4	Excelente, me pareció interesante Scratch y sus movimientos
P5	Bien, pero unas cositas no nos cuadraban, aunque ya las pudimos corregir
P6	Chévere, porque aprendemos más de la tecnología y aprendemos a trabajar con los demás

Tabla 6.2 Entrevista 1 y 2. ¿Cuál fue la actividad que más te gustó? ¿Por qué?

Entrevista 1 (después de la sesión 1)	
P1	Me gustaron las actividades en el computador
P2	El cuento porque debíamos hacer un cuento con solamente una imagen
P3	Me gustó cuando desarrollamos el juego del marciano, porque tocó en compañía y fue fácil hacerlo
Entrevista 2 (después de la sesión 4)	
P4	El Scratch fue lo que más me gustó, porque es entretenido
P5	Cuando pusimos las imágenes del cuento, es entretenida y aprendo
P6	Cuando hicimos un gato y un ratón, cuando el gato perseguía al ratón era chévere hacer que se movieran.

Tabla 6.3 Entrevista 1 y 2. ¿Cuál fue la actividad que menos te gustó? ¿Por qué?

Entrevista 1 (después de la sesión 1)	
P1	Todas me gustaron estuvieron fáciles y me gustaron
P2	Que en el desafío se debía repetir muchas veces para poder dar la respuesta
P3	Todas me gustaron
Entrevista 2 (después de la sesión 4)	
P4	Escribir el cuento, porque solamente tenía que escribir algo corto
P5	Scratch estaba en inglés y no entendíamos y luego no funcionaba lo que hacíamos
P6	Escribir el inicio porque no sabíamos cómo empezar

Por otro lado, al finalizar la sesión 2, se realizó un *focus group*, registrado mediante video donde se observa la actitud de los participantes. Todos se mostraron satisfechos, con respecto al trabajo en equipo, a la creación literaria, al trabajo con Pilas Bloques, a la construcción de nuevos aprendizajes, a mejorar habilidades como expresión oral y el trabajo con los demás. Algunos ejemplos de los comentarios que se realizaron con respecto a esta cuestión fueron:

- *“Me gustó porque aprendimos a hacer el nudo de un cuento y aprendimos a trabajar en Pilas Bloques”.*
- *“Aprendimos a trabajar con nuestros compañeros”*
- *“Mejoramos nuestra habilidad de hablar en público”*
- *“En lugar de usar tantas fichas usamos bloques de repetición”*

6.2.1.2. Nivel de satisfacción respecto del trabajo realizado en equipo.

En las entrevistas realizadas después de la primera y tercera sesión, se preguntó a 3 de los participantes en cada una de ellas si les gustó trabajar con otro compañero y ¿por qué? A lo cual los estudiantes respondieron, en su mayoría afirmativamente, solamente el P5 responde que no se sintió cómodo trabajando con otro compañero porque tenía ideas diferentes, pero pudieron llegar a acuerdos. La tabla 6.4 presenta extractos de la transcripción realizada a partir del registro audiovisual en la entrevista 1 de la sesión uno y la entrevista 2 de la sesión cuatro.

Tabla 6.4 Entrevista 1 y 2. ¿Te gustó trabajar con otro compañero? ¿Por qué?

Entrevista 1 (después de la sesión 1)	
P1	Si me gusto trabajar con mi compañero
P2	Sí, porque me ayudó, me gusta trabajar grupal porque podemos colaborarnos
P3	Sí, pero mi compañero es un poquito fastidioso porque me dice que haga esto y después lo hace él
Entrevista 2 (después de la sesión 4)	
P4	Sí, porque ella me daba más ideas y fue fácil trabajar con ella
P5	No, tenía ideas diferentes, pero llegamos a acuerdos
P6	Sí, porque casi siempre trabajo con mujeres y no con hombres, mi compañero me colaboró, los dos teníamos las mismas ideas

En las pruebas con escala de *Likert* 1 y 2 aplicadas en la sesión 3 y 6, respectivamente, se consultó sobre el trabajo en equipo. Los participantes en su mayoría se mostraron satisfechos como se observa en la Figura 6.2 En la primera sesión se indicaron niveles Muy bien por parte de 13 estudiantes y Bien, referido por 3 estudiantes. En la segunda sesión se 9 estudiantes indicaron el nivel Muy bien, 6 estudiantes el nivel Bien y un solo estudiante el Regular.

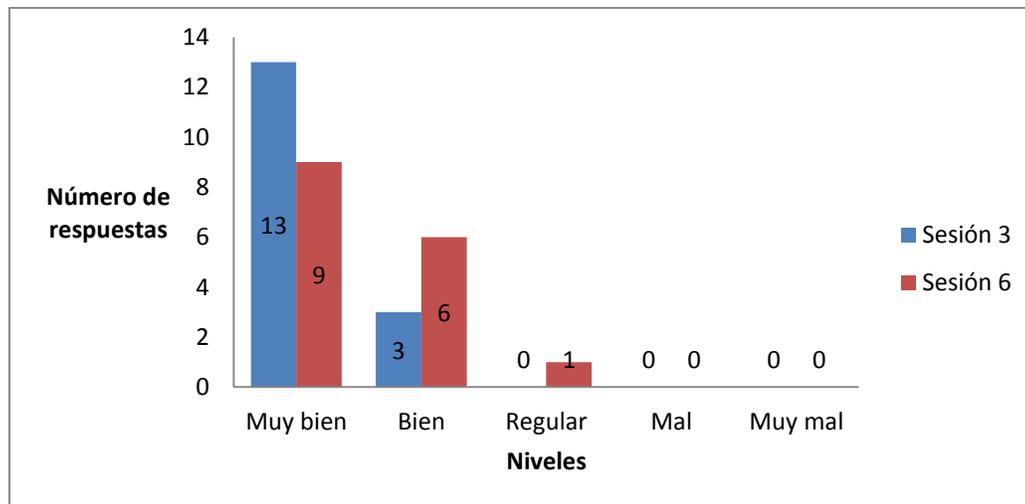


Figura 6.2 – Prueba con escala de Likert 1 y 2. Indica cómo te sentiste con respecto a trabajar en equipos de 2 personas

Al preguntar a los participantes en las pruebas con escala de *Likert 1 y 2* aplicadas en la sesión 3 y sesión 6, sobre la facilidad o dificultad (Fácil 1- Difícil 5) para llegar a acuerdos con el compañero de trabajo, como se puede observar en la Figura 6.3, la mayoría de los estudiantes consideraron que fue fácil llegar a acuerdos. En la sesión 3 se ubicaron en Nivel 1, 13 estudiantes, en Nivel 2, 3 estudiantes y un solo estudiantes en Nivel 5.

En la sesión 6, se ubicaron en Nivel 1, 6 estudiantes, en Nivel 2, 7 estudiantes, Nivel 3, 2 estudiantes y Nivel 5 un solo estudiante. Se observa a partir de los resultados entre ambas sesiones que, aunque los estudiantes al final llegan a acuerdos para poder cumplir las metas, estos acuerdos se dificultan cuando más compleja es la meta por cumplir. En el primer caso (sesión 3) el trabajo asignado era solucionar un desafío en *Minecraft* hora del código, actividad que les resultó sencilla. En tanto que en el segundo caso (sesión 6) la actividad se basaba en tratar de terminar el cuento programando en Scratch, trabajo que les resultó un poco más complejo.

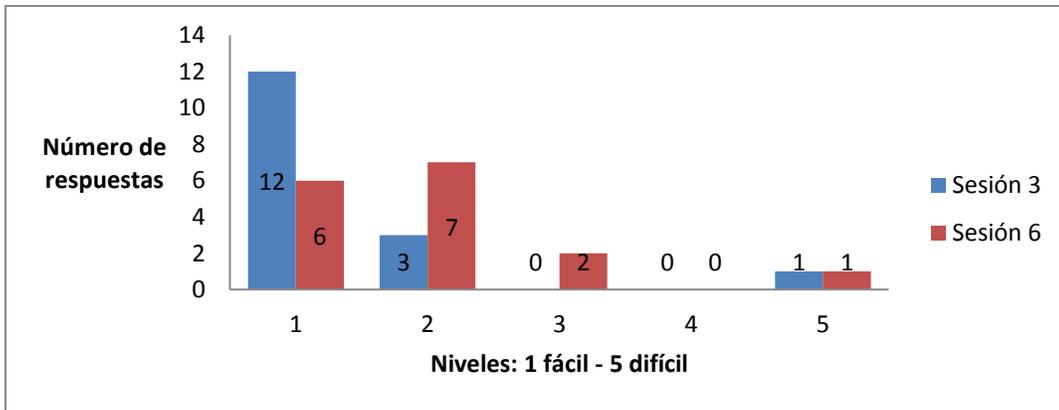


Figura 6.3 - Prueba con escala de Likert 1 y 2. Llegar a acuerdos con el compañero de equipo.

Al preguntar a los participantes en las pruebas con escala de *Likert* 1 y 2 aplicadas en la sesión 3 y sesión 6 sobre la Importancia (Importante 1- Sin importancia 5) de contar con un compañero de trabajo o trabajar en equipo con un compañero, como se observa en la Figura 6.4, a la mayoría de los estudiantes les parece importante. En la sesión 3, se ubicaron en Nivel 1, 12 estudiantes y Nivel 2, 4 estudiantes. En la sesión 6, se ubicaron en Nivel 1, 11 estudiantes, en Nivel 2, 4 estudiantes y en Nivel 3, un solo estudiante.

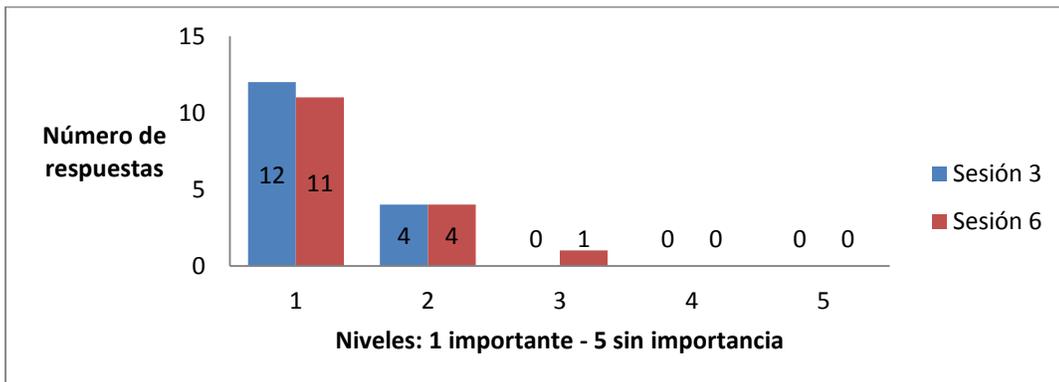


Figura 6.4 – Prueba con escala de Likert 1 y 2. Contar con un compañero de trabajo o trabajar en equipo con un compañero.

A partir de las respuestas que los estudiantes manifiestan en el *focus group* se revela una actitud positiva con respecto al trabajo en equipo, según las intervenciones. Los estudiantes dicen conocer mejor a sus compañeros y compartir, colaborar entre sí y aunque piensen diferente pueden llegar a

acuerdos. Algunos ejemplos de los comentarios que realizaron con respecto a esta cuestión fueron:

- *“Me fue bien trabajando con mi compañera”*
- *“Me fue bien porque pudimos conocernos y ayudarnos”*
- *“bien, porque pude compartir con mis compañeros, pero mi compañera a veces es terca”*
- *“Podemos trabajar mejor porque si yo no entendía algo mi compañera me corregía y si ella no entendía algo yo le corregía”*

6.2.1.3. Nivel de satisfacción respecto a los dispositivos utilizados en las actividades.

Según la Prueba con escala de *Likert 1* aplicada en la sesión 3, como se observa en la figura 6.5, los estudiantes en su mayoría consideraron sentirse muy bien con respecto a trabajar en la sala de informática, seleccionando los niveles Muy bien y Bien: 10 eligieron Muy bien y 5 Bien, únicamente 1 estudiante dice sentirse Regular trabajando en la sala de informática.

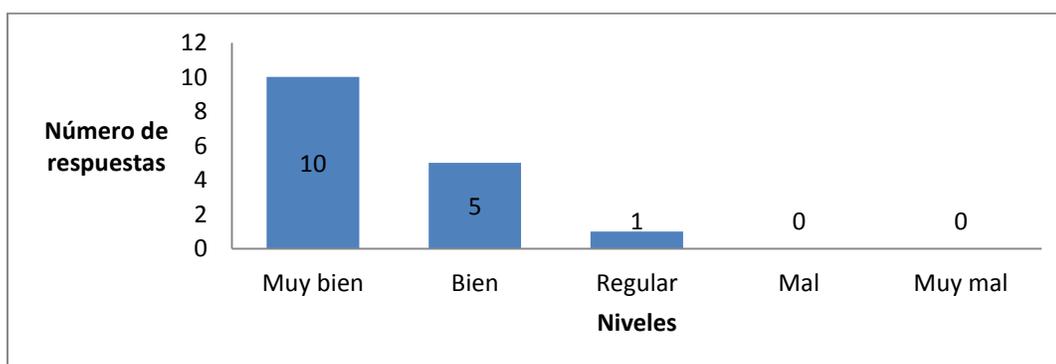


Figura 6.5 – Prueba con escala de *Likert 1*. Trabajar en la sala de informática.

En cuanto a la importancia que le dan los estudiantes al poder trabajar con las computadoras de la institución (Importante: 1- Sin importancia: 5) según la prueba con escala de *Likert 1* sesión 3, como se muestra en la Figura 6.6, los participantes en su mayoría indican que les parece importante, seleccionando los Niveles 1 y 2: 10 eligieron el Nivel 1, y 5 eligieron el Nivel 2, únicamente 1 estudiante elige Nivel 3 indicando que le resulta medianamente importante.

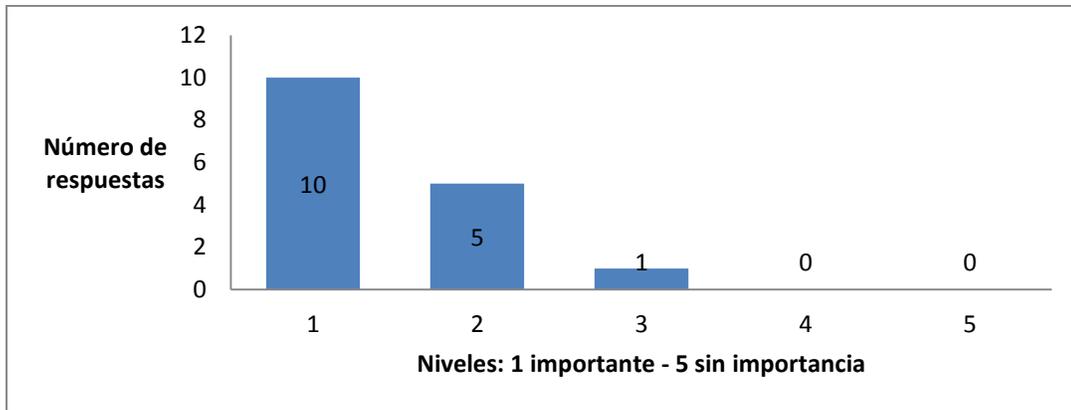


Figura 6.6 – Prueba con escala de Likert 1. Poder trabajar con las computadoras.

Como se observa en la figura 6.7, prueba con escala de *Likert* 1 sesión 3 los alumnos consideraron importante poder visualizar las explicaciones con ayuda del Video *Beam*. Se seleccionaron en un 100% de los casos, los niveles 1 y 2: 12 eligieron el Nivel 1 y 4 eligieron el Nivel 2.

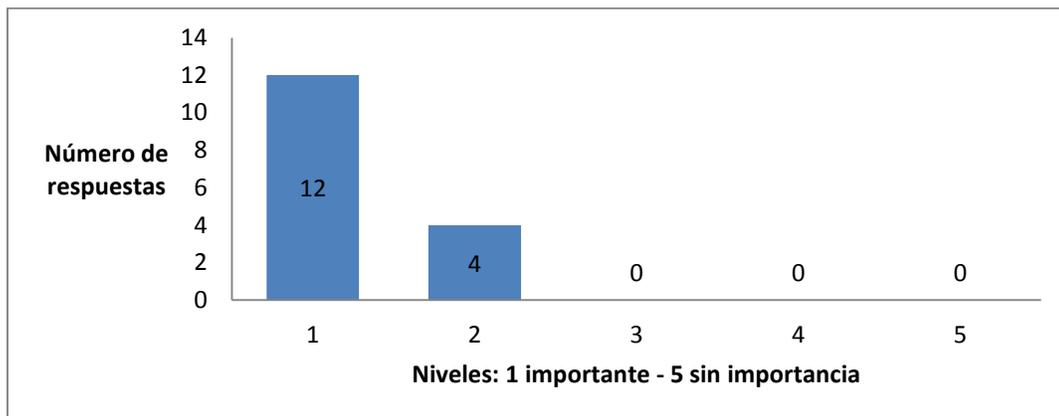


Figura 6.7 - Prueba con escala de Likert 1. Ver las explicaciones con ayuda del Video *Beam*.

Según se observó en las sesiones y el registro audiovisual, los estudiantes se muestran interesados en utilizar los dispositivos con los que se cuenta en la sala de informática de la institución, les agrada el trabajo con las computadoras para dar solución a las actividades planteadas y manifiestan la importancia que tienen las explicaciones realizadas con el uso del Video *Beam*.

6.2.1.4. Nivel de satisfacción respecto de los entornos de programación utilizados.

En la entrevista a los estudiantes y en respuesta a las preguntas: ¿Cómo te pareció el entorno de programación? ¿Pudiste solucionar los desafíos que se propusieron? Y ¿Cómo te pareció el entorno de programación *Scratch*? ¿Fue fácil o difícil utilizarlo? Respondieron, en su mayoría, de forma positiva. En las tablas 6.5 y 6.6 se pueden observar algunos extractos de las respuestas transcritas del registro audiovisual.

Tabla 6.5 Entrevista 1. ¿Cómo te pareció el entorno de programación? ¿Pudiste solucionar los desafíos que se propusieron?

Entrevista 1 (después de la sesión 1)	
P1	Me gustó, de uno a diez, diez, tienen su lógica para hacerlo porque digamos uno se puede confundir con una cosita y hay cosas que mejorar
P2	Bien, pude solucionarlos
P3	Me pareció bien, me parecieron fáciles los desafíos

*Tabla 6.6 Entrevista 2. ¿Cómo te pareció el entorno de programación *Scratch*? ¿Fue fácil o difícil utilizarlo?*

Entrevista 2 (después de la sesión 4)	
P4	Me gustó <i>Scratch</i> , fue un poquito difícil utilizarlo porque nos quedaba mal y pedí ayuda para solucionarlo
P5	Me pareció chévere y entendí fácil, fácil porque ya sé cómo poner la fichitas y difícil para encontrar las fichitas correctas.
P6	Chévere, fue fácil utilizarlo

Y a la pregunta: ¿Qué fue lo que más te gustó de este entorno de programación? Y con relación a *Scratch* ¿Qué fue lo que no te gustó? Los extractos de las respuestas se presentan a continuación en las tablas 6.7 y 6.8

Tabla 6.7 Entrevista 1. ¿Qué fue lo que más te gustó de este entorno de programación? (Pilas Bloques)

Entrevista 1 (después de la sesión 1)	
P1	Los desafíos
P2	El marciano porque debía escribir el algoritmo
P3	Que podemos trabajar juntos en los desafíos

Tabla 6.8 Entrevista 2. ¿Qué fue lo que más te gustó y qué no te gustó de este entorno de programación? (Scratch)

Entrevista 2 (después de la sesión 4)	
P4	Los cambios que se pueden hacer de fondos de imágenes, todo me gustó
P5	Me gustó todo, me gustó ver como todo iba quedando
P6	Me gustó como corrían las imágenes y no me gustó porque se detenía

En la prueba con escala de *Likert 1* sesión 3 y escala de *Likert 2* sesión 6 con respecto a Utilizar Pilas Bloques, *Minecraft* y Utilizar *Scratch* para animar el cuento, como se observa en la Figura 6.8, en su mayoría los estudiantes se sintieron muy bien. Se seleccionaron en un 100% de los casos en las 2 sesiones, los Niveles Muy bien y Bien, 12 eligieron Muy bien y 4 eligieron Bien.

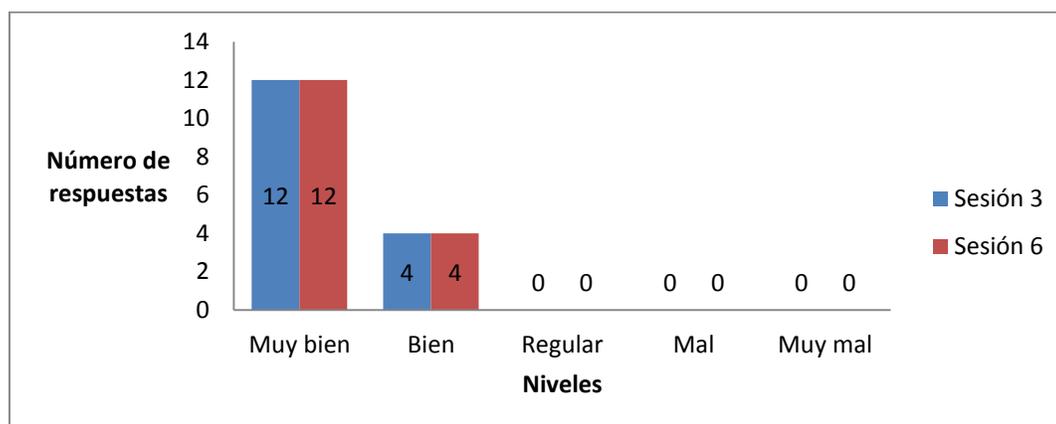


Figura 6.8 – Prueba con escala de *Likert 1* y 2. Utilizar Pilas Bloques, *Minecraft* y *Scratch*.

Se preguntó en las pruebas con escala de *Likert 1* y 2 de las sesiones 3 y sesión 6, en cuanto a la importancia (Importante 1- Sin importancia 5) que le dan los estudiantes al trabajo con los entornos de programación, en su mayoría respondieron que es importante como se observa en la figura 6.9. En la sesión 3, eligieron el Nivel 1, 10 estudiantes, Nivel 2, 5 estudiantes y un estudiante en Nivel 3. En la sesión 6, se ubicaron en Nivel 1, 13 estudiantes y en Nivel 2, 3 estudiantes. Como se puede observar a medida que las sesiones trascurrían el interés de los chicos por aprender sobre los entornos de programación aumentaba.

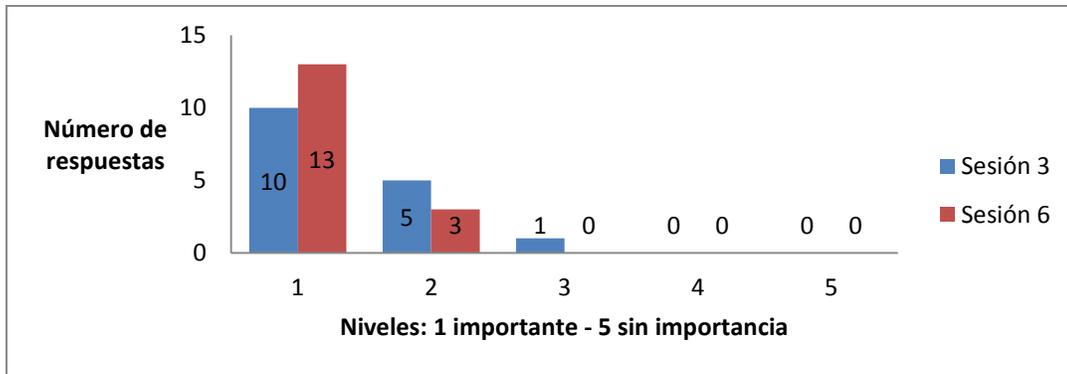


Figura 6.9 – Prueba con escala de Likert 1 y 2. Conocer diferentes entornos de programación (Pilas bloques, Minecraft y Scratch)

Con respecto a los entornos de programación utilizados, Pilas Bloques, *Minecraft* hora del código y *Scratch*, los participantes mencionaron en el *focus group* que los entornos les resultaron entretenidos y mostraron interés en continuar trabajando y aprender a utilizarlos mejor. Algunos ejemplos de los comentarios que realizaron con respecto a esta cuestión fueron:

- *“Me siento bien porque aprendimos a manejar nuevas fichas”*
- *“Si los utilizamos podemos aprender a utilizar comandos sin equivocarnos”*
- *“Son demasiado divertidos”*
- *“Podemos aprender sobre las aplicaciones, como funcionan”*
- *“Me parece bien porque aprendemos sobre programación”*

6.2.2. Análisis sobre el nivel de dificultad encontrado en las actividades y dinámicas propuestas

6.2.2.1. Nivel de dificultad en relación a las actividades de lectoescritura.

En la entrevista 1 se preguntó a los participantes con respecto al escrito realizado. Las preguntas fueron las siguientes: ¿Cómo te quedó el inicio del cuento que escribiste? ¿Fue fácil hacerlo? Los tres estudiantes que fueron entrevistados respondieron positivamente al trabajo realizado como se muestra en la tabla 6.9.

Tabla 6.9 Entrevista 1. ¿Cómo te quedó el inicio del cuento que escribiste? ¿Fue fácil hacerlo?

Entrevista 1 (después de la sesión 1)	
P1	Bien, aunque fue un poco difícil escribirlo porque debía tener coherencia y cohesión
P2	Bien, fue fácil
P3	Bien, el marciano se perdió, la comida favorita eran las manzanas , encontró un árbol lleno de manzanas y se comió casi todas las manzanas

En la entrevista 2 se preguntó a los participantes si pudieron animar y terminar la primera parte del cuento, a lo cual responden como se muestra en la tabla 6.10, según los extractos tomados del registro audiovisual

Tabla 6.10 Entrevista 2. ¿Pudiste animar el principio de tu cuento en Scratch? ¿Lo terminaste?

Entrevista 2 (después de la sesión 4)	
P4	Si lo pude animar y lo terminé
P5	Si, casi todo lo hice
P6	Sí, me faltó porque estábamos eligiendo el fondo y los personajes

Como se observa en la figura 6.10, los participantes se sintieron satisfechos al realizar una creación literaria, según las pruebas escala de *Likert 1 y 2* de las sesiones 3 y 6. En la sesión 3, seleccionaron Nivel 1, 11 estudiantes, Nivel 2, 4 estudiantes y Nivel 3 un estudiante. En la sesión 6, los estudiantes eligieron Nivel 1, 9 estudiantes y Nivel 2, 7 estudiantes.

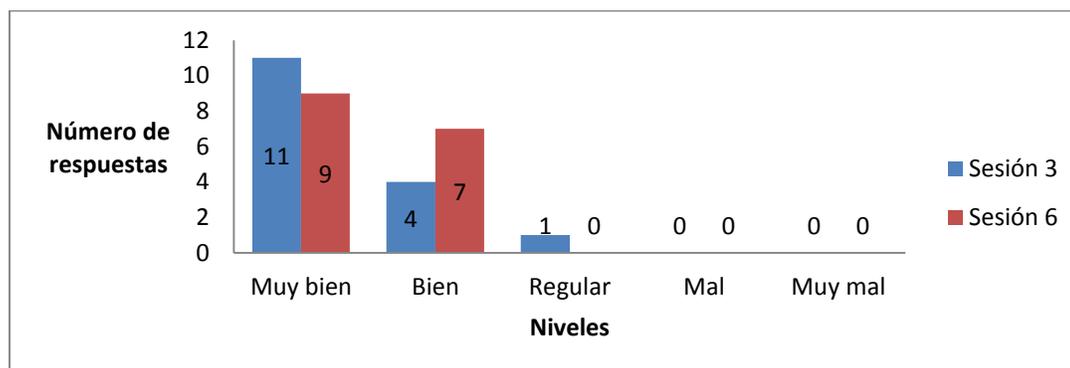


Figura 6.10 - Prueba con escala de Likert 1 y 2. Realizar una creación literaria

6.2.2.2. Nivel de dificultad en relación a hacer uso de bloques de programación.

En las pruebas con escala de *Likert* 1 y 2 sesiones 3 y 6, se pidió a los participantes calificar la actividad (Fácil: 1- Difícil: 5) relacionada con usar bloques de comando tanto en Pilas Bloques como en *Scratch*. Como se observa a la figura 6.11, en la sesión 3 se ubicaron en Nivel 1, 14 estudiantes, en Nivel 2 y Nivel 3, 1 estudiante. Para la sesión 6, se ubicaron en Nivel 1, 8 estudiantes, en Nivel 2, 6 estudiantes y en Nivel 3 y 4 un estudiante. Esto indica que, aunque para los estudiantes resultó fácil el trabajo con bloques de comando, con el paso de las sesiones la dificultad en su uso fue aumentando, acorde se trabajó con bloques con semántica de mayor complejidad.

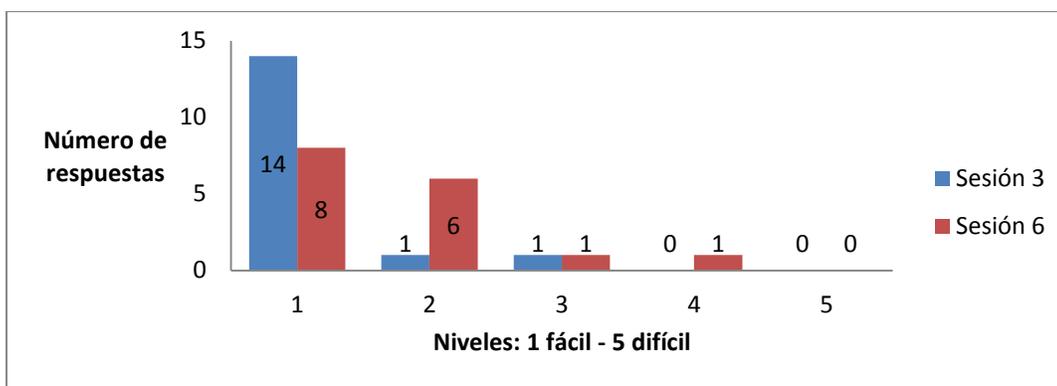


Figura 6.11 - Prueba con escala de *Likert* 1 y 2. Nivel de dificultad al utilizar bloques de comando

Como se observa a la figura 6.12 con respecto a la actividad en donde se usan bloques de repetición, en las pruebas con escala de *Likert* 1 y 2 sesión 3 y 6 debían calificar (Fácil: 1- Difícil: 5) las actividades realizadas. En la sesión 3 los estudiantes se ubicaron en Nivel 1, 9 estudiantes y Nivel 2, 7 estudiantes. En la sesión 6, se ubicaron en Nivel 1, 8 estudiantes, Nivel 2, 5 estudiantes, Nivel 3, 2 estudiantes y Nivel 4 un estudiante. La tendencia muestra que a la mayoría de los estudiantes les parece fácil el uso de bloques de repetición, pero a la vez se indica que se aumentó en dificultad a medida que se avanzó en las sesiones.

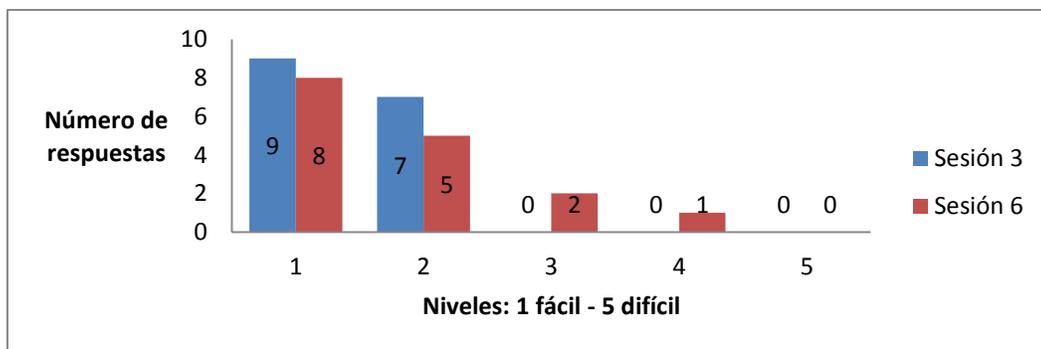


Figura 6.12 - Prueba con escala de Likert 1 y 2. Utilizar bloques de repetición

En relación a utilizar bloques de alternativa condicional como se muestra en la Figura 6.13, según las pruebas con escala de *Likert* sesiones 3 y 6. Los estudiantes calificaron (Fácil: 1- Difícil: 5) las actividades realizadas. En la sesión 3, 10 estudiantes se ubicaron en Nivel 1 y 6 estudiantes en Nivel 2. En la sesión 6, 8 estudiantes se ubicaron en Nivel 1, en Nivel 2, 5 estudiantes, en Nivel 3, 2 estudiantes y en Nivel 4 un estudiante, mostrando nuevamente que utilizaron con facilidad bloques de alternativa condicional, pero a medida que avanzaron en las sesiones aumentó la complejidad en su uso.

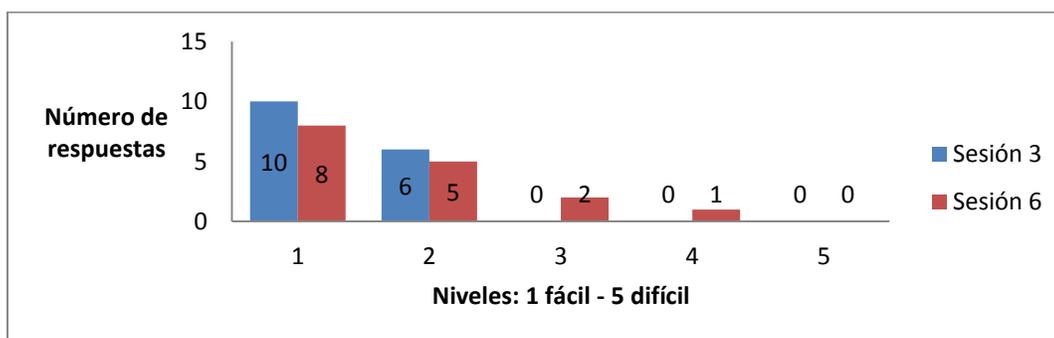


Figura 6.13 - Prueba con escala de Likert 1 y 2. Utilizar bloques de alternativa condicional

Según la observación que se hizo en cada una de las sesiones, los estudiantes muestran facilidad para utilizar los distintos bloques de programación en el entorno Pilas Bloques, desde los bloques de comandos hasta los bloques de repetición y de alternativa condicional, que tienen mayor complejidad. De esta manera dieron respuesta a los desafíos propuestos. Con relación al uso de los bloques utilizados en *Scratch* los estudiantes mostraron mayor dificultad en el momento de querer

programar los movimientos para sus creaciones literarias, algunos grupos necesitaron indicaciones de la docente para poder lograr las acciones que querían alcanzar.

Algunos ejemplos de las preguntas que realizaron con respecto a esta cuestión fueron:

- “¿Cómo hago para esconder a los dos personajes y que después aparezcan?”
- “¿Cómo hacemos para que camine bien?”
- “¿Cómo hacemos que este personaje no aparezca aquí sino en la otra escena?”
- “¿Cómo hacemos para que todos los personajes aparezcan en ese fondo?”
- “No sé, este personaje tiene que aparecer en otro escenario, ¿Cómo hago?”

6.2.2.3. Nivel de dificultad en relación a la solución de los desafíos propuestos.

En la entrevista 1, cuando se preguntó a los participantes: ¿El tiempo que te dieron fue suficiente para terminar los desafíos que debías trabajar? Como se observa en la tabla 6.11 los estudiantes responden que se sintieron conformes con el tiempo dado para solucionar los desafíos.

Tabla 6.11 Entrevista 1 primera sesión. ¿El tiempo que te dieron fue suficiente para terminar los desafíos que debías trabajar?

Entrevista 1 (después de la sesión 1)	
P1	El tiempo fue suficiente
P2	Para mí fue suficiente
P3	El tiempo fue suficiente

Al aplicar las pruebas con escala de *Likert* 1 y 2, correspondientes a la sesión 3 y 6, se pidió a los participantes indicar (Fácil: 1- Difícil: 5) a la afirmación “Entender la explicación de las profes” (ver la figura 6.14). En la sesión 3, 14 estudiantes se

ubicaron en Nivel 1, y 2 en Nivel 2. En la sesión 6, 13 estudiantes se ubicaron en Nivel 1 y 3 estudiantes en Nivel 2. Como se puede percibir, a la mayoría de estudiantes les resultó fácil entender las explicaciones dadas por las docentes en distintas sesiones.

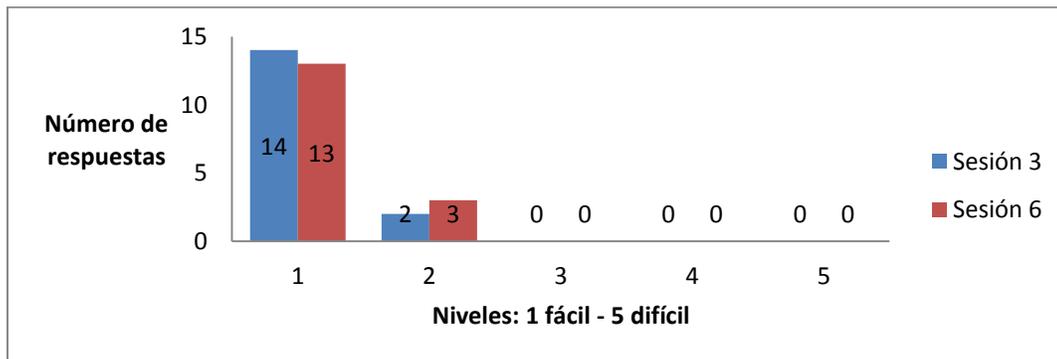


Figura 6.14 - Prueba con escala de Likert 1 y 2. Entender la explicación de las profes

Como se dijo anteriormente, con respecto a Pilas Bloques los participantes dieron solución a los desafíos propuestos, sin mostrar ningún tipo de dificultad, dirigiéndose a las docentes para hacer preguntas sobre cómo eliminar algunos bloques o al terminar cada desafío para mostrar la solución dada. Además, se los desafíos fueron resueltos correctamente y se utilizó la información de las escenas como base para sus creaciones literarias.

En cuanto a la creación en *Scratch* cada parte del cuento se convirtió en un desafío al cual los estudiantes dieron distintas respuestas. Unos equipos de estudiantes lograron avanzar sin dificultad, mientras otros mostraron dudas con respecto al uso de los bloques para lograr mover los personajes como lo tenían planeado. También hubo dudas sobre el uso de distintos escenarios, que se solventaron preguntando a otros compañeros o a la docente. Algunos ejemplos de los comentarios que realizaron con respecto a esta cuestión fueron:

- “Profe, ¿para que este personaje llegue acá y diga algo cómo puedo trabajar?”
- “Profe, ¿Cómo hacemos para que el personaje aparezca?”

6.2.2.4. Nivel de dificultad en relación a la abstracción de información para la creación literaria.

En la entrevista 1, cuando se preguntó a los participantes ¿Cuál es el título de tu cuento? ¿Por qué elegiste ese nombre?, la respuesta fue:

Tabla 6.12 Entrevista. ¿Cuál es el título de tu cuento? ¿Por qué elegiste ese nombre?

Entrevista 1 (después de la sesión 1)	
P1	“El marciano en el desierto” Lo escogí porque era fácil para empezar a hacer el inicio del cuento
P2	“Desafío de manzanas”, porque en el desafío el marciano debía comer las manzanas
P3	“El marciano del desierto” me basé en la imagen

En la prueba con escala de *Likert* 1 de la sesión 3, se pidió a los participantes indicar (Fácil: 1- Difícil: 5) el grado de facilidad respecto a “Tomar información de las escenas de Pilas Bloques y *Minecraft* para escribir el cuento”. Como se observa en la figura 6.15 la mayoría de los estudiantes se ubicó en Nivel 1 y 2: 12 estudiantes eligieron Nivel 1 y 4 estudiantes en Nivel 2.

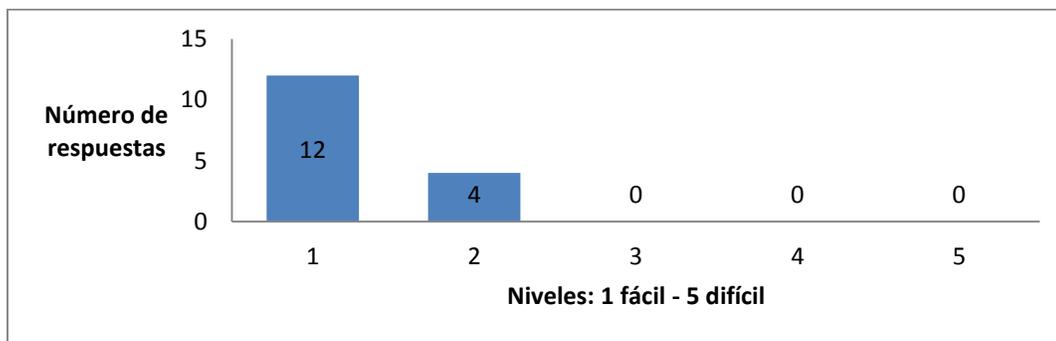


Figura 6.15 - Prueba con escala de *Likert* 1. Tomar información de las escenas de Pilas Bloques y *Minecraft* para escribir el cuento

En la misma prueba con escala de *Likert* se pidió a los participantes indicar (Muy importante 1- Sin importancia 5) respecto a: “Aprender a escribir una narración con base en las escenas de los desafíos”. Como se observa en la figura 6.16, la mayoría de los estudiantes se ubicó en el Nivel 1 y 2: 15 estudiantes eligieron Nivel 1 y un estudiante se ubicó en Nivel 2.

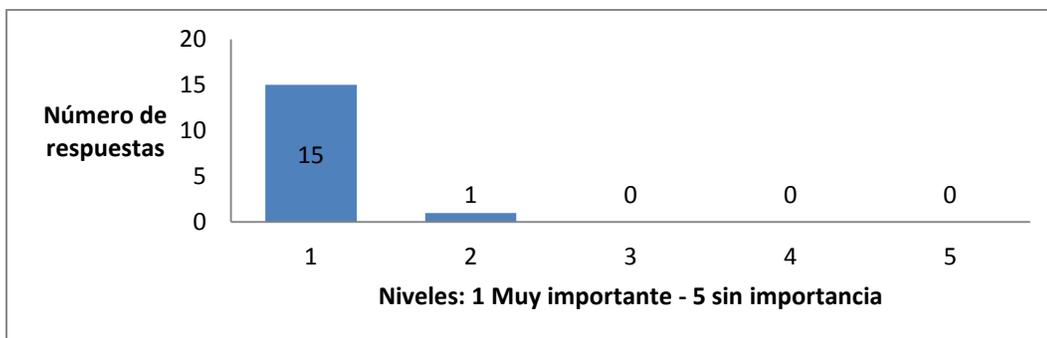


Figura 6.16 - Prueba con escala de Likert 1. Aprender a escribir una narración con base en las escenas de los desafíos

6.2.2.5. Nivel de dificultad en relación al diseño del cuento en Scratch.

Al preguntar a los participantes en la prueba con escala de *Likert 2* aplicada en la sesión 6, como se observa en la figura 6.17, los estudiantes en su mayoría consideraron sentirse muy bien con respecto a “Utilizar Scratch para animar el cuento”. Así seleccionaron los niveles Muy bien y Bien: 12 estudiantes Muy bien y 4 estudiantes Bien.

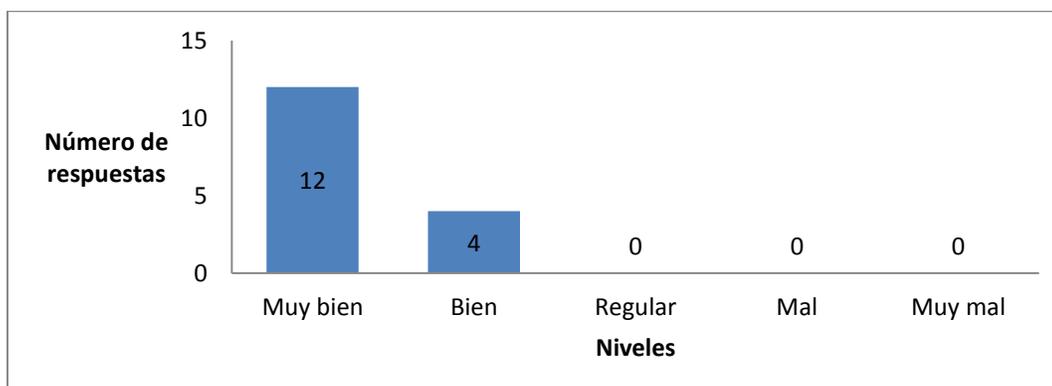


Figura 6.17 - Prueba con escala de Likert 2. Utilizar Scratch para animar el cuento

En la prueba con escala de *Likert 2* sesión 6, se pidió a los participantes indicar (Muy importante 1- Sin importancia 5) respecto a “Realizar animaciones en Scratch”. Como se observa en la figura 6.18, la mayoría de los estudiantes se ubicó en el Nivel 1 y 2: 11 estudiantes eligieron Nivel 1 y 5 estudiantes en Nivel 2.

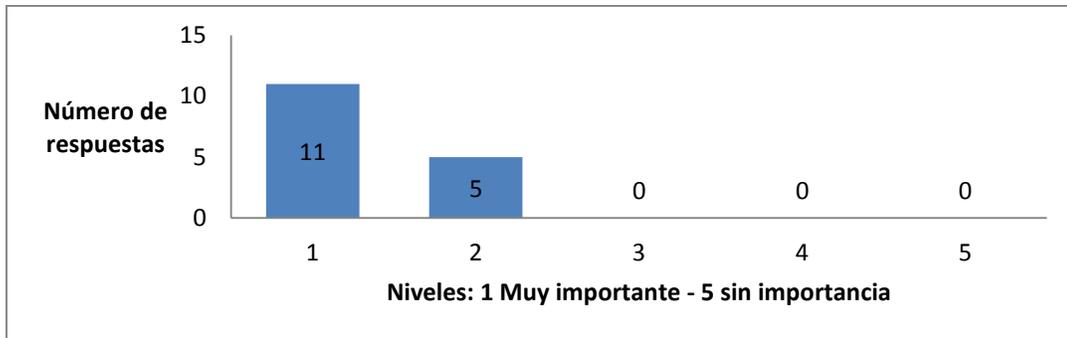


Figura 6.18 - Prueba con escala de Likert 2. Realizar animaciones en Scratch

En la misma prueba con escala de *Likert* se pidió a los participantes indicar (Muy importante 1- Sin importancia 5) respecto a “Mostrar el resultado del trabajo realizado”. Como se observa en la figura 6.19 los estudiantes se ubicaron en el Nivel 1, 13 estudiantes, Nivel 2, 2 estudiantes y Nivel 3 un estudiante. Mostrando de esta manera que a los estudiantes les motivó poder compartir sus creaciones con los compañeros.

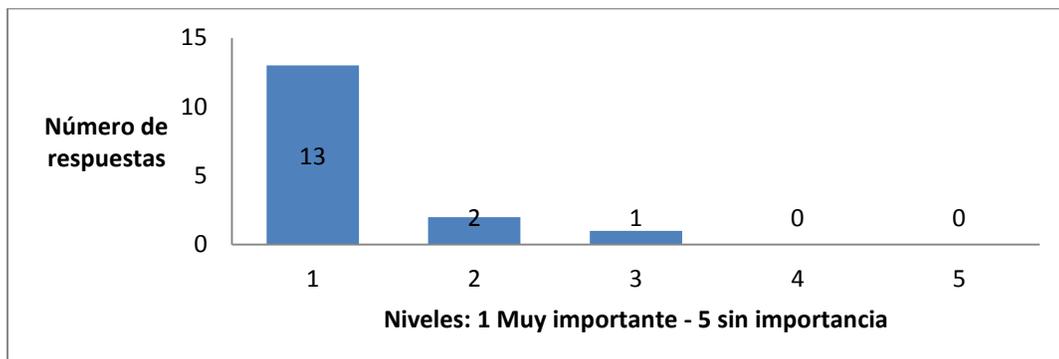


Figura 6.19 - Prueba con escala de Likert 2. Mostrar el resultado del trabajo realizado

6.3. Resultados del pretest - postest

Para reconocer aspectos concernientes al avance en el desarrollo de las habilidades, conocimientos y actitudes relacionadas con el pensamiento computacional en los estudiantes, se hizo un seguimiento mediante un pretest y un postest (ver ambos instrumentos en el Anexo II). Estos se estructuraron a partir de afirmaciones relacionadas con las características abordadas en el trabajo de

aula, de manera que fueran calificadas y se pudiera hacer un paralelo entre ambos momentos y de ahí sacar conclusiones al respecto.

A continuación, se hace su análisis:

Una de las afirmaciones a calificar por parte de los estudiantes se relacionó con el uso de la computadora en la solución de problemas. Para ello se utilizó la siguiente escala: Siempre **S**, Algunas veces **A**, y Nunca **N**. El resultado, como se presenta en la figura 6.20, muestra que mediante el trabajo realizado se logró que todos los estudiantes propusieran soluciones en alguna medida mediante la computadora, dejando el rango **N** sin puntuación. Cabe considerarse que en el postest se les aclaró que no consideraran la experiencia como parte del uso de la computadora para solucionar problemas. Esto es porque lo que se quería evaluar es si habían comenzado a considerar su uso en otras situaciones.

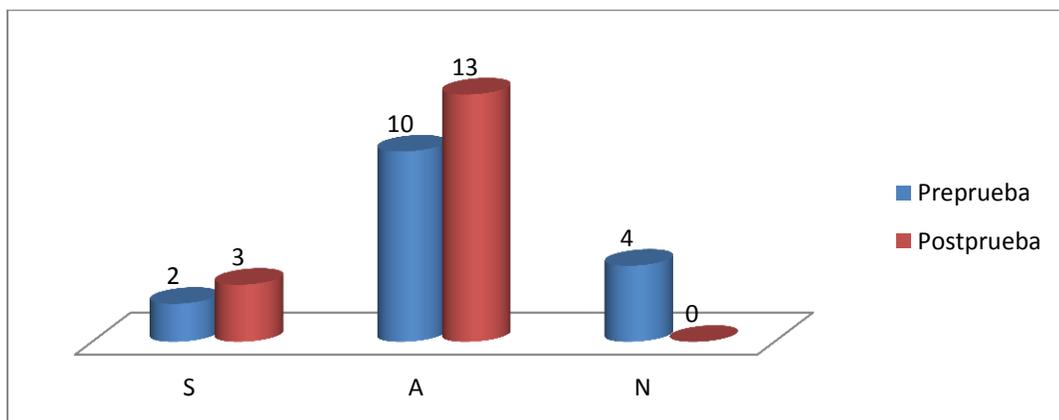


Figura 6.20 - Uso de la computadora en la solución de problemas

Para continuar, la afirmación que los estudiantes debían calificar estuvo relacionada con abstraer la información relevante para reconocer datos principales. Como se presenta en la figura 6.21 el resultado muestra que el logro alcanzado, luego de realizar la experiencia, es que una mayor cantidad de estudiantes consideren que pueden abstraer datos principales de diversa información.

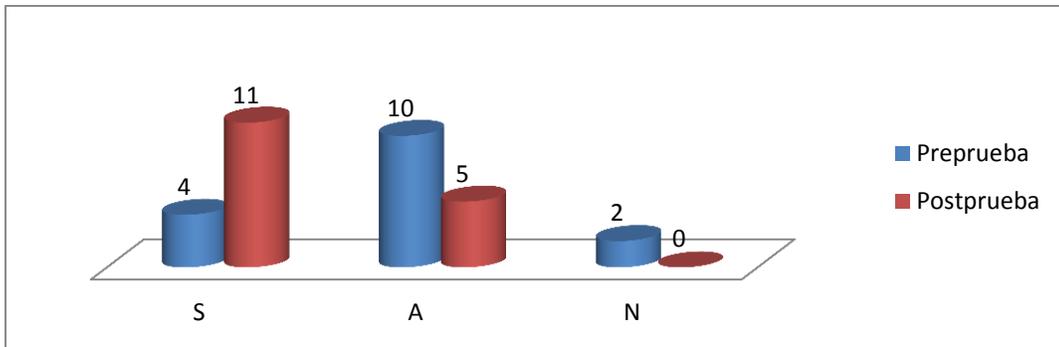


Figura 6.21 - Abstraer la información relevante para reconocer datos principales

Otra de las afirmaciones que se calificó, está relacionada con diseñar algoritmos para llegar de manera ordenada a la solución. Las respuestas de los participantes se muestran en la figura 6.22, comparando los resultados del pretest y postest. Para ello se utilizó la siguiente escala: Siempre **S**, Algunas veces **A**, y Nunca **N**. El resultado refleja que, en el postest los estudiantes consideran los valores de **S** y **A**, dejando el rango **N** sin puntuación. Este resultado puede estar relacionado con el desconocimiento inicial del concepto de algoritmo, que luego fue abordado a lo largo de las sesiones.

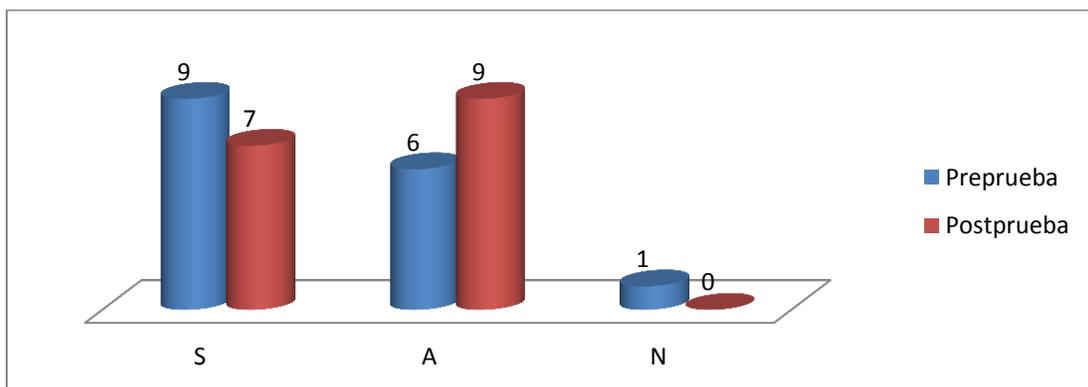


Figura 6.22 - Diseñar algoritmos para llegar de manera ordenada a la solución

Siguiendo con las afirmaciones calificadas, se evaluó una con relación a descomponer procesos o problemas en partes pequeñas y manejables. Para ello se utilizó la siguiente escala: Siempre **S**, Algunas veces **A**, y Nunca **N**. En este sentido, los estudiantes se ubican en los rangos como se muestra en la figura 6.23, correspondientes a pretest y postest. En este sentido se logra aumentar la

ubicación en el rango **S** y disminuir el rango **N**, dejándolo sin puntuación. Esta respuesta lleva a pensar que los estudiantes encontraron en la descomposición de tareas complejas, una herramienta para facilitar la solución a cierto tipo de problemas.

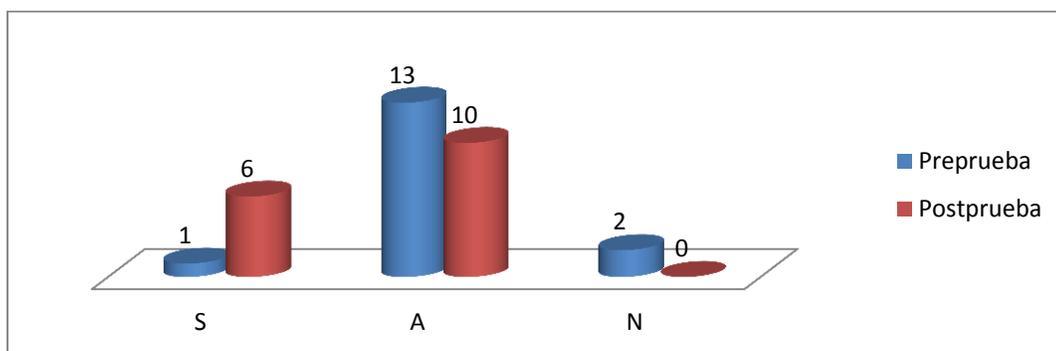


Figura 6.23 - Descomponer procesos o problemas en partes pequeñas y manejables

Con respecto a la afirmación que se vincula con la habilidad para comunicarse y trabajar con otros en pos de alcanzar una meta o solución común, los estudiantes manifestaron lo que se muestra en la figura 6.24. Para ello se utilizó la siguiente escala: Siempre **S**, Algunas veces **A**, y Nunca **N**. En este aspecto, se observa que los estudiantes se mantuvieron en el rango **S** y **A**, aunque en la postprueba, se nota mayor dificultad para llegar a acuerdos con los compañeros de trabajo, como se dijo anteriormente (Ver figura 6.2). Se observó en el trabajo con los chicos que entre mayor exigencia tiene la meta por cumplir, mayor es la dificultad para llegar a acuerdos. Esto se refleja luego de las sesiones en la postprueba.

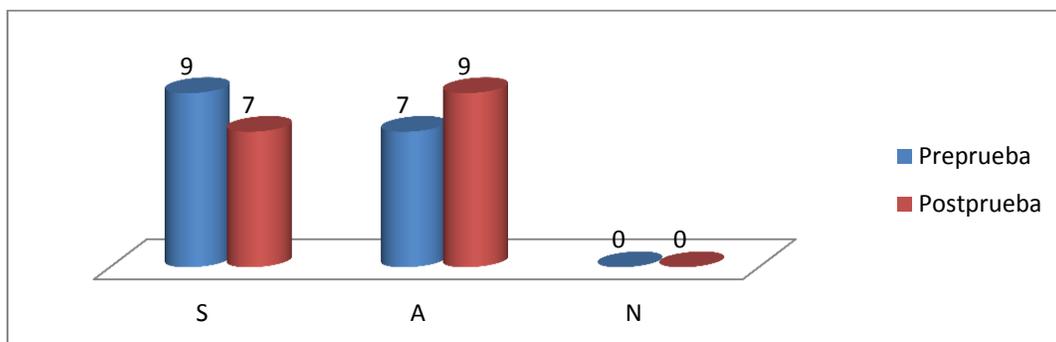


Figura 6.24 - Generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común

En el formato postest además de calificar las afirmaciones anteriores relacionadas a las habilidades abordadas, se pidió a los estudiantes que eligieran cuáles de un conjunto de alternativas corresponden a los aprendizajes alcanzados en las sesiones. Como se puede notar a continuación en la figura 6.25, todas las alternativas presentadas fueron calificadas. La mayoría de los estudiantes reconocen que lograron abordar las competencias que se plantearon inicialmente como objetivo de las sesiones.

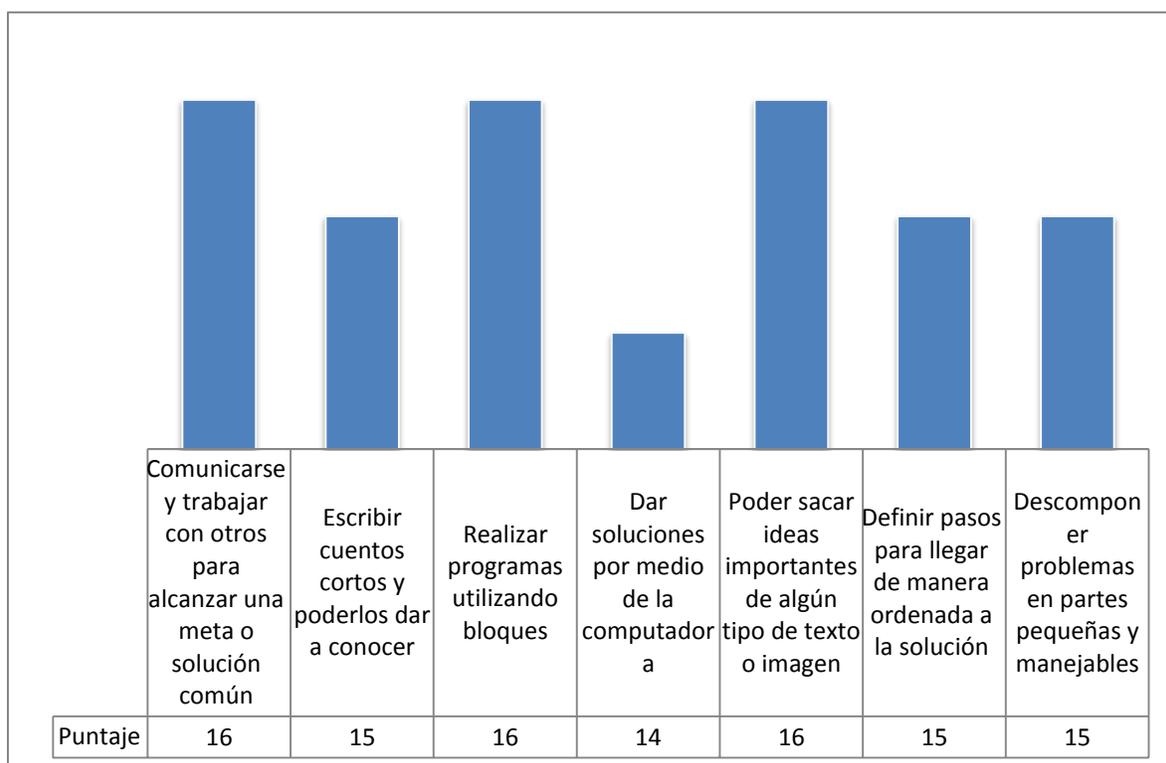


Figura 6.25 – Aprendizajes alcanzados en las sesiones

6.4. Resultados con relación a los objetivos del estudio de caso

Como se planteó en el capítulo anterior, el estudio pretende conocer los alcances y limitaciones de la metodología propuesta en el momento de ser utilizada y aplicada por un docente.

Según el análisis realizado a la información recopilada y a la observación de los registros videográficos, se puede señalar que la metodología es apropiada y acertada, en la medida que sigue un proceso organizado y con recomendaciones

que sirven en el momento de tomar decisiones. Las etapas de la metodología que guían el proceso de diseño, no implicaron un esfuerzo adicional, sino que mediante las preguntas orientadoras, se facilitó la tarea de la docente participante.

Además, las preguntas orientadoras consideran de manera suficiente los requisitos necesarios en el momento de diseñar actividades con TIC para el desarrollo del pensamiento computacional. Esto quedó parcialmente demostrado a partir de haber facilitado el diseño y la implementación completa de las sesiones llevadas a cabo con éxito.

Esta metodología también abre la posibilidad de trabajar transversalmente con otras áreas del conocimiento, pues las etapas a seguir en el diseño de actividades resultan claras para docentes de otras áreas. Aunque resulta necesario trabajar interdisciplinariamente en estas situaciones.

Por medio del uso de la metodología, se diseñaron de manera organizada y coherente siete sesiones de trabajo con sus respectivas actividades, en el momento de la implementación. Estas resultaron entretenidas, interesantes y apropiadas para el trabajo con los estudiantes del “Club cuenta un cuento con tecnología”, según se observó a través de la recogida de datos.

En cuanto a las limitaciones, se percibe que la metodología requiere contar con algún profesional cercano al área de tecnología, ya que es probable que se necesite poder utilizar apropiadamente los dispositivos y aplicaciones informáticas, que pudieran aportar al desarrollo del pensamiento computacional.

Este estudio pretende examinar si las actividades diseñadas a partir de la metodología fomentan el desarrollo de las características del pensamiento computacional que se abordan mediante la experiencia. En este caso y según la información analizada, se puede concluir que en cierta medida se logra que los estudiantes progresen en el desarrollo de las habilidades. En el pretest se observó que los estudiantes traían consigo conocimientos previos y algunas habilidades con relación al pensamiento computacional. Sin embargo, no eran capaces de reconocerlas ni utilizarlas de manera consciente. En la experiencia realizada se evidencia que, a través de las sesiones, los estudiantes reflexionan sobre este tipo

de habilidades, conocimientos y actitudes que les sirve para dar solución a problemas tanto académicos como del entorno.

Los resultados que se muestran en este capítulo están respaldados en primera medida desde la base conceptual, a partir de la cual se genera la metodología propuesta. Así se consideran los modelos como el *TPACK*, la propuesta de Manso et al. (2011).

6.5. Conclusiones

En este capítulo se analizan los resultados de las actividades desarrolladas en las sesiones a partir de la metodología propuesta en esta tesis. De esta manera, se puso en juego dicha metodología, en el marco de un estudio de caso, lo que implicó el trabajo conjunto con una docente del área de Lengua Castellana interesada

La aplicación de la iniciativa educativa se llevó a cabo por medio de siete sesiones, en las cuales se trabajó en actividades orientadas al desarrollo del pensamiento computacional por medio de ejercicios enfocados hacia la solución de problemas mediante la computadora, la abstracción de información para reconocer datos principales, el diseño de algoritmos, la descomposición de problemas y la generación de habilidades para trabajar con otros.

A partir del análisis de la información recopilada mediante distintas herramientas de registro, con relación al trabajo realizado entre docentes con la metodología propuesta, se puede considerar que:

- La metodología guio el diseño de las actividades y la experiencia educativa en su totalidad.
- La iniciativa despertó el interés en la docente de Lengua Castellana que participó.
- Las etapas resultaron fáciles de seguir y comprensibles por medio de las preguntas orientadoras.
- Los requisitos considerados en las etapas fueron precisos para cubrir las necesidades del grupo con el que se trabajó.
- El diseño generado resultó propicio para la aplicación en el aula.

Por otro lado, respecto al resultado de la experiencia con los estudiantes, se puede considerar que:

- Todas las actividades realizadas en las siete sesiones resultaron satisfactorias para los participantes.
- Aunque el trabajo en equipo resultó complejo, sobre todo en las actividades de mayor dificultad, los estudiantes lograron llegar a acuerdos y solucionar los desafíos propuestos.
- Los estudiantes se sintieron satisfechos al poder utilizar los dispositivos y espacios con que cuenta la institución.
- Los entornos de programación utilizados despertaron interés y agrado, el trabajo con Pilas Bloques y *Minecraft* Hora del Código les resultó fácil, mientras que en *Scratch* manifestaron algo de dificultad a la hora de programar su creación literaria.
- Las actividades planeadas para desarrollar habilidades lectoescritoras resultaron agradables e interesantes para los participantes, generando creaciones literarias acorde a su nivel escolar.
- El uso de bloques de programación al igual que los desafíos que debían solucionar, les pareció fácil en un comienzo y luego mostraron alguna dificultad al aumentar la complejidad de las actividades por realizar.
- Los participantes lograron abstraer información de forma fácil, al observar las escenas de los desafíos para utilizar esa información en la creación de obras literarias. De igual manera lograron tomar información importante de los cuentos creados, teniendo en cuenta los personajes existentes en *Scratch*, para luego animarlos y convertirlos en cuentos digitales.

Capítulo 7

Conclusiones

7.1. Introducción

El presente trabajo de investigación ha tenido como objetivo principal desarrollar una metodología que guíe el diseño de experiencias educativas para el desarrollo de competencias relacionadas con el pensamiento computacional. Esta metodología se pone en juego por medio de un estudio de caso en el cual se comprueban los alcances de dicha propuesta metodológica, haciendo una revisión de sus limitaciones y posibilidades.

Por medio del trabajo realizado en esta investigación y al analizar los resultados del diseño de actividades y la implementación en aula, se puede decir que los objetivos planteados se han alcanzado dado que:

- Se ha revisado y se describe cuidadosamente un conjunto de 12 iniciativas para la enseñanza de la programación o temas afines que abordan habilidades, conocimientos y actitudes relacionadas con el pensamiento computacional.
- Se han explorado diversos lenguajes de programación, varios de ellos utilizados en el marco de las iniciativas relevadas.
- Se ha realizado el diseño de una propuesta metodológica para el desarrollo de conocimientos, habilidades y actitudes relacionadas con el pensamiento computacional, que se pone en juego mediante un estudio de caso.
- Se ha realizado el análisis de los resultados obtenidos a partir del estudio de caso, que permiten revisar las limitaciones y posibilidades de la metodología.

Para llegar a la consecución de los objetivos de esta tesis, se realizó la revisión y descripción cuidadosa de iniciativas encontradas que se destacan por sus buenas prácticas en pro del desarrollo de competencias relacionadas con el pensamiento computacional y/o la enseñanza de la programación. De igual manera, se exploraron y caracterizaron algunos lenguajes de programación que también han sido utilizados en experiencias de este tipo. Esto permitió tener una

visión general de cómo posiblemente se podría organizar la metodología a proponer.

Teniendo en cuenta la información con respecto a metodologías y herramientas informáticas, junto con propuestas desde lo académico y pedagógico como el modelo *TPACK*, la propuesta de Manso et al. (2011) y el modelo de proceso de desarrollo de *software* en cascada, se diseña una metodología conformada por 4 etapas, que a partir de preguntas orientadoras guían la toma de decisiones con respecto a las estrategias, actividades y herramientas a utilizar en el aula. Al mismo tiempo, ofrece un acompañamiento para aquellos docentes que deseen planificar una propuesta de diseño de actividades de aula con TIC y su correspondiente evaluación.

Para continuar, por medio de un estudio de caso se implementó la propuesta metodológica, enfocándose en la eficacia de las etapas y preguntas orientadoras que guían el proceso de diseño. De igual manera se analizó cómo la metodología aportó para diseñar y alcanzar en la experiencia el desarrollo de ciertas habilidades del pensamiento computacional. Finalmente, se revisaron las barreras encontradas a lo largo de todo el proceso.

7.2. Aportes de la investigación

Esta investigación ha permitido:

- La revisión de 48 documentos que dio lugar a la caracterización de estrategias metodológicas para el desarrollo del pensamiento computacional, y la exploración y descripción de herramientas para la enseñanza de la programación.
- El estudio de modelos teóricos como el *TPACK*, la propuesta de Manso et al. (2011) y el modelo de proceso de desarrollo de *software* en cascada, que aportaron al diseño de la propuesta metodológica presente en este trabajo de tesis.
- El diseño de una metodología que guía la planificación de experiencias con TIC en el aula, que promueven el desarrollo del pensamiento

computacional. La misma ofrece un acompañamiento para aquellos docentes que desean planificar una propuesta de tal tipo.

- El diseño y puesta en marcha de un estudio de caso para evaluar los alcances de la metodología.

Esta tesis aporta:

- Una recopilación descriptiva de estrategias metodológicas relacionadas con la enseñanza de la programación o temas a fines, que abordan el desarrollo de competencias del pensamiento computacional.
- Una descripción de herramientas para la enseñanza de la programación, donde se detalla la facilidad de uso, disponibilidad, información técnica y requerimientos de cada una de ellas.
- Una metodología para el diseño de actividades de aula con TIC, para el desarrollo del pensamiento computacional.
- Unos productos como resultado de la implementación de la metodología propuesta, disponibles en: <https://cuentacuentotecno.blogspot.com/> y las creaciones literarias animadas por medio del lenguaje de programación *Scratch*, compartidas en la comunidad *Scratch* <https://scratch.mit.edu/>
- Difusión de resultados preliminares, obtenidos en la realización de esta investigación y dados a conocer por medio de la publicación de un artículo en congreso con el título de **Análisis comparativo de propuestas y experiencias en Iberoamérica que potencian el pensamiento computacional**, cuyo resumen se puede encontrar en: <http://www.grupokenta.co/archivos/memoriasCisiet2017.pdf>

7.3. Conclusiones

El desarrollo de la presente investigación permitió llegar a las siguientes conclusiones:

Respecto al diseño metodológico

La metodología propuesta guio el diseño de las actividades y la iniciativa educativa en general, logrando transversalizar dos áreas de conocimiento como

son el área de Lengua Castellana y el área de Tecnología. De este modo se puede decir que la metodología puede ser utilizada por un docente que esté interesado en el tema y que cuente con conocimientos tecnológicos o con asesoría en la parte tecnológica.

Las etapas y las preguntas orientadoras que forman parte de la metodología y que guían la planificación de las actividades de aula, resultaron apropiadas y fáciles de comprender y de seguir. En estas etapas se consideran los requisitos necesarios a tener en cuenta según el entorno en el cual se van a implementar las actividades. En el estudio de caso, esto fue valorado positivamente y se comprobó que se pudo alcanzar, de manera satisfactoria, el diseño de actividades con tecnologías digitales en el aula, con consideración del grupo destinatario y el contexto. Además, permitió durante las sesiones planificadas abordar las habilidades relacionadas con el pensamiento computacional, junto con las habilidades lectoescritoras que se optaron por desarrollar.

Respecto al trabajo con estudiantes en el estudio de caso

Se realizaron siete sesiones donde se aplicaron las actividades diseñadas, las cuales resultaron satisfactorias para los participantes: docentes y estudiantes. Se hizo uso efectivo de los dispositivos, espacios y herramientas informáticas con que cuenta la institución.

Una de las competencias que se buscó abordar fue: generar habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común. Con respecto a esta habilidad, los estudiantes mostraron dificultad para trabajar en equipo, especialmente en aquellas tareas con mayor grado de complejidad. Sin embargo, por medio de las actividades propuestas lograron remediar estas dificultades, para llegar a acuerdos y cumplir las metas planteadas.

En cuanto a la habilidad de diseñar algoritmos para llegar de manera ordenada a la solución, descomponer problemas en partes pequeñas y manejables, se realizaron actividades por medio de los desafíos en los entornos de programación Pilas Bloques, *Minecraft* Hora del Código y *Scratch*. Estas herramientas despertaron interés y agrado en los estudiantes, los dos primeros se utilizaron sin

inconveniente alguno y el tercero, *Scratch*, se utilizó con cierta dificultad en algunos casos. Al concluir las sesiones todos los estudiantes lograron animar sus creaciones literarias con la asesoría de la docente experta en el área, en este caso la tesista.

Por otro lado en el transcurso de las siete sesiones se trabajaron actividades (desafíos y creaciones literarias) que plantearon problemas factibles de solución haciendo uso de la computadora. Estas actividades lograron que los estudiantes propusieran soluciones con el uso de aplicaciones informáticas.

De igual manera, las actividades de lectoescritura lograron conjugar ejercicios en los cuales los estudiantes debían abstraer información importante de textos escritos o gráficos, como medio para conocer los datos principales y así cumplir las metas propuestas.

En síntesis, la iniciativa educativa, diseñada a partir de la metodología propuesta, logró fortalecer habilidades, conocimientos y actitudes relacionadas con el pensamiento computacional, de modo que estas características sean reconocidas y utilizadas por los estudiantes para solucionar problemas académicos y de su entorno. Al mismo tiempo, despertó en una docente de un área no tecnológica, el interés por desarrollar experiencias interdisciplinarias.

7.4. Líneas de trabajo o de investigación a futuro

Por los resultados favorables obtenidos en esta investigación con relación al trabajo transversal de las áreas y el desarrollo de competencias relacionadas con el pensamiento computacional, se hace importante buscar estrategias para ampliar el rango de impacto en la Institución Educativa Departamental Pío XII. De este modo, se espera avanzar en el trabajo con otras áreas de conocimiento y abrir espacios académicos donde interactúen mayor cantidad de estudiantes.

Otra línea de trabajo futuro está relacionada con la difusión de los resultados de esta investigación, de modo que despierte el interés en docentes que puedan aplicar la metodología propuesta en otras instituciones educativas. Para lograr la interacción con estos docentes, es de interés la creación de una red académica que permita la retroalimentación de las experiencias. De esta manera, se propone

también realizar una evaluación en profundidad de la metodología con el objetivo de hacer los ajustes que sean necesarios.

Referencias Bibliográficas

- Aybar, V., Querigua, C., y Banchoff, C. (2012). Aplicaciones complementarias a ROBOCODE que faciliten el aprendizaje de programación en escuelas secundarias. Facultad de Informática UNLP, 112. Abril 19 de 2016, desde: http://sedici.unlp.edu.ar/bitstream/handle/10915/47050/Documento_completo_.pdf?sequence=1
- Astudillo, G. J., Bast, S. G., y Willging, P. A. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación, 12(2015), 125–142. Consultado: Abril 10 de 2017, desde: <https://revistas.unc.edu.ar/index.php/vesc/article/view/14739>
- Barr, V., y Stephenson, C. (2011). Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? ACM Inroads, 2(1), 48–54. Consultado: Abril 1 de 2017, desde: <https://csta.acm.org/Curriculum/sub/CurrFiles/BarrStephensonInroadsArticle.pdf>
- Barrio, I., Gonzalez, J., Padín, L., Peral, P., Sánchez, I., y Tarín, E. (2011). Estudio de casos. Métodos de Investigación Educativa, 1–16. Consultado: Septiembre 10 de 2016, desde: https://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Est_Casos_doc.pdf
- Basel, V. (2012). Robótica pedagógica con software y hardware libre. 9 Jornadas Argentinas de Software Libre. Consultado: Agosto 9 de 2017, desde: http://www.conicet.gov.ar/new_scp/detalle.php?keywords=&id=29563&articulos=yes&detalles=yes&art_id=1862386
- Brennan, K., y Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Annual American Educational Research Association Meeting, Vancouver, BC, Canada, 1–25. Consultado:

Septiembre 17 de 2016, desde:
http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf

Capot, R. B., y Espinoza, R. M. (2015). Desarrollo del Pensamiento Computacional con Scratch, 616–620. Consultado: Agosto 22 de 2017, desde:
<http://www.tise.cl/volumen11/TISE2015/616-620.pdf>

Cataldi, Z., Lage, F., & Cabero, J. (2010). La promoción de competencias en el trabajo grupal con base en tecnologías informáticas y sus implicancias didácticas. *PixelBit. Revista de Medios y Educación*, julio-dici (37), 209–224. Consultado: Marzo 17 de 2017, desde:
[https://www.academia.edu/30066593/La Promoción De Competencias en El Trabajo Grupal Con Base en Tecnologías Informáticas y Sus Impli cancias Didácticas Promotion of Compences in Groupal Work Based on Informatic Technologies and This Didactic Consideratio](https://www.academia.edu/30066593/La_Promoción_De_Competicencias_en_El_Trabajo_Grupal_Con_Base_en_Tecnologías_Informáticas_y_Sus_Impli_cancias_Didácticas_Promotion_of_Compences_in_Groupal_Work_Based_on_Informatic_Technologies_and_This_Didactic_Consideratio)

Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., y Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *Revista de Educación a Distancia (RED)*, (46). Consultado: Mayo 3 de 2016, desde: <https://doi.org/10.6018/red/45/11>

Dapozo, G., Petris, R., Greiner, C., Espíndola, M. C., Company, A. M., y López, M. (2016). Capacitación en programación para incorporar el pensamiento computacional en las escuelas. *Teyet Revista Unlp*, 113–121. Consultado: Marzo 15 de 2017, desde: <http://teyet-revista.info.unlp.edu.ar/wp-content/uploads/2016/08/Capacitación-en-programación-para-incorporar-el-pensamiento-computacional-en-las-escuelas.pdf>

Dapozo, G., Greiner, C., y Petris, R. (2016). Herramientas lúdicas como apoyo a la enseñanza de la programación, 1311–1319. Consultado: Agosto 22 de 2017, desde: <http://sedici.unlp.edu.ar/handle/10915/56298>

da Silva, A., da Silva Martins, L. A., Ramos Nascimento, G., y do Santos Marints, thais. (2016). *Trabalhando Com Robótica Educacional No Ensino Fundamental*. (C. e T. do A. – I. C. P. Instituto Federal de Educação, Ed.). Consultado: Agosto 22 de 2017, desde: <http://www.tise.cl/volumen12/TISE2016/489-494.pdf>

Díaz, J., Banchoff, C., Martin, S., y López, F. (2012). *Aprendiendo a programar con juegos y robots*. Facultad de Informática, 6. Consultado: Septiembre 3 de 2016, desde: http://robots.linti.unlp.edu.ar/uploads/docs/aprendiendo_a_programar_con_juegos_y_robots_libro_.pdf

García, E., Sanchez, J., y Guerra, E. (2015). *Analizador estático de código Ruby*. Universidad Autonoma De Madrid, 55. Consultado: Marzo 25 de 2017, desde: https://repositorio.uam.es/xmlui/bitstream/handle/10486/668677/Garcia_Bravo_Esteban_tfg.pdf?sequence=1

Harari, V., y Banchoff, C. (2011). *Juegos educativos como nexo entre desarrollos de cátedra universitaria y necesidades escolares reales* Resumen Introducción. Consultado: Agosto 23 de 2017, desde: <http://sedici.unlp.edu.ar/handle/10915/18468>

Harari, V., y Tzancoff, C. (2014). *Desarrollando juegos educativos para incrementar la participación de los alumnos en una materia de programación*, 1–13. Consultado: Agosto 22 de 2017, desde: www.oei.es/historico/congreso2014/memoriactei/1178.pdf

Hernandez Sampieri, R., Fernandez Collado, C., & Baptista Lucio, M. del P. (2010). *Metodología de la investigación*. Metodología de la investigación. Consultado: Febrero 18 de 2017, desde: https://www.esup.edu.pe/descargas/dep_investigacion/Metodologia%20de%20la%20investigaci%C3%B3n%205ta%20Edici%C3%B3n.pdf

- ISTE, y CSTA. (2011). Operational Definition of Computational Thinking. Report, 1030054. Consultado: Abril 1 de 2016, desde: <https://csta.acm.org/Curriculum/sub/CompThinking.html>
- Izcará, P. S. P. (2007). Introducción al muestreo. México. Consultado: Abril 3 de 2017, desde: <http://libros.uat.edu.mx/omp/index.php/editorialuat/catalog/book/122>
- Kitchenham, B. (2004). Procedures for performing systematic reviews (Technical Report No. TR/SE-0401) (p. 27). Reino Unido: Keele University. Consultado: Marzo 4 de 2017, desde: http://testszingarelli.googlecode.com/svn-history/r336/trunk/2-Disciplinas/MetodPesquisa/kitchenham_2004.pdf.
- Koehler, M. J., y Mishra, P. (2015). ¿Qué son los Saberes Tecnológicos y Pedagógicos del Contenido (TPACK)? Virtualidad, Educación Y Ciencia, 10, 9–23. Consultado: Mayo 22 de 2017, desde: <https://revistas.unc.edu.ar/index.php/vesc/article/view/11552>
- López, R. N., Muñoz, R., y Barría, M. (2012). Un taller de robótica para el apoyo de la enseñanza de programación de computadores basado en estilos de aprendizaje, 336–343. Consultado: Abril 19 de 2017, desde: <http://www.tise.cl/volumen8/TISE2012/48.pdf>
- Lovos, E., Gibelli, T., y Bertone, R. (2014). Una Experiencia Explorando Python. Consultado: Agosto 22 de 2017, desde: <http://sedici.unlp.edu.ar/handle/10915/50639>
- Lugo, M. T., Toranzos, L., Lopez, N., & Corbetta, S. (2014). Políticas tic en los sistemas educativos de América Latina. Consultado: Agosto 22 de 2017, desde: <http://unesdoc.unesco.org/images/0023/002300/230080s.pdf>
- Lugo, M. (2015). Diálogo con M. Teresa Lugo. Avances en la integración de las TIC en los sistemas educativos latinoamericanos. Consultado: Mayo 12 de

2016, desde: <http://www.siteal.iipe.unesco.org/debates/521/dialogo-con-teresa-lugo-avances-en-la-integracion-de-las-tic-en-los-sistemas-educativo>

Magadán, C. (2012). Proyectos y tareas con TIC. Consultado: Septiembre 17 de 2016, desde: http://postitulo.secundaria.infed.edu.ar/archivos/repositorio/750/994/EyAT_clase4.pdf

Manso, M., Perez, P., Libedinsky, M., Light, D., y Garzón, M. (2011). Las TIC en las aulas: Experiencias latinoamericanas. Buenos Aires: Paidós. Consultado: Abril 22 de 2016, desde: <https://dialnet.unirioja.es/descarga/articulo/4902135.pdf>

Martínez López, P., Bonelli, E., y Sawady O'Connor, F. (2007). El nombre verdadero de la programación, 1–23. Consultado: Julio 20 de 2017, desde: <http://elaulayeltrabajo.proyectoslibres.unq.edu.ar/images/3/35/MartinezLopez-Bonelli-Sawady.pdf>

Ministerio de Educación Nacional de Colombia. (2014). e-Módulo 7. CREA-TIC. Construyendo capacidades en uso de tic para innovar en educación, 7, 35. Consultado: Mayo 15 de 2017, desde: <http://creatic.colombiaaprende.edu.co/emodulo/e-Modulo7.pdf>

Morales, P., y Landa, V. (2004). Aprendizaje Basado En Problemas. Theoria, 13(1), 145–157. Consultado: Marzo 15 de 2017, desde: <https://doi.org/10.4067/S0718-50062012000500003>

Muñoz, R., Barcelos, T. S., Villarroel, R., Barría, M., Becerra, C., Noel, R., y Silveira, I. F. (2015). Uso de Scratch y Lego Mindstorms como Apoyo a la Docencia en Fundamentos de Programación. XXI Jornadas de La Enseñanza Universitaria de Informática, 248–254. Consultado: Abril 22 de 2017, desde: http://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2015/mu_usod.pdf

- Murillo, J., Madera Payeta, A., Monasterio Martin, I., Jaraiz Lara, A., Cantador Gutierrez, R., Sánchez Sánchez, J. C., y Varas Moreno, R. (2010). Estudio de casos. Métodos de La Investigación Educativa. Universidad Autónoma de Madrid, 1–14. Consultado: Agosto 15 de 2017, desde: http://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/EstCasos_Trabajo.pdf
- Pozo, J. (2008). Aprendices y maestros: la psicología cognitiva del aprendizaje. Madrid: Alianza. pp. 121-148.
- Queiruga, C., Fava, L., Gómez, S., Kimura, I. M., y Bartneche, B. (2014). El juego como estrategia didáctica para acercar la programación a la escuela secundaria Resúmen JET : Java en Escuelas Técnicas secundaria, 358–362. Consultado: Abril 18 de 2016, desde: http://jets.linti.unlp.edu.ar/uploads/docs/wicc_2014.pdf
- Ramírez, M. del P., Lucio, M., Garza, J. J., García, L. del C., y Vargas, J. A. (2011). “alice”: un entorno diferente para aprender programación orientada a objetos, 64–68. Consultado: Junio 18 de 2016, desde: <http://www.revistaciencia.uat.edu.mx/index.php/CienciaUAT/article/view/58/46>
- Ram, A. O. (2010). Python como primer lenguaje de programación, 1–16. Consultado: Abril 18 de 2017, desde: http://webcem01.cem.itesm.mx:8005/publicaciones/primer_lenguaje_30_jun_2010.pdf
- Rederjo, J. (2013). Uso de AppInventor en la asignatura de Tecnologías de la Información y la Comunicación | Observatorio Tecnológico, 1–19. Consultado: Mayo 25 de 2017, desde: <http://recursostic.educacion.es/observatorio/web/en/software/programacion/1090-uso-de-appinventor-en-la-asignatura-de-tecnologias-de-la-comunicacion-y-la-informacion>

- Rios, P., y Cury, D. (2016). Utilizando O SCRATCH No Desenvolvimento De Lógica De Programação Como Contribuição Interdisciplinar. (U. F. do E. S.-UFES, Ed.). Consultado: Julio 8 de 2016, desde: <http://www.tise.cl/2016/img/Actas TISE 2016.pdf>
- Roa, O. L. (1985). 1. Ingeniería De Software, 1–34. Consultado: Mayo 10 de 2016, desde: http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:is1:01_lectura_ingenieria_software.pdf
- Rodriguez, C., León, C., Miranda, G., Segredo, E., y Segura, C. (2013). Prácticas de laboratorio en ruby para “lenguajes y paradigmas de programación.” Consultado: Junio 25 de 2017, desde: <https://dialnet.unirioja.es/servlet/libro?codigo=574268>
- Rosales, V. A., Queiruga, C., Kimura, I. M., Barnetche, M. B., y Gómez, S. (2015). Enseñando a programar con RITA en escuelas secundarias. Consultado: Septiembre 14 de 2016, desde: http://sedici.unlp.edu.ar/bitstream/handle/10915/50644/Documento_completo.pdf-PDFA.pdf?sequence=1
- Sarmiento, M., Gorga, G., y Sanz, C. (2016). Análisis de experiencias y estrategias educativas con TIC para el desarrollo del pensamiento computacional en estudiantes de secundaria y primeros años de universidad en Iberoamérica (trabajo final). Universidad Nacional de La Plata, La Plata, Buenos Aires. Consultado: Febrero 14 de 2018, desde: <http://sedici.unlp.edu.ar/handle/10915/60186>
- Sass, E. J. “Motivation in the College Classroom: What students tell us. Teaching of Psychology, 1989, 86-88. Consultado: Junio 15 de 2016, desde: http://journals.sagepub.com/doi/pdf/10.1207/s15328023top1602_15
- Saturio, L. M., García, S., y Hernandez, M. (2015). Videojuegos para aprender a programar videojuegos. Universidad Complutense de Madrid, 161.

- Consultado: Abril 19 de 2017, desde:
<https://eprints.ucm.es/32853/1/Memoria%20del%20Proyecto%20-%20Videojuegos%20para%20Aprender%20a%20Programar%20Videojuegos.pdf>
- Schroeder, M. (2014). Manual programa boom apps, 1–38. Consultado: Septiembre 5 de 2016, desde:
http://pruebasweb00.fod.ac.cr/Anterior/boomApps/assets/descargas/Tutorial_Moda.pdf
- Tzancoff, B., y Queiruga, C. (2014). Experiencias de la Facultad de Informática en la Enseñanza de Programación en Escuelas con Software Libre, 1–19. Consultado: Octubre 20 de 2016, desde:
www.oei.es/congreso2014/memoriactei/1426.pdf
- Vázquez-cano, A., y Delgado, F. (2015). La creación de videojuegos con Scratch en educación secundaria, 4, 63–73. Consultado: Agosto 1 de 2017, desde:
<https://dialnet.unirioja.es/servlet/articulo?codigo=5182831>
- Werner, L., Denner, J., Bliesner, M., y Rex, P. (2009). Can Middle-Schoolers use Storytelling Alice to Make Games? Results of a Pilot Study. Proceedings of the 4th International Conference on Foundations of Digital Games, 207–214. Consultado: Febrero 25 de 2017, desde:
<https://doi.org/10.1145/1536513.1536552>
- Wing, J. M. (2011). Computational Thinking: What and Why? Consultado: Abril 1 de 2016, desde:
<http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>
- Zúñiga, M., y Brenes, M. (2014). Estándares de desempeño de estudiantes en el aprendizaje con tecnologías digitales. FUNDACIÓN OMAR DENGO, 1–28. Consultado: Abril 14 de 2016, desde:
www.fod.ac.cr/estandares/docs/estandares_desempeno.pdf

ANEXO I

***Formato consentimiento informado
y ficha personal de estudiantes***

CONSENTIMIENTO INFORMADO - PADRES O ACUDIENTES DE ESTUDIANTES

Institución Educativa Departamental Pío XII
Docentes: Maira Sarmiento, Diana Ahumada

Yo _____
yo _____
o yo _____, mayor
de edad, [] madre, [] padre, [] acudiente o [] representante legal del estudiante
_____ de _____ años de edad,
he (hemos) sido informado(s) acerca de la participación de nuestro hijo en el Club de creación
literaria “Cuenta un cuento con tecnología” y la grabación de videos y entrevistas de práctica
educativa, las cuales se requieren para que la docente de tecnología Maira Sarmiento realice su
trabajo de tesis de Maestría y que a su vez servirá para que mi hijo (a) desarrolle habilidades de
lectoescritura que son muy importantes en su desempeño escolar. Este taller será apoyado por la
docente Diana Ahumada licenciada en español quién con su conocimiento en el tema hará el
seguimiento respectivo.

El club se reunirá los días lunes después de la jornada escolar.

Luego de haber sido informado(s) sobre las condiciones de la participación de mí (nuestro) hijo(a)
en las grabaciones y las entrevistas, resuelto todas las inquietudes y comprendido en su totalidad
la información sobre esta actividad, entiendo (entendemos) que:

- La participación de mi (nuestro) hijo(a) en este video y entrevista o los resultados
obtenidos por la docente en su tesis de maestría no tendrán repercusiones o
consecuencias en sus actividades escolares, evaluaciones o calificaciones en el curso.
- La participación de mi (nuestro) hijo(a) en el video y entrevista no generará ningún gasto,
ni recibiremos remuneración alguna por su participación.
- No habrá ninguna sanción para mí (nuestro) hijo(a) en caso de que no autoricemos su
participación.
- La identidad de mi (nuestro) hijo(a) no será publicada y las imágenes y sonidos registrados
durante la grabación se utilizarán únicamente para los propósitos y como evidencia de la
práctica educativa del docente.
- Las docentes garantizarán la protección de las imágenes de mi (nuestro) hijo(a) y el uso de
las mismas, de acuerdo con la normatividad vigente, durante y posteriormente al proceso
de investigación a realizar.

Atendiendo a la normatividad vigente sobre consentimientos informados, y de forma consciente y
voluntaria [] DOY (DAMOS) EL CONSENTIMIENTO [] NO DOY (DAMOS) EL CONSENTIMIENTO para
la participación de mi (nuestro) hijo (a) en el club “cuenta un cuento con la tecnología” en las
instalaciones de la Institución Educativa Departamental Pío XII sede B.

Lugar y Fecha: _____

FIRMA MADRE _____ CC/CE: _____

FIRMA PADRE _____ CC/CE: _____

FIRMA ACUDIENTE O REPRESENTANTE LEGAL _____

CC/CE: _____

**FORMATO DE FICHA PERSONAL
CLUB “CUENTA UN CUENTO CON TECNOLOGÍA”**

DATOS PERSONALES

Nombres y Apellidos: _____
 Fecha de nacimiento: _____ Edad: _____
 Dirección: _____
 Teléfono: _____

DATOS FAMILIARES

¿Actualmente vives con tu familia? SI NO
 Escribe los datos de las personas con las que vives:

N	Parentesco	Nombre	Edad	Sexo	Ocupación
1					
2					
3					
4					
5					
6					

INFORMACIÓN TECNOLÓGICA

Marca con una X

¿Cuentas con herramientas informáticas en tu casa?
 Computador _____ Celular _____ Tablet _____

¿Cuentas con conexión a Internet?
 Internet domiciliario _____ Móvil _____

¿Conoces algún entorno de programación? Si, ¿Cuál?
 Scratch _____ Kodu _____ Pilas Bloques _____ Otro _____
 No conozco ninguno _____

ANEXO II

Instrumentos para la recolección de información

FORMATO PREPRUEBA



INSTITUCIÓN EDUCATIVA DEPARTAMENTAL PÍO XII

CLUB “CUENTA UN CUENTO CON TECNOLOGÍA”

Nombre y apellido: _____ Grado _____

Edad _____

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige entre Siempre (**S**), Algunas veces (**A**), Nunca (**N**).

	S	A	N
Utilizo las computadoras para solucionar problemas			
Al hacer una lectura saco la idea principal con facilidad			
Cuando soluciono un problema organizo paso a paso lo que debo hacer para dar la solución			
Cuando tengo un trabajo escolar lo descompongo en varias partes y pido ayuda a mis compañeros para dar solución a cada una de esas partes (divide y vencerás)			
Cuando trabajo en grupo me comunico con mis compañeros fácilmente			
Utilizo entornos de programación			
Me reúno fuera del horario de clase con mis compañeros para aprender de tecnología			

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige 1 cuando sea Muy importante y 2, 3, 4 y 5 a medida que la actividad no sea tan importante para ti. (1 significa que es muy importante y 5 nada importante).

	MUY IMPORTANTE			SIN IMPORTANCIA	
	1	2	3	4	5
Formar parte de un club relacionado con el uso de la tecnología					
Aprender a utilizar las computadoras para solucionar problemas					
Trabajar en grupo para lograr metas comunes					
Aprender a comunicarse con los compañeros del grupo					
Aprender a solucionar problemas paso a paso y de manera organizada					
Sacar ideas principales de cualquier tipo de lectura (texto o imagen)					
Aprender a utilizar de entornos de programación					

FORMATO POSTPRUEBA



INSTITUCIÓN EDUCATIVA DEPARTAMENTAL PÍO XII

CLUB “CUENTA UN CUENTO CON TECNOLOGÍA”

Nombre y apellido: _____

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige entre Siempre (S), Algunas veces (A), Nunca (N).

	S	A	N
Si pudiera elegir entre trabajar solo o acompañado, elegiría trabajar acompañado			
Aunque no me lleve bien con los compañeros de equipo, podemos llegar a un acuerdo para realizar el trabajo encomendado			
Leyendo imágenes o textos puedo encontrar las ideas más importantes			
Descompongo un problema en parte pequeñas y así lo soluciono fácilmente			
Soluciono problemas mediante una secuencia de pasos ordenados			
Puedo solucionar algunos problemas utilizando una computadora			

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige 1 cuando sea Muy importante y 2, 3,4 y 5 a medida que la actividad no sea tan importante para ti. (1 significa que es muy importante y 5 sin importancia).

	MUY IMPORTANTE			SIN IMPORTANCIA	
	1	2	3	4	5
Llegar a acuerdos con los compañeros de grupo					
Seguir pasos ordenados para llegar a una meta					
Identificar y analizar posibles soluciones antes de dar la respuesta definitiva					
Utilizar las computadoras para solucionar desafíos o problemas					
Utilizar las computadoras para crear y compartir esas creaciones					

¿Qué cuestiones crees que aprendiste de la experiencia realizada? Marca una X junto a cada frase, puedes marcar una o varias opciones.

Comunicarse y trabajar con otros para alcanzar una meta o solución común	
Escribir cuentos cortos y poderlos dar a conocer	
Realizar programas utilizando bloques	
Dar soluciones por medio de la computadora	
Poder sacar ideas importantes de algún tipo de texto o imagen	
Definir pasos para llegar de manera ordenada a la solución	
Descomponer problemas en partes pequeñas y manejables	

PREGUNTAS ENTREVISTA SEMIESTRUCTURADA PARA ESTUDIANTES

SESIÓN 1



INSTITUCIÓN EDUCATIVA DEPARTAMENTAL PÍO XII

CLUB “CUENTA UN CUENTO CON TECNOLOGÍA”

Entrevista 1 (sesión1)

1. ¿Cómo te pareció el trabajo que hicimos hoy con tus compañeros? ¿por qué?
2. ¿Cuál fue la actividad que más te gustó? ¿por qué?
3. ¿Cuál fue la actividad que menos te gustó? ¿por qué?
4. ¿Te gustó trabajar con otro compañero? ¿por qué?
5. ¿Cómo te pareció el entorno de programación Minecraft? ¿pudiste solucionar los desafíos que se propusieron?
6. ¿Qué fue lo que más te gustó de este entorno de programación?
7. ¿El tiempo que te dieron fue suficiente para terminar los desafíos que debías trabajar en Minecraft?
8. ¿Te gustaría seguir trabajando y aprendiendo a programar en estos entornos donde arrastras y sueltas bloques?
9. ¿Cómo te quedó el inicio del cuento que escribiste? ¿fue fácil hacerlo?
10. ¿Cuál es el título de tu cuento? ¿Por qué elegiste ese nombre?

PREGUNTAS ENTREVISTA SEMIESTRUCTURADA PARA ESTUDIANTES SESIÓN 4



INSTITUCIÓN EDUCATIVA DEPARTAMENTAL PÍO XII **CLUB “CUENTA UN CUENTO CON TECNOLOGÍA”** **Entrevista 2 (sesión4)**

1. ¿Cómo te pareció el trabajo que hicimos hoy con tus compañeros? ¿por qué?
2. ¿Cuál fue la actividad que más te gustó? ¿por qué?
3. ¿Cuál fue la actividad que menos te gustó? ¿por qué?
4. ¿Te gustó trabajar con otro compañero? ¿por qué?
5. ¿Escribiste el inicio de tu cuento? ¿sobre qué lo hiciste? ¿por qué?
6. ¿Cómo te pareció el entorno de programación Scratch? ¿fue fácil o difícil utilizarlo?
7. ¿Qué fue lo que más te gustó y qué no te gustó de este entorno de programación?
8. ¿Pudiste animar el principio de tu cuento en Scratch? ¿lo terminaste?
9. ¿Viste el trabajo de tus demás compañeros? ¿Qué te pareció el trabajo que ellos realizaron? ¿de los trabajos que viste cuál te gustó más?

PRUEBA CON ESCALA DE LIKERT SESIÓN 4



INSTITUCIÓN EDUCATIVA DEPARTAMENTAL PÍO XII

CLUB “CUENTA UN CUENTO CON TECNOLOGÍA” Sesión 3

Nombre y apellido: _____

Elige una de las caritas que están junto a cada frase para indicar como te sentiste con el trabajo realizado en estas 3 primeras sesiones.

<i>Indica cómo te sentiste con respecto a:</i>	 <i>Muy bien</i>	 <i>Bien</i>	 <i>Regular</i>	 <i>Mal</i>	 <i>Muy mal</i>
Pertenecer al club Cuenta un cuento con tecnología					
Trabajar en la sala de informática					
Trabajar en equipos de 2 personas					
Utilizar Pilas Bloques y Minecraft					
Realizar una creación literaria					
Leer en voz alta el escrito					
La cantidad de tiempo trabajado					

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige 1 cuando sea muy fácil y 2, 3, 4 y 5 a medida que la actividad sea más difícil. (1 significa que es muy fácil y 5 muy difícil).

	FÁCIL					DIFÍCIL				
	1	2	3	4	5	1	2	3	4	5
Entender la explicación de la profe										
Tomar información de las escenas de Pilas Bloques y Minecraft para escribir el cuento										
Resolver los desafíos usando bloques de comando (sesión 1)										
Resolver los desafíos usando bloques de repetición (sesión 2)										
Resolver los desafíos usando bloques de alternativa condicional (sesión 3)										
Llegar a acuerdos con el compañero de equipo										
Compartir la creación literaria con todos los del club										

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige 1 cuando sea Muy importante y 2, 3, 4 y 5 a medida que la actividad no sea tan importante para ti. (1 significa que es muy importante y 5 nada importante).

	MUY IMPORTANTE			SIN IMPORTANCIA	
	1	2	3	4	5
Formar parte del club Cuenta un cuento con tecnología					
Poder trabajar con las computadoras					
Ver las explicaciones con ayuda del Video Beam					
Trabajar con entornos de programación con bloques (Pilas Bloques y Minecraft)					
Contar con un compañero de trabajo					
Contar con la ayuda de las profes					
Aprender a escribir una narración con base en las escenas de los desafíos					

PRUEBA CON ESCALA DE LIKERT SESIÓN 6



INSTITUCIÓN EDUCATIVA DEPARTAMENTAL PÍO XII

CLUB “CUENTA UN CUENTO CON TECNOLOGÍA” Sesión 6

Nombre y apellido: _____

Elige una de las caritas que están junto a cada frase para indicar como te sentiste con el trabajo realizado en estas sesiones.

<i>Indica cómo te sentiste con respecto a:</i>	 <i>Muy bien</i>	 <i>Bien</i>	 <i>Regular</i>	 <i>Mal</i>	 <i>Muy mal</i>
Crear un cuento con temática libre					
Aprender a utilizar Scratch					
Trabajar en equipos de 2 personas					
Utilizar Scratch para animar el cuento					
Compartir con los compañeros la animación creada					

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige 1 cuando sea muy fácil y 2, 3,4 y 5 a medida que la actividad sea más difícil. (1 significa que es muy fácil y 5 muy difícil).

	FÁCIL					DIFÍCIL				
	1	2	3	4	5	1	2	3	4	5
Entender la explicación de las profes										
Escribir el cuento con tema libre										
Utilizar bloques de comandos en Scratch										
Utilizar bloques de repetición en Scratch										
Utilizar bloques de alternativa en Scratch										
Llegar a acuerdos con el compañero de equipo										
Terminar la animación del cuento en esta sesión										

En el siguiente cuadro elige una casilla junto a cada frase y escribe una X según creas conveniente. Elige 1 cuando sea Muy importante y 2, 3,4 y 5 a medida que la actividad no sea tan importante para ti. (1 significa que es muy importante y 5 Sin importancia).

	MUY IMPORTANTE					SIN IMPORTANCIA				
	1	2	3	4	5	1	2	3	4	5
Conocer diferentes entornos de programación (Pilas bloques, Minecraft y Scratch)										
Aprender programación										
Crear cuentos con temas libres										
Trabajar en equipo con un compañero										
Realizar animaciones en Scratch										
Contar con la ayuda de las profes										
Mostrar el resultado del trabajo realizado										