



**UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES**

**CENTRO DE CIENCIAS BÁSICAS
DEPARTAMENTO DE SISTEMAS DE INFORMACIÓN**

TESIS

MOTIVACIÓN EN EL APRENDIZAJE DE LA PROGRAMACIÓN A NIVEL
BACHILLERATO UTILIZANDO UN LENGUAJE DE PROGRAMACIÓN EDUCATIVO

PRESENTA

L.I. Neira Yashira García Ortega

PARA OBTENER EL GRADO DE MAESTRA EN INFORMÁTICA Y
TECNOLOGÍAS COMPUTACIONALES

TUTOR

Dr. Carlos Argelio Arévalo Mercado

COMITÉ TUTORAL

Dra. Ma Loecelia Guadalupe Ruvalcaba Sánchez

Dra. Esthela Lizbeth Muñoz Andrade

Aguascalientes, Ags., Junio de 2016



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

NEIRA YASHIRA GARCÍA ORTEGA
MAESTRÍA EN INFORMÁTICA Y TECNOLOGÍAS COMPUTACIONALES
PRESENTE.

Estimada alumna:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis y/o caso práctico titulado: **“Motivación en el aprendizaje de la programación nivel bachillerato utilizando un lenguaje de programación educativo”**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

ATENTAMENTE

Aguascalientes, Ags., a 27 de mayo de 2016

“Se lumen proferre”

EL DECANO

A handwritten signature in black ink, appearing to read 'Jose de Jesus Ruiz Gallegos'.

M. en C. JOSE DE JESUS RUIZ GALLEGOS

c.c.p.- Archivo.



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

M. EN C. JOSÉ DE JESÚS RUÍZ GALLEGOS
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
PRESENTE

Por medio de la presente, como Director de Tesis designado de la estudiante L.I. **NEIRA YASHIRA GARCÍA ORTEGA** con ID **173438** quien realizó el trabajo de Tesis titulado: **MOTIVACIÓN EN EL APRENDIZAJE DE LA PROGRAMACIÓN A NIVEL BACHILLERATO UTILIZANDO UN LENGUAJE DE PROGRAMACIÓN EDUCATIVO** de la **Maestría en Informática y Tecnologías Computacionales**, me permito emitir **VOTO APROBATORIO** para la impresión de su documento de tesis, ya que cumple con los requisitos de contenido y forma exigidas en la Universidad Autónoma de Aguascalientes para continuar con el procedimiento administrativo para la obtención del grado.

Sin más por el momento, quedo a sus órdenes para cualquier aclaración al respecto, reciba un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., 25 de Mayo del 2016.

A handwritten signature in black ink, consisting of a large loop and several strokes extending to the left and right.

Dr. Carlos Argelio Arévalo Mercado
Director de Tesis





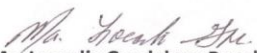
UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

M. EN C. JOSÉ DE JESÚS RUÍZ GALLEGOS
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
PRESENTE

Por medio de la presente, como Integrante del Comité Tutorial designado de la estudiante **L.I. NEIRA YASHIRA GARCÍA ORTEGA** con ID **173438** quien realizó el trabajo de Tesis titulado: **MOTIVACIÓN EN EL APRENDIZAJE DE LA PROGRAMACIÓN A NIVEL BACHILLERATO UTILIZANDO UN LENGUAJE DE PROGRAMACIÓN EDUCATIVO** de la **Maestría en Informática y Tecnologías Computacionales**, me permito emitir **VOTO APROBATORIO** para la impresión de su documento de tesis, ya que cumple con los requisitos de contenido y forma exigidas en la Universidad Autónoma de Aguascalientes para continuar con el procedimiento administrativo para la obtención del grado.

Sin más por el momento, quedo a sus órdenes para cualquier aclaración al respecto, reciba un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., 25 de Mayo del 2016.


Dra. Ma. Loecelia Guadalupe Ruvalcaba Sánchez
Integrante del Comité Tutorial



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

M. EN C. JOSÉ DE JESÚS RUÍZ GALLEGOS
DECANO DEL CENTRO DE CIENCIAS BÁSICAS
PRESENTE

Por medio de la presente, como Integrante del comité tutorial designado de la estudiante L.I. NEIRA YASHIRA GARCÍA ORTEGA con ID 173438 quien realizó el trabajo de Tesis titulado: **MOTIVACIÓN EN EL APRENDIZAJE DE LA PROGRAMACIÓN A NIVEL BACHILLERATO UTILIZANDO UN LENGUAJE DE PROGRAMACIÓN EDUCATIVO** de la **Maestría en Informática y Tecnologías Computacionales**, me permito emitir **VOTO APROBATORIO** para la impresión de su documento de tesis, ya que cumple con los requisitos de contenido y forma exigidas en la Universidad Autónoma de Aguascalientes para continuar con el procedimiento administrativo para la obtención del grado.

Sin más por el momento, quedo a sus órdenes para cualquier aclaración al respecto, reciba un cordial saludo.

ATENTAMENTE
"Se Lumen Proferre"
Aguascalientes, Ags., 25 de Mayo del 2016.

A handwritten signature in black ink, appearing to read 'Lizbeth'.

Dra. Esthela Lizbeth Muñoz Andrade
Integrante del Comité Tutorial

AGRADECIMIENTOS

A todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial al Dr. Carlos A. Arévalo Mercado, director de esta investigación, por la orientación, el seguimiento y la supervisión de la misma; así como al comité tutorial.

A los alumnos que participaron en el experimento y al personal académico del BACHUAA, en especial al Maestro José Eder Guzmán ya que sin su apoyo no hubiera sido posible el término de la investigación.

A mi familia, en especial a mi madre y abuelo Toño por el ánimo recibido y su apoyo incondicional. A ustedes que ahora forman parte de mi vida.

A Dios por permitirme culminar este proyecto.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) y a la Universidad Autónoma de Aguascalientes por el apoyo económico brindado durante el programa de Maestría.

ÍNDICE GENERAL

INTRODUCCIÓN 7

CAPÍTULO I – PROBLEMA DE INVESTIGACIÓN 9

 1.1 Importancia del aprendizaje de la programación 9

 1.2 Factores en la dificultad del aprendizaje de la programación 10

 1.3 Herramientas en la enseñanza de la programación básica 11

 1.4 Estado del arte..... 15

 1.3 Objetivos de investigación 17

 1.4 Preguntas de investigación 18

 1.5 Hipótesis..... 18

 1.6 Justificación 18

CAPÍTULO II – MARCO TEÓRICO 20

 2.1 Motivación 20

 2.1.1 Modelo motivacional de expectativas y valores 22

 2.1.2 Medición de la motivación (MSLQ)..... 27

 2.2 Resolución de Problemas 31

 2.3 Lenguajes de programación educativos (LPE) 32

 2.3.1 Scratch..... 33

 2.4 Justificación del uso de Scratch 35

 2.5 Contexto del Cuasi-Experimento..... 36

CAPÍTULO III – METODOLOGÍA 37

 3.1 Tipo y diseño de investigación 37

 3.2 Diseño del Cuasi-Experimento..... 37

 3.3 Población objetivo..... 39

 3.4 Prueba piloto noviembre 2015 39

 3.4.1 Pre prueba y post prueba de motivación 39

 3.4.2 Pre prueba y post prueba de programación básica 40

 3.4.3 Selección de participantes..... 41

3.4.4 Aplicación de las pre pruebas de motivación y de programación básica.....	42
3.4.5 Tratamiento experimental.....	42
3.4.6 Aplicación de las post pruebas de motivación y de programación básica	43
3.5 Replica marzo 2016.....	44
3.5.1 Calibración del experimento	44
3.5.2 Aplicación de las pre pruebas de motivación y de programación básica.....	44
3.5.3 Tratamiento experimental.....	45
3.5.4 Aplicación de las post pruebas de motivación y de programación básica	45
3.6 Técnicas estadísticas.....	45
3.6.1 Alfa de Cronbach.....	45
3.6.2 Prueba t para muestras relacionadas o dependientes.....	46
CAPÍTULO IV – RESULTADOS.....	49
4.1 Prueba piloto noviembre 2015	49
4.1.1 Resultados de información demográfica de los participantes	49
4.1.2 Resultados pre prueba de aprendizaje de la programación	49
4.1.3 Resultados post prueba de aprendizaje de la programación	50
4.1.4 Resultados pre prueba de motivación	52
4.1.5 Resultados post prueba de motivación.....	53
4.2 Réplica 2016.....	54
4.2.1 Resultados de información demográfica de los participantes	54
4.2.2 Resultados pre prueba de aprendizaje de la programación	55
4.2.3 Resultados post prueba de aprendizaje de la programación	55
4.2.4 Resultados pre prueba de motivación	57
4.2.5 Resultados post prueba de motivación.....	58
4.3 Análisis estadístico de los resultados.....	59
4.3.1 Prueba piloto 2015	59
4.3.2 Replica 2016	62
CONCLUSIONES	66
BIBLIOGRAFÍA	69
ANEXOS.....	80

ÍNDICE DE TABLAS

Tabla 1. Escala de motivación del MSLQ 28

Tabla 2. Estructura del curso diseñado 43

Tabla 3. Resultados información demográfica durante la prueba piloto..... 49

Tabla 4. Contraste post prueba - pre prueba de calificaciones durante la prueba piloto ... 51

Tabla 5. Alfa de Cronbach por sub-escala en la pre prueba durante la prueba piloto 52

Tabla 6. Media y desviación estándar por sub-escala en la pre prueba durante la prueba piloto 52

Tabla 7. Alfa de Cronbach por sub-escala en la post prueba durante la prueba piloto 53

Tabla 8. Media y desviación estándar por sub-escala en la post prueba durante la prueba piloto 53

Tabla 9. Resultados información demográfica durante la réplica..... 54

Tabla 10. Contraste post prueba - pre prueba de calificaciones durante la réplica 56

Tabla 11. Alfa de Cronbach por sub-escala en la pre prueba durante la réplica. 57

Tabla 12. Media y desviación estándar por sub-escala en la pre prueba durante la réplica 57

Tabla 13. Alfa de Cronbach por sub-escala en la post prueba durante la réplica 58

Tabla 14. Media y desviación estándar por sub-escala en la post prueba durante la réplica 58

Tabla 15. Datos estadísticos pre y post pruebas de aprendizaje de la programación durante la prueba piloto..... 60

Tabla 16. Resultados prueba t pre y post pruebas de aprendizaje de la programación durante la prueba piloto..... 60

Tabla 17. Datos estadísticos pre y post pruebas de motivación durante la prueba piloto . 61

Tabla 18. Resultados prueba t pre y post pruebas de motivación durante la prueba piloto 62

Tabla 19. Datos estadísticos pre y post pruebas de aprendizaje de la programación durante la réplica..... 63

Tabla 20. Resultados prueba t pre y post pruebas de aprendizaje de la programación durante la réplica..... 63

Tabla 21. Datos estadísticos pre y post pruebas de motivación durante la réplica 64

Tabla 22. Resultados prueba t pre y post pruebas de motivación durante la réplica..... 64

ÍNDICE DE FIGURAS

Figura 1. Scratch..... 13

Figura 2. Alice 13

Figura 3. Code.org - La hora del código 14

Figura 4. Blocky Game..... 15

Figura 5. Componentes de la motivación (McKeachie & others, 1987)..... 23

Figura 6. Modelo de resolución de problemas de Polya, adaptado de: (“Synthesis of Research on Problem Solving”, s/f))..... 31

Figura 7. Interfaz de Scratch 34

Figura 8. Diseño de la investigación 37

Figura 9. Metodología durante la prueba piloto 38

Figura 10. Ejemplo pre-prueba y post-prueba de programación básica durante la prueba piloto 41

Figura 11. Calificaciones de la pre prueba durante la prueba piloto 50

Figura 12. Calificaciones de la post prueba durante la prueba piloto 50

Figura 13. Calificaciones de las pre y post pruebas durante la prueba piloto..... 51

Figura 14. Resultados de las pre y post pruebas de motivación durante la prueba piloto .54

Figura 15. Calificaciones de la pre prueba durante la réplica 55

Figura 16. Calificaciones de la post prueba durante la réplica..... 56

Figura 17. Calificaciones de las pre y post pruebas durante la prueba piloto..... 56

Figura 18. Resultados de las pre y post pruebas de motivación durante la réplica 59

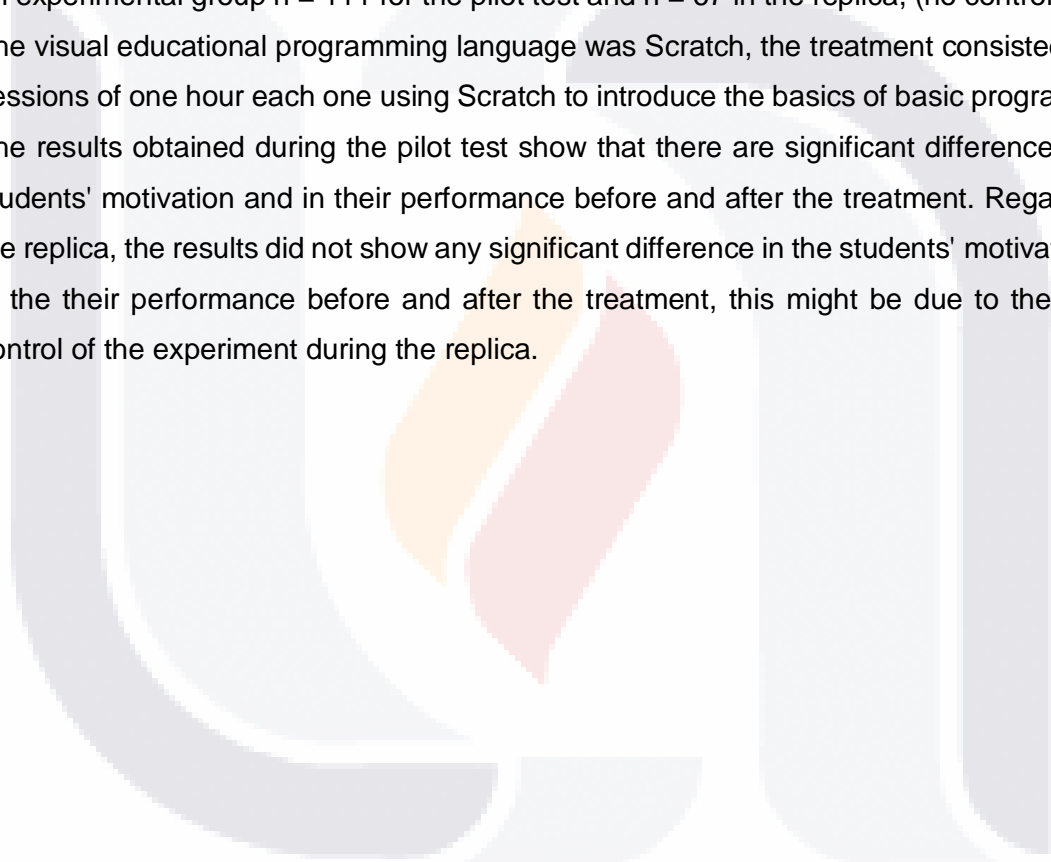
Figura 19. Medias pre y post pruebas de aprendizaje de la programación durante la prueba piloto 61

RESUMEN

La presente investigación consistió en un cuasi-experimento con el objetivo de identificar si el uso de un lenguaje de programación educativo (LPE) visual produce diferencias estadísticamente significativas en la motivación y en la adquisición de habilidades de resolución de problemas en el ámbito del aprendizaje de la programación básica en estudiantes de educación media superior. El cuasi-experimento se llevó a cabo en el BACHUAA con alumnos de quinto semestre para la prueba piloto y posteriormente de sexto semestre para la réplica, la edad promedio de los participantes es de 17 años contando con un grupo experimental de $n = 111$ para la prueba piloto y $n = 67$ en la réplica, sin grupo control. El LPE utilizado fue Scratch, el tratamiento consistió de cinco sesiones de una hora cada una de ellas, en el cual se hizo uso de Scratch para introducir los conceptos básicos de programación básica. Los resultados obtenidos durante la prueba piloto muestran que existen diferencias significativas en el grado de motivación de los alumnos así como en su desempeño antes y después del tratamiento. Respecto a la réplica, los resultados no mostraron diferencias estadísticamente significativas en el grado de motivación ni en desempeño de los estudiantes antes y después del tratamiento, esto pudo deberse a la falta de control del experimento durante la réplica.

ABSTRACT

A quasi-experiment was conducted in this research in order to identify whether the use of a visual educational programming language produces statistically significant differences in motivation and the acquisition of skills of problem solving in the field of learning basic programming in higher education students. The quasi-experiment was carried out in the BACHUAA fifth semester participants in the pilot test and sixth semester participants in the replica, the participants' average age was 17 years old. The quasi-experiment consisted on an experimental group $n = 111$ for the pilot test and $n = 67$ in the replica, (no control group). The visual educational programming language was Scratch, the treatment consisted of five sessions of one hour each one using Scratch to introduce the basics of basic programming. The results obtained during the pilot test show that there are significant differences in the students' motivation and in their performance before and after the treatment. Regarding in the replica, the results did not show any significant difference in the students' motivation and in their performance before and after the treatment, this might be due to the lack of control of the experiment during the replica.



INTRODUCCIÓN

A pesar de que los estudiantes del siglo XXI han crecido en un mundo predominantemente digital, siguen enfrentando dificultades en el estudio de las ciencias de la computación, específicamente en la programación de computadoras (Malliarakis, Satratzemi, & Xinogalos, 2014). Estudios realizados en las últimas dos décadas indican que estas dificultades siguen presentes y los estudiantes están menos interesados en programación (Lahtinen, Ala-Mutka, & Järvinen, 2005).

Se ha documentado en la literatura que la programación de computadoras es un tema difícil para estudiantes de nivel superior (Jenkins, 2002; Kelleher, 2009). Sin embargo, existe un movimiento social en países desarrollados que busca acercar la programación a los estudiantes de nivel básico e intermedio (primaria, secundaria y bachillerato), haciendo énfasis en la educación STEM (Science, Technology, Engineering and Mathematics) (An & Park, 2011; Burke & Kafai, 2010; Kammer, Brauner, Leonhardt, & Schroeder, 2011; Malliarakis, Satratzemi, & Xinogalos, 2014; Rodger et al., 2010; Shannon & Ward, 2014), es decir el aprendizaje de Ciencia, Tecnología, Ingeniería y Matemáticas, además existen iniciativas que tienen como objetivo promover la integración de la enseñanza de la programación, algunas de estas son code.org y codeweek (“Code.org”, 2015, “Codeweek”, 2016).

Por otro lado, se ha identificado que la habilidad para resolver problemas de programación, comprende principalmente aspectos cognitivos, metacognitivos y motivacionales. (Mayer, 1998; Parham, Gugerty, & Stevenson, 2010). Como se menciona en (Park & Hyun, 2014) la motivación y las habilidades son parte importante para el éxito en la programación de computadoras, además de la autoeficacia, la cual esta correlacionada positivamente con las habilidades computacionales y afecta el rendimiento académico en la programación de computadoras (Askar & Davenport, 2009).

De acuerdo con (Jenkins, 2002) los lenguajes de programación con los que se pretende enseñar a programar no están diseñados para tal objetivo. Algunos investigadores han optado por un enfoque lúdico para desarrollar las habilidades de programación en niños y jóvenes mediante lenguajes de programación educativos (LPE), entre los cuales se encuentran Logo (An & Park, 2011; Serafini, 2011), Alice (Kelleher, 2009; T. Wang, Mei, Lin, Chiu, & Lin, 2009), Kodu (Hui & Tam, 2014) y Scratch (Baytak & Land, 2011; García &

Peña, 2014; Garneli, Giannakos, Chorianopoulos, & Jaccheri, 2013; Giannakos, Jaccheri, & Leftheriotis, 2014; Høiseth & Jaccheri, 2011; McInerney, 2010; Nikou & Economides, 2014; L.-C. Wang & Chen, 2010).

Con esta investigación se pretende identificar si el uso de un LPE genera diferencias estadísticamente significativas en la motivación de los estudiantes y en la adquisición de habilidades para la resolución de problemas en el contexto de la programación básica a nivel bachillerato.

El capítulo I “Problema de investigación” se refiere a la importancia del aprendizaje de la programación y los factores que influyen en el aprendizaje de la misma, además se mencionan algunos artefactos en la enseñanza de la programación básica así como estudios relacionados. El capítulo II “Marco teórico” corresponde a los fundamentos teóricos en los que se basó la investigación. En el capítulo III “Metodología” se especifica el tipo de estudio llevado a cabo, así como el material y herramientas en él utilizados. En el capítulo IV “Resultados” se presentan los resultados obtenidos al analizar la información recabada mediante los instrumentos de medición utilizados conforme a las necesidades de la investigación. Por último se presentan las conclusiones a las que se llegó de acuerdo a los resultados obtenidos, además de las limitaciones de la investigación.

CAPÍTULO I – PROBLEMA DE INVESTIGACIÓN

En el presente capítulo se expone la importancia de la programación y algunos factores identificados en la literatura, que intervienen en el aprendizaje de la programación, así como artefactos de uso en la enseñanza de la programación y estudios relacionados respecto al tema.

1.1 Importancia del aprendizaje de la programación

A pesar de que los estudiantes del siglo XXI han estado creciendo en un mundo altamente digital continúan enfrentando dificultades en cursos de ciencias de la computación tales como la programación (Malliarakis, Satratzemi, & Xinogalos, 2014).

La demanda de profesionales en tecnologías de la información continúa creciendo, pero el número de estudiantes que seleccionan computación ha disminuido (Doerschuk, Liu, & Mann, 2012; Kammer et al., 2011; “Univision”, s/f), por lo tanto uno de los objetivos planteados es optimizar la enseñanza en las escuelas en calidad y cantidad con el fin de mejorar el interés de los jóvenes en la ciencia, tecnología, ingeniería y matemáticas, STEM por sus siglas en inglés (Kammer et al., 2011). La organización code.org, la cual promueve la enseñanza de la programación, cree necesario que las ciencias de la computación así como la programación deben ser parte del programa educativo junto a otros cursos de STEM (“Code.org”, 2015).

De acuerdo a Grover, citado en (Suárez, Andrés, & Sarmiento Porras, 2015) el objetivo de la enseñanza de la programación no es solo enseñar una sintaxis particular para escribir secuencias de instrucciones con el fin de que la computadora las ejecute, sino que pretende centrar la atención en conceptos fundamentales del pensamiento computacional o algorítmico, el cual permite que los estudiantes establezcan una serie de pasos para dar solución a un problema.

La disminución de las dificultades en el aprendizaje de la programación es importante, ya que la enseñanza y el aprendizaje exitoso de la programación puede ser beneficioso para los estudiantes del siglo XXI, puesto que la programación permite no solo el entendimiento de un lenguaje de programación, sino también el desarrollo de varias competencias tales como la creatividad, el pensamiento crítico (permitiendo a los estudiantes crear sus propios programas) (An & Park, 2011; Malliarakis et al., 2014; “Scratch”, 2015), el análisis y la

TESIS TESIS TESIS TESIS TESIS

resolución de problemas puesto que estas dos últimas competencias suelen ser necesarias para descifrar un determinado escenario e interpretarlo en líneas de código (An & Park, 2011; Malliarakis et al., 2014), además los estudiantes aprenden a trabajar en grupos y colaborar unos con otros en su esfuerzo para desarrollar programas ejecutables mientras intercambian conocimientos e ideas (Malliarakis et al., 2014; “Scratch”, 2015).

Por las razones anteriores y de acuerdo a Law, Lee y Yu (2010) la programación permite a los estudiantes convertirse en aprendices de por vida, la cual es una característica cada vez más importante en la era del conocimiento permitiéndoles transferir sus habilidades no solo a campos de la computación sino también a otros dominios (An & Park, 2011; Malliarakis et al., 2014).

1.2 Factores en la dificultad del aprendizaje de la programación

La programación de computadoras requiere tanto de habilidades matemáticas, como de habilidades para la resolución de problemas (Jenkins, 2002). Puede ser muy difícil para estudiantes iniciales de todas las edades, ya que además de los retos de aprender a formular soluciones estructuradas a problemas y entender cómo los programas son ejecutados, los estudiantes iniciales tienen que aprender una sintaxis rígida, la cual puede tener palabras confusas (Kelleher & Pausch, 2003). Deben ser capaces de utilizar la computadora eficientemente, crear programas en un archivo, compilarlo, obtener el resultado, probarlo, encontrar defectos en él y corregirlos (Jenkins, 2002).

De acuerdo a (Dehnadi & Bornat, 2006), a pesar de los cambios que han tenido lugar desde que la computación electrónica fue inventada, la mayoría de las personas no son capaces de programar y entre el 30% y 60 % de los estudiantes de los departamentos universitarios de informática fallan en el primer curso de programación.

Existen varios factores involucrados en la dificultad del aprendizaje de la programación, algunos son inherentes a la materia en cuestión, mientras que otros están relacionados con el *modus operandi* de los departamentos de enseñanza, otros factores están profundamente relacionados con expectativas, aptitudes y experiencia previa tanto de los profesores como de sus estudiantes (Jenkins, 2002), así como del mal enfoque de enseñanza entre las generaciones más jóvenes (Ogar, Shabalina, Davtyan, & Kizim, 2014).

TESIS TESIS TESIS TESIS TESIS

Factores adicionales relacionados con la dificultad del aprendizaje de la programación son la dificultad que esta implica (Kelleher & Pausch, 2003) debido a aspectos citados en (Ogar et al., 2014) que argumentan que la programación requiere de comprensión de *conceptos abstractos*, *pensamiento lógico*, solución de problemas y un alto nivel de abstracción; mientras que el autor Jenkins, señala que no hay dificultad, sino que los estudiantes no tienen la aptitud, a pesar de lo que argumenta, el autor sugiere que para entender la dificultad de aprender a programar se deben abordar temas cognitivos del proceso de aprendizaje, de los cuales menciona la motivación (Jenkins, 2002) y estilo de aprendizaje (Jenkins, 2002; Kelleher & Pausch, 2003).

Entre otros factores que contribuyen a la dificultad de aprender a programar se encuentran: la necesidad de múltiples habilidades y múltiples procesos, novedad educativa, el lenguaje de programación en el que se pretende enseñar, interés de los estudiantes, reputación e imagen de la programación (Jenkins, 2002) así como la mecánica de la programación (Jenkins, 2002; Kelleher & Pausch, 2003), además del ritmo del curso (Jenkins, 2002) ya que, como se menciona en (Ogar et al., 2014) los estudiantes piensan que necesitan más tiempo de práctica para dominar efectivamente la programación. Otro de los factores que influye según los autores del artículo "*What Emotions Do Novices Experience during Their First Computer Programming Learning Session?*" es el factor emocional (Bosch, D'Mello, & Mills, 2013).

Esta investigación se basó en los factores **motivación** y **lenguaje de programación** haciendo uso de un Lenguaje de Programación Educativo (LPE).

1.3 Herramientas en la enseñanza de la programación básica

La creación de herramientas que tienen como fin acercar la programación a personas de todas las edades data desde principios de 1960; los investigadores han construido lenguajes de programación y entornos, con el propósito de hacer la programación accesible a un número mayor de personas (Kelleher & Pausch, 2003). Existe evidencia de que se logran mejores resultados al hacer uso de herramientas y lenguajes didácticos cuyo objetivo es facilitar el aprendizaje de los conceptos de programación (Sovic, Jagust, & Sersic, 2014).

Los autores Kelleher y Pausch (2003) describen una taxonomía en la cual dividen los sistemas para el aprendizaje de la programación en dos grandes grupos: sistemas de enseñanza y sistemas que intentan apoyar el uso de la programación en diferentes campos.

Los sistemas de enseñanza, los que interesan en esta investigación, fueron diseñados con el fin de ayudar a las personas a aprender a programar, la mayoría de estos son o incluyen herramientas de programación sencillas que proporcionan a los programadores iniciales algunos aspectos fundamentales del proceso de programación. Algunos de estos sistemas intentan proporcionar apoyo en el aprendizaje de un lenguaje de programación más general, por esta razón los sistemas de enseñanza tienen similitudes con los lenguajes de propósito general como Java, C o C++.

Los sistemas de enseñanza se enfocan en áreas que pueden ser difíciles para los programadores principiantes. La mayoría de ellos abordan la mecánica de la programación: cómo dar instrucciones a la computadora (sintaxis), cómo organizar esas instrucciones (estilo de programación) y cómo hacer que la computadora las ejecute, algunos son: BASIC, Turing, GRAIL, GNOME, Play, LogoBlocks, Alice 2, LegoSheets, Leogo, Pascal, Smalltalk y Karel. Algunos otros intentan colocar a la programación en un contexto accesible y motivador a un público más amplio de personas ya sea proporcionando razones concretas para programar o incitando el trabajo conjunto de los programadores iniciales para que aprendan unos de otros, entre ellos se encuentran AlgoBlock, AlgoArena y Robocode (Kelleher & Pausch, 2003).

Algunas de las herramientas más utilizadas en la enseñanza de la programación actualmente son Alice y Scratch, esta última cuenta con versión web. En seguida se describen brevemente.

Scratch. Es un proyecto del grupo Lifelong Kindergarten del laboratorio de medios del MIT, comenzó en 2006, cuenta con una gran comunidad de usuarios que comparten proyectos y experiencias, se ofrece de forma gratuita, está basado en el paradigma de “arrastrar y soltar” bloques de código que encajan unos con otros. Scratch permite la creación de historias interactivas, juegos y animaciones, ayuda a jóvenes a aprender a pensar creativamente, razonar sistemáticamente y trabajar colaborativamente, las cuales son habilidades esenciales del siglo XXI (“Scratch”, 2015), la Figura 1 corresponde a la pantalla de Scratch en la versión de escritorio y versión web respectivamente.

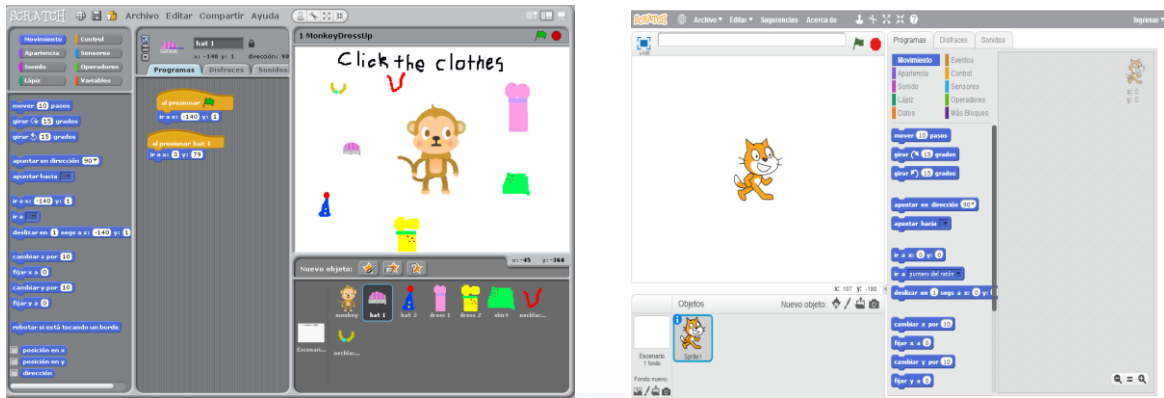


Figura 1. Scratch. Fuente: Scratch, Scratch.mit.edu

Alice. Entorno de programación en 3D. Es una herramienta de enseñanza gratuita diseñada para acercar a los estudiantes a la programación orientada a objetos, fue desarrollado en la Universidad Carnegie Mellon y está programado en Java. El entorno de Alice, al igual que Scratch se basa en el paradigma de “arrastrar y soltar”. Permite que los estudiantes aprendan los conceptos fundamentales de programación mediante la creación de películas animadas y videojuegos simples. Cuenta con un gran número de objetos pre-programados que los estudiantes pueden combinar para la creación de animaciones e historias en un mundo virtual (“Alice.org”, s/f). En la Figura 2 se muestra la pantalla de Alice.

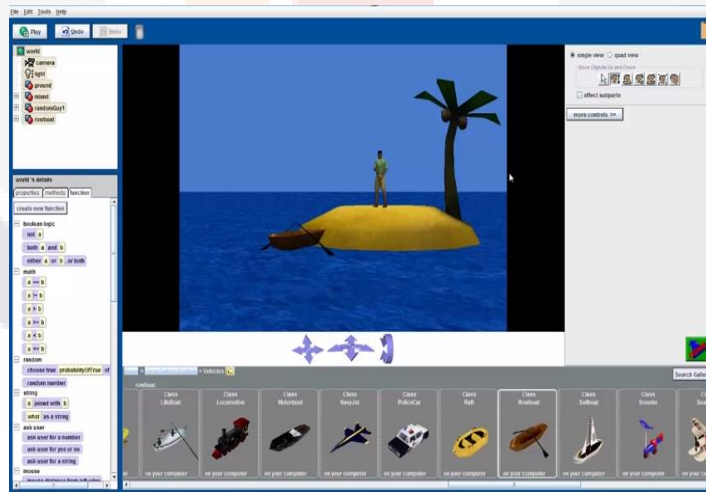


Figura 2. Alice. Fuente: Alice

Además de las herramientas mencionadas y descritas anteriormente existen varias iniciativas que intentan acercar la programación a personas de todas las edades, algunas de ellas son code.org, codeweek, y Blockly Game.

Code.org. Es una organización sin ánimo de lucro cuyo objetivo es promover la integración de la enseñanza de la programación informática en los planes de estudio. Fue creada en 2013 por *Ali y Hadi Partovi*. Su sitio web es code.org, en el cual se puede practicar la programación de manera divertida, se pueden compartir proyectos y ver proyectos de otros usuarios. Está basado en el paradigma “arrastrar y soltar” al igual que Scratch. Su campaña *La Hora del código* ha tenido un enorme éxito de participación (“Code.org”, 2015). En el mes de diciembre de 2015, la hora del código en México se organizó en el estado de Aguascalientes mediante el CECOI (Centro de Competitividad e Innovación del estado de Aguascalientes) (“Hora de Código - Mexico”, s/f). En la Figura 3 se muestra el sitio web de code.org correspondiente a la hora del código, en esta ocasión se muestra el entorno de Star Wars, ya que Code.org está a la vanguardia.

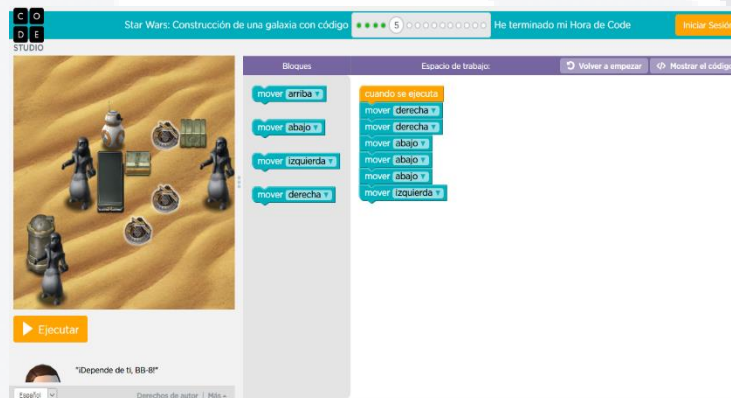


Figura 3. La hora del código. Fuente: “Code.org”, 2015

Blockly Game. Es un proyecto de Google para animar a los programadores del mañana. Es una serie de juegos educativos que enseñan programación, los cuales están diseñados para avanzar al ritmo de cada persona de acuerdo a su autoaprendizaje. Está diseñado para personas que no han tenido experiencia previa con la programación de computadoras. Al terminar los juegos, los usuarios están listos para usar lenguajes basados en texto convencionales (“Blockly Games”, s/f). La Figura 4 corresponde a la pantalla de un ejemplo en Blockly Games.

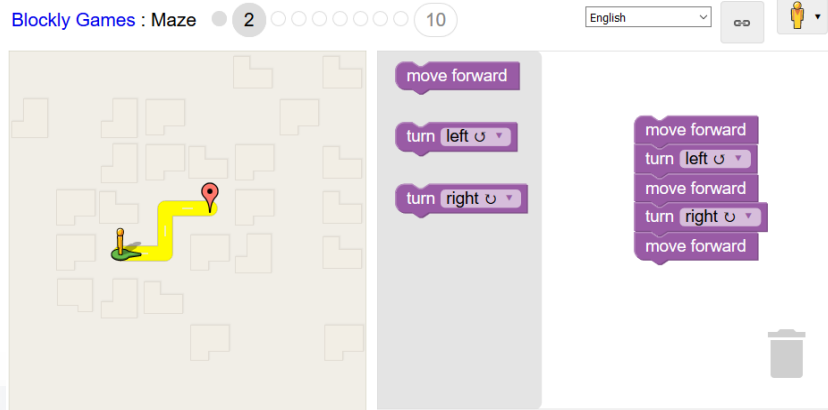


Figura 4. Blockly Game. Fuente: “Blockly Games”, s/f

Codeweek. En la Unión Europea también se han desarrollado proyectos con objetivos similares a la *Hora del código* como *La Semana Europea de la Programación* la cual es una semana en la que se organizan actividades y eventos en los que será posible introducirse en el mundo de la programación de una forma divertida y emocionante (codeweek.eu/) con el fin de potenciar la competencia digital en los jóvenes y así facilitarles su futura incorporación al mundo laboral. En el 2016 la semana europea de la programación se llevará a cabo del 15 al 23 de octubre (“codeweek”, 2016).

1.4 Estado del arte

En México así como a nivel internacional, se han llevado a cabo investigaciones, con el fin de disminuir los porcentajes de deserción en materias relacionadas con la programación, tanto en bachillerato, como a nivel universitario para despertar el interés y aumentar el grado de motivación en los alumnos hacia la materia en cuestión (Arévalo Mercado, Andrade, & Reynoso, 2008; Arévalo Mercado & Reynoso, 2010; Arévalo Mercado & Solano Romo, 2013; Brito & Sá-Soares, 2010; Malliarakis et al., 2014; McInerney, 2010).

Con el fin de acercar la programación a estudiantes de nivel básico e intermedio (primaria, secundaria y bachillerato) y despertar su interés y motivación así como aumentar sus habilidades de resolución de problemas, en países desarrollados ha tenido auge el uso de lenguajes de programación educativos como lo son Scratch (Baytak & Land, 2011; García & Peña, 2014; Garneli et al., 2013; Høiseth & Jaccheri, 2011; McInerney, 2010; L.-C. Wang & Chen, 2012), Logo (An & Park, 2011; Serafini, 2011), Alice (Kelleher, 2009; T. Wang et al., 2009) y Kodu (Hui & Tam, 2014).

TESIS TESIS TESIS TESIS TESIS

Algunos países que han optado en realizar estudios utilizando lenguajes de programación educativos en nivel bachillerato y/o con estudiantes en edad de estudiar este nivel educativo han sido Estados Unidos (Burke & Kafai, 2010; Javidi & Sheybani, 2009; Rizvi, Humphries, Major, Jones, & Lauzun, 2011; Rodger et al., 2010; Wolz, Leitner, Malan, & Maloney, 2009), Grecia (Nikou & Economides, 2014), Uruguay (de Kereki, 2008), Irlanda (McInerney, 2010), China (Wang & Zhou, 2011), España (Cearreta-Urbieta & others, 2015) y Ecuador (Quevedo Ordoñez, 2015). En dichos estudios no se encontraron diferencias estadísticamente significativas en lo que respecta al desempeño de los estudiantes, más si en el factor motivacional en el aprendizaje de la programación. A pesar de ello, aun en la literatura existen pocos estudios realizados con estudiantes de nivel bachillerato utilizando un lenguaje de programación educativo que tengan como objetivo aumentar la motivación de los estudiantes y mejorar las experiencias en la programación, así como analizar la influencia de los mismos en el desempeño de los estudiantes y las tasas de deserción en comparación con los cursos tradicionales de programación.

En algunos estudios a nivel bachillerato y universidad se ha hecho uso del lenguaje de programación educativo Scratch (Burke & Kafai, 2010; de Kereki, 2008; McInerney, 2010; Nikou & Economides, 2014; Rizvi et al., 2011; X. Wang & Zhou, 2011; Wolz et al., 2009) con la finalidad de que los estudiantes tengan una agradable transición a un lenguaje de programación orientado a objetos como lo es Java, obteniendo resultados positivos en estudiantes que no habían tenido experiencia alguna en materias de programación, despertando así su interés en ellas (de Kereki, 2008; Malan & Leitner, 2007).

Un estudio relevante respecto al factor motivacional en el aprendizaje de la programación es "*Transition in Student Motivation during a Scratch and an App Inventor course*", realizado en Grecia, basado en el aprendizaje autorregulado y aprendizaje basado en problemas (BPL), en el cual, los autores encontraron efectos positivos en el grado de motivación intrínseca (medida mediante el cuestionario MSLQ) de los estudiantes, en dicho estudio los estudiantes hicieron uso de Scratch y una aplicación denominada App Inventor for Android (Nikou & Economides, 2014).

De acuerdo a la literatura revisada se han llevado a cabo pocos estudios haciendo uso de un LPE a nivel bachillerato. En el estado de Aguascalientes, México, no existe evidencia de investigaciones realizadas en el contexto de esta investigación, he aquí el interés de llevarla a cabo.

1.3 Objetivos de investigación

En este apartado se especifican tanto el objetivo general como los objetivos específicos de la investigación realizada.

Objetivo general

Identificar si la utilización de un lenguaje de programación educativo visual introduce diferencias estadísticamente significativas ya sea en la motivación o en la adquisición de habilidades de resolución de problemas en el ámbito del aprendizaje de la programación básica en estudiantes de educación media superior.

Objetivos específicos

1. Seleccionar un lenguaje de programación educativo, diseñado para un público adolescente.
2. Seleccionar y adaptar un instrumento de medición del grado de motivación aplicable al contexto del aprendizaje de la programación básica en estudiantes de educación media superior.
3. Diseñar un curso utilizando el LPE seleccionado, con el fin de introducir los conceptos básicos de programación básica con el grado de dificultad adecuado de acuerdo al nivel educativo al que va dirigido.
4. Diseñar un instrumento de medición de habilidades de resolución de problemas en el ámbito del aprendizaje de la programación básica para estudiantes de educación media superior.
5. Identificar si existen o no diferencias significativas en la motivación en una población de estudiantes de educación media superior, al utilizar un lenguaje de programación visual educativo.
6. Identificar si existen o no diferencias significativas en la adquisición de habilidades de resolución de problemas, en el ámbito del aprendizaje de la programación básica, después de un tratamiento usando un lenguaje de programación educativo visual.

1.4 Preguntas de investigación

- 1 ¿El uso de un lenguaje de programación educativo puede incrementar significativamente el grado de motivación de un estudiante de bachillerato hacia el aprendizaje de la programación básica?
- 2 ¿El uso de un lenguaje de programación educativo puede incrementar significativamente la adquisición de habilidades de resolución de problemas en estudiantes de bachillerato, en el ámbito de la programación básica?

1.5 Hipótesis

Ho. No existen diferencias, estadísticamente significativas, en la motivación de los estudiantes al hacer uso de un lenguaje de programación educativo en el aprendizaje de la programación básica a nivel bachillerato.

Ha. Existen diferencias, estadísticamente significativas, en la motivación de los estudiantes al hacer uso de un lenguaje de programación educativo en el aprendizaje de la programación básica a nivel bachillerato.

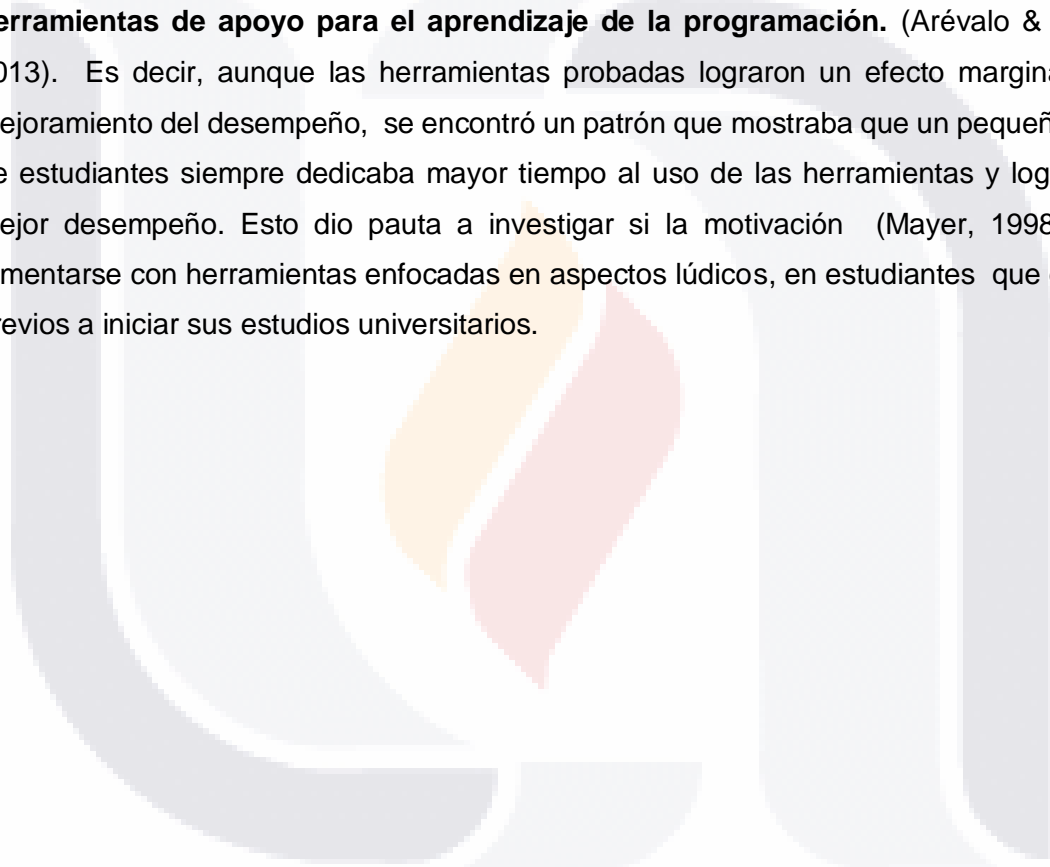
Ho. No existen diferencias, estadísticamente significativas, en la adquisición de habilidades de resolución de problemas al hacer uso de un lenguaje de programación educativo en el aprendizaje de la programación básica a nivel bachillerato.

Ha. Existen diferencias, estadísticamente significativas, en la adquisición de habilidades de resolución de problemas al hacer uso de un lenguaje de programación educativo en el aprendizaje de la programación básica a nivel bachillerato.

1.6 Justificación

En la literatura revisada se reportan altos porcentajes de reprobación en materias de programación a nivel universitario (respecto a nivel bachillerato, en la literatura no se encontró evidencia de estadísticas). De acuerdo a (Dehnadi & Bornat, 2006) entre el 30% y 60% de los estudiantes fallan el primer curso de programación a nivel universitario.

La UAA, es una de la muchas instituciones educativas, a nivel mundial, que tratan de hacer frente al problema del aprendizaje de la programación, tanto a nivel universitario como a nivel bachillerato, implementado diferentes herramientas de software como lo son objetos de aprendizaje y tutores cognitivos (Arévalo Mercado & Solano Romo, 2013) (Arévalo, Andrade, & Reynoso, 2008) (Arévalo & Reynoso, 2010). En estas investigaciones no se encontraron diferencias significativas respecto al desempeño entre los grupos participantes en cada estudio, pero si se reportaron cifras interesantes respecto **a un efecto positivo en el aspecto motivacional y en la correlación entre desempeño y tiempo de uso de herramientas de apoyo para el aprendizaje de la programación.** (Arévalo & Solano, 2013). Es decir, aunque las herramientas probadas lograron un efecto marginal en el mejoramiento del desempeño, se encontró un patrón que mostraba que un pequeño grupo de estudiantes siempre dedicaba mayor tiempo al uso de las herramientas y lograba un mejor desempeño. Esto dio pauta a investigar si la motivación (Mayer, 1998) podía fomentarse con herramientas enfocadas en aspectos lúdicos, en estudiantes que estaban previos a iniciar sus estudios universitarios.



CAPÍTULO II – MARCO TEÓRICO

En este capítulo se especifican las bases de la investigación, como lo es la variable motivación, así como el modelo motivacional de expectativas y valores, en el cual está basado el instrumento de medición utilizado en el experimento (MSLQ), además del modelo de resolución de problemas utilizado para el diseño del tratamiento utilizado y el LPE implementado.

2.1 Motivación

La motivación es la variable clave en esta investigación, por lo tanto a continuación se proporcionan algunas definiciones de esta:

En (Esquivel Alcocer & Roja Cáceres, 2005) se citan un par de definiciones generales acerca de la motivación: para Woolfolk (1996) la motivación es "aquello que energiza y dirige la conducta"; de acuerdo a Good y Brophy (1996) la motivación se define como "un constructo hipotético usado para explicar el inicio, dirección, intensidad y persistencia de la conducta dirigida hacia un objetivo". Mientras que (Hernández & García, 1991) la definen como el impulso, consciente o inconsciente, que surge de una necesidad, que determina las acciones de los organismos en busca de incentivos o metas.

De acuerdo a Trench, estar académicamente motivado significa desear desempeñarse bien en un contexto académico (Castillo, 2005). En este contexto (Polanco Hernández, 2005) menciona definiciones de varios autores respecto a esta variable, desde el punto de vista del alumno así como del docente, las cuales son:

"El grado en que los alumnos se esfuerzan para conseguir metas académicas que perciben como útiles y significativas". Desde el punto de vista del docente, significa "motivar al estudiante a hacer algo, por medio de la promoción y sensibilización" – Santos (1990).

"Motivar supone predisponer al estudiante a participar activamente en los trabajos en el aula. El propósito de la motivación consiste en despertar el interés y dirigir los esfuerzos para alcanzar las metas establecidas" – Campanario (2002).

En la literatura referente a la motivación, se identifican en general dos tipos de motivación: intrínseca y extrínseca (Ballester, 2002; Huertas, 1997; P. R Pintrich, Smith, García, &

McKeachie, 1991; Paul R. Pintrich & De Groot, 1990). La motivación intrínseca, es interna, se refiere a la realización de acciones por satisfacción personal independientemente de si se recibe algún incentivo externo o premio, lo contrario de la motivación extrínseca, la cual es externa, consiste en realizar acciones, no por gusto, sino porque se sabe que al realizarlas se obtendrá una recompensa.

Lepper (1988) citado por (Rinaudo, Chiecher, & Donolo, 2003) menciona que es probable que un estudiante motivado intrínsecamente está más dispuesto a aplicar un esfuerzo mental significativo al realizar una tarea así como a emplear estrategias de aprendizaje más efectivas que un estudiante que solo está extrínsecamente motivado. Por su parte (Howey, 1999) sustenta que existen evidencias empíricas que respaldan la idea de que la motivación intrínseca está relacionada positivamente con el desarrollo de las competencias de un individuo, mientras que (Ballester, 2002) agrega que lo ideal es una combinación de los dos tipos de motivación, potenciando la motivación intrínseca y complementando con la motivación extrínseca.

De acuerdo con Santrock, citado en (Pereira, 2009) son tres las perspectivas fundamentales de la motivación: la conductual, la humanista y la cognitiva. La conductual señala que las recompensas externas y los castigos influyen en la determinación de la motivación de las personas. La humanista hace énfasis en que las personas actúan en sus ambientes haciendo elecciones respecto de qué hacer, interesándose más en su desarrollo personal y realización de su potencial eliminando obstáculos para alcanzar sus metas, por otra parte, la perspectiva cognitiva se enfoca en cómo el humano percibe el mundo que lo rodea, es decir, se enfoca en la forma en que las personas procesan la información haciendo uso de ella en situaciones particulares.

Una de las clasificaciones de las teorías sobre motivación es la que especifica Valdés, citado por (Pereira, 2009), en la cual señala dos grandes grupos: las teorías de contenido y las teorías de proceso. El primer grupo estudia y considera aspectos, entre ellos las necesidades, las aspiraciones y el nivel de satisfacción, los cuales pueden motivar a las personas. Entre estas teorías se encuentran la Teoría de la Jerarquía de Necesidades de Abraham Maslow, así como la Teoría de las Necesidades de McClelland y la Teoría de Existencia, Relación y Crecimiento (ERG, por sus siglas en inglés) de Clayton Alderfer. Las teorías del segundo grupo estudian y hacen énfasis en el proceso de pensamiento que motiva a la persona, entre estas teorías destacan la Teoría de las Expectativas de Vroom,

Teoría de la Equidad de Stacey Adams y la Teoría de la Modificación de la Conducta de Skinner.

Sin duda alguna, la literatura referente al tema de la motivación brinda una amplia variedad de conceptos y teorías vinculadas con este constructo. En esta investigación se tuvo en cuenta sólo aquellos aspectos que forman parte del marco conceptual del instrumento de medición de la motivación elegido, el MSLQ, el cual está basado en una visión cognitiva de la motivación y de las estrategias de aprendizaje (Credé & Phillips, 2011; Roces, Tourón, & González-Torres, 1995), en específico, el modelo conceptual de la motivación es una adaptación del modelo motivacional de expectativas y valores de Eccles (1983) (Paul R. Pintrich & De Groot, 1990). Para dicho modelo conceptual de la motivación Pintrich y DeGroot consideran tres componentes teóricos los cuales son: componente de valor, componente de expectativa y componente afectivo (Credé & Phillips, 2011; Duncan & McKeachie, 2005; Llera, 1995; Roces et al., 1995), dichos componentes son aspectos relevantes para la motivación que deben tomarse en cuenta en el contexto educativo (Pereira, 2009).

2.1.1 Modelo motivacional de expectativas y valores

Los modelos de *expectativas y valores* están basados en el modelo de motivación para el logro de Atkinson (1964), en dicho modelo, la probabilidad de éxito de los estudiantes estaba definida objetivamente en términos de la dificultad de la tarea. Algunos investigadores entre ellos Eccles (1983) señalan que la dificultad de la tarea no determina la expectativa de éxito de los alumnos, pero si la percepción de probabilidad de éxito, dada su percepción de la dificultad de la tarea y su habilidad percibida.

En la Figura 5 se muestran las relaciones generales entre los componentes de expectativa y valor de la tarea y la relación de estos con el logro. La Figura 5 está basada en el modelo de Eccles (1983) con adiciones y refinamientos añadidos por (McKeachie & others, 1987) para integrar varios constructos motivacionales de otros investigadores.

2.1.1.1 Ruta de la expectativa

La ruta de la expectativa, como se muestra en la Figura 5, fluye desde la eficacia de los estudiantes, creencias de control y de resultados hasta percepciones de la tarea y de su percepción de la propia competencia hacia la expectativa del éxito. La combinación de la

expectativa con el valor de la tarea se considera que provoca la participación y el logro de las tareas (McKeachie & others, 1987). En seguida se describen las relaciones entre los componentes de la expectativa descritos en (McKeachie & others, 1987).

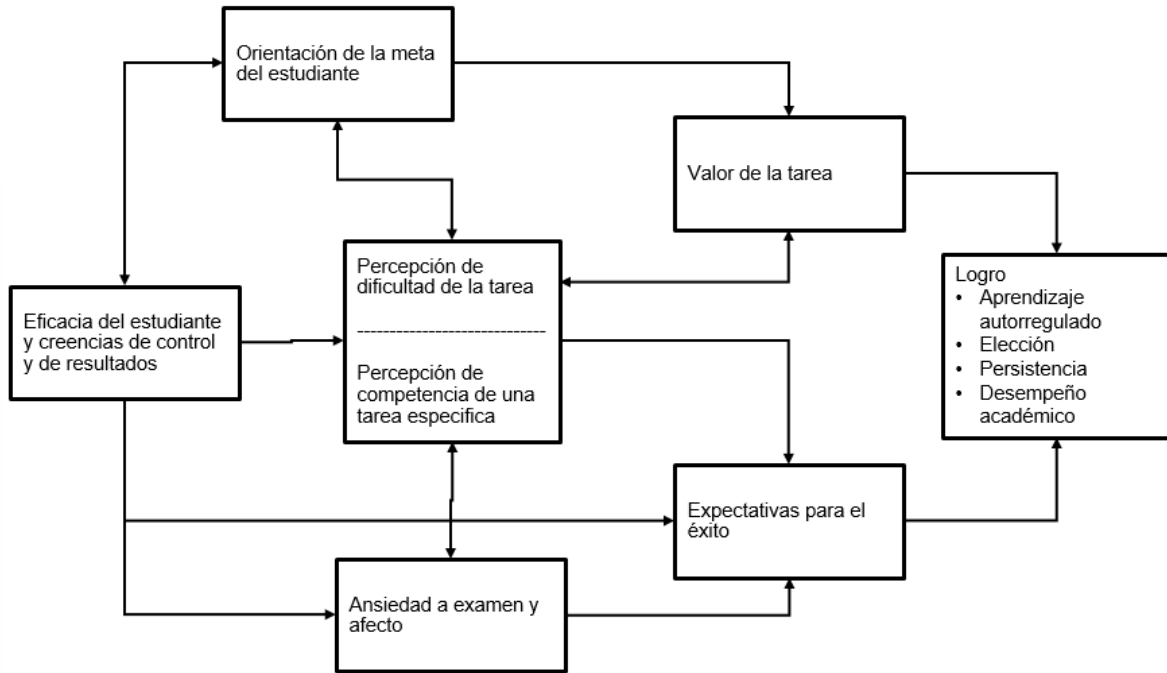


Figura 5. Componentes de la motivación. Fuente: McKeachie & others, 1987

1. Expectativa

El componente de expectativa está más investigado que el componente de valor. La expectativa está definida como la creencia del estudiante acerca de su probabilidad de éxito o fracaso en una tarea en particular.

2. Percepción de la auto-competencia, auto-concepto

Existen dos tipos de percepción del estudiante que juegan un rol directo en la formación de la expectativa, estos son: percepción de auto-competencia y percepción de dificultad de la tarea. La auto-competencia, no es lo mismo que la expectativa de éxito, se define como la percepción de los estudiantes acerca de su habilidad para realizar una tarea determinada. Se diferencia de la expectativa de éxito en que el estudiante puede tener una alta percepción de competencia para una tarea, bajo ciertas condiciones (estrés, actividades aburridas), pero no una alta expectativa de éxito. De acuerdo a Dweck & Elliot (1983) y Eccleses (1983) es la interacción de la percepción de la dificultad de la tarea y la percepción

de competencia la que produce la expectativa del estudiante, Harter (1983) considera que la percepción de competencia es más estable que la expectativa.

3. Percepción de dificultad de la tarea

El modelo propuesto por (McKeachie & others, 1987) asume que las percepciones de los estudiantes acerca de la dificultad de la tarea son mediadores importantes de la conducta de logro en contraste con la dificultad de la tarea. El modelo sugiere que las percepciones de los estudiantes de la dificultad de la tarea puede modular y cambiar la relación de la dificultad de la tarea para la expectativa de éxito de los estudiantes y su posterior conducta de logro, incluida la elección de cursos. La dificultad de la tarea también interactúa con el valor de la tarea para hacer una elección.

4. Ansiedad a examen y afecto

Se considera que la ansiedad a examen tiene dos componentes, componente de preocupación (o cognitiva) y componente emocional, de acuerdo a Tobias (1985), investigaciones demuestran que el componente cognitivo está más asociado con el decremento del desempeño. Como se observa en la Figura 5 los autores asumen que las creencias del estudiante influyen en la ansiedad a examen y que dicha ansiedad se relaciona negativamente con la expectativa de éxito. Una variedad de investigadores citados en Tobias (1985) muestran que los estudiantes que registran un alto grado de ansiedad a exámenes frecuentemente tienen habilidades de estudio menos eficaces. Los estudiantes que no tienen buenas estrategias para presentar exámenes podrían tener bajo desempeño, incluso si están bien preparados.

Aunque la ansiedad a examen incluye afecto, hay otros tipos de emociones que pueden generarse en el contexto de logro, las cuales están relacionadas con la motivación y el rendimiento de los estudiantes. El modelo sugiere que el afecto de los estudiantes está influenciado por sus creencias de eficacia, control y resultados (ver Figura 5).

5. Eficacia del estudiante y creencias de control y de resultados

El quinto componente de la ruta de la expectativa en el modelo propuesto por Pintrich y sus colegas es la creencia acerca de la eficacia de los estudiantes así como las creencias de control y de resultados. Existe una serie de constructos y teorías propuestas sobre el rol de

estas creencias en el rendimiento de los estudiantes. Por ejemplo, los estudios iniciales sobre el *locus de control* reportan que los estudiantes que creían tener el control de su comportamiento y podían influir en el contexto tendían a alcanzar mejores resultados.

La teoría en la que se basa este constructo es la Teoría de las Atribuciones, la cual propone que las atribuciones causales para el éxito y el fracaso influyen en las expectativas futuras. Un gran número de estudios demuestran que los estudiantes que tienden a atribuir el éxito a la capacidad esperan tener buenos resultados en exámenes futuros porque se supone que la capacidad es estable en el tiempo. Por el contrario, estudiantes que atribuyen su éxito a otras causas, por ejemplo, dificultad de una actividad, un examen fácil o al esfuerzo, no tendrán expectativas altas porque la actividad o el esfuerzo cambian con el tiempo, por lo tanto los estudiantes que atribuyen su mal desempeño a la capacidad tienden a tener menor expectativa de éxito en exámenes futuros.

Las tres principales dimensiones en que las atribuciones toman lugar son, locus, estabilidad y capacidad de control. La primera de ellas se refiere a la naturaleza ya sea interna o externa de la causa en relación con el individuo, por ejemplo la capacidad y el esfuerzo son causas internas, mientras que la dificultad de la tarea y la suerte son causas externas. La estabilidad se refiere a la manera en que la causa puede oscilar en el tiempo, por ejemplo la capacidad es una causa interna que se supone es estable en el tiempo, mientras que el esfuerzo se supone que es cambiante en el tiempo. La suerte y la dificultad de la tarea son causas externas que se supone son inestables, de acuerdo a Weiner (1986) la dimensión de la estabilidad está más estrechamente ligada con los cambios en la expectativa de éxito en el contexto de logro. En cuanto a la capacidad de control, se refiere a la capacidad del individuo para controlar la causa, por ejemplo el estado de ánimo, la fatiga y el esfuerzo son causas internas e inestables, sin embargo el esfuerzo esta generalmente bajo control voluntario de la persona, mientras que el estado de ánimo y la fatiga no lo están. De igual forma, la aptitud (causa interna y estable) no se considera estar bajo el control de la persona, mientras que la habilidad (causa interna e inestable) puede estar bajo el control del estudiante.

2.1.1.2 Ruta del valor de la tarea

El flujo general del valor de la tarea en el modelo de la Figura 5 va de orientación de la meta de los estudiantes a valor de la tarea. En seguida se describen los componentes de este constructo.

1. Componente de valor de la tarea

El componente de valor de la tarea fue definido originalmente como el valor que una persona liga al éxito o fracaso de una tarea. Eccles (1983) propone tres componentes importantes de valor de la tarea en la dinámica del logro para objetivos futuros, estos son: el valor de realización de la tarea, el valor intrínseco o el valor de interés y el valor de utilidad de la tarea, dichos componentes se describen a continuación (McKeachie & others, 1987).

Valor de realización de la tarea. Este componente se refiere a la percepción del estudiante acerca de la capacidad de la tarea para proporcionar un reto, cumplir con ciertas necesidades de logro y para confirmar un aspecto sobresaliente del mismo, por ejemplo la competencia. Un alto valor de logro debería conducir a una mayor implicación en la tarea.

Valor intrínseco o interés de una tarea. Se refiere al goce inherente de la persona al realizar una tarea. El interés intrínseco influye en la participación de los estudiantes en una tarea y en su rendimiento futuro. El interés es parcialmente una función de las preferencias de la persona.

Valor de utilidad. Este constructo está determinado por la importancia de la tarea en la facilitación de las metas del estudiante.

2. Vistas motivacionales de orientación a la meta del estudiante.

Las metas de los estudiantes influyen en su valor para ciertas tareas. Existen tres perspectivas generales, las cuales son: modelo de motivación intrínseca, modelo de autoestima y modelo de la formación de metas cognitivas.

La primera perspectiva, es decir el modelo de motivación intrínseca, se relaciona con las creencias de control y percepción de competencias hacia la expectativa de éxito, en cuanto al modelo de autoestima, este asume que una de las fuerzas impulsoras de la dinámica motivacional de los estudiantes es el mantenimiento de la autoestima. En este modelo los

estudiantes no necesariamente deben estar intrínsecamente motivados para el reto o el dominio, sino que están motivados para aumentar sus sentimientos de autovaloración y la autoestima o al menos para protegerla. El contexto del aula es visto como un sistema competitivo con énfasis en la capacidad de los estudiantes. La tercer y última perspectiva propone que existen dos tipos de metas que los estudiantes pueden adoptar en su compromiso en una tarea, estos son: metas de desempeño y metas de aprendizaje. Los estudiantes que adoptan una meta de desempeño se enfocan en su propia capacidad para realizar la tarea, su rendimiento en relación con algún criterio y en su resultado obtenido, además tenderán a ver los errores como ejemplos de fracaso. Por el contrario, estudiantes que adopten una meta de aprendizaje se centraran en el proceso de cómo hacer la tarea, su rendimiento en relación con su desempeño pasado y su implicación en la tarea en contraste con el resultado viendo los errores como útiles.

Las orientaciones motivacionales de los estudiantes son un componente importante de cualquier modelo motivacional del comportamiento del estudiante. La orientación motivacional influye en el nivel de implicación del estudiante en una tarea, el valor que le dan a la tarea y su expectativa y percepción de competencia para dicha tarea (McKeachie & others, 1987).

2.1.2 Medición de la motivación (MSLQ)

El cuestionario de motivación y estrategias de aprendizaje, MSLQ por sus siglas en inglés, es un instrumento de auto-reporte diseñado para evaluar las orientaciones motivacionales de los estudiantes y el uso de diferentes estrategias de aprendizaje, tanto cognitivas como metacognitivas, en un curso específico (Paul R. Pintrich & others, 1991). Este cuestionario es el que se utiliza para la medición de la variable de interés en esta investigación.

El MSLQ fue desarrollado entre 1982 y 1986 por Pintrich y sus colaboradores en la Universidad de Michigan. Está conformado por dos escalas generales, la escala de motivación y la escala de estrategias de aprendizaje, con un total de 81 ítems de acuerdo a la última versión del MSLQ desarrollada en 1991. Ambas escalas son modulares, por lo cual pueden ser administradas de forma independiente según los requerimientos de la investigación.

La escala de motivación consta de 31 ítems que evalúan las metas de los estudiantes y sus creencias de valor y de su habilidad para tener éxito en su aprendizaje, además de su ansiedad ante los exámenes en un curso específico. La escala de estrategias de aprendizaje, al igual que la escala de motivación, incluye 31 ítems acerca del uso de diferentes estrategias tanto cognitivas como metacognitivas de los estudiantes. Además, la escala de estrategias de aprendizaje incluye 19 ítems relacionados con la utilización efectiva, por parte de los estudiantes, de los recursos con los que cuentan. El cuestionario está estructurado en una escala likert de 7 puntos, donde 1 significa “muy en desacuerdo” y 7 “muy de acuerdo”.

Esta investigación solo se basó en la escala de motivación, por lo tanto solo se tomó en cuenta la escala, del cuestionario, que mide dicha variable, en la Tabla 1 se muestran las dimensiones de la escala utilizada, así como las sub-escalas que la conforman y el número de preguntas que contiene cada una de ellas.

Tabla 1. Escala de motivación del MSLQ. Fuente: Pintrich et al., 1991

Motivación		
Dimensión	Sub-escala	# Preguntas
Componente de valor	Orientación Intrínseca	4
	Orientación Extrínseca	4
	Valor de la tarea	6
Componente de expectativas	Creencia de control	4
	Autoeficacia	8
Componente afectivo	Ansiedad a examen	5

Como se mencionó anteriormente la escala de motivación del MSLQ está basada en el modelo de expectativas y valores, en la Tabla 1 se observa que la escala de motivación está compuesta por 3 componentes los cuales son el componente de valor, el componente de expectativas y el componente afectivo, dichos componentes se describen en las siguiente subsecciones.

2.1.2.1 Componente de Valor

Para los autores del MSLQ el concepto de valoración es definido como un proceso de tipo atribución mediante el cual se desea o rechaza una meta, actividad u objeto (McKeachie & others, 1987).

1. Orientación de meta intrínseca

La orientación de meta se refiere a la percepción que tiene un estudiante de las razones del por qué está participando en una tarea de aprendizaje. En el MSLQ la orientación de meta se refiere a las metas u orientación general de los estudiantes para un curso en su totalidad.

La meta de orientación intrínseca se refiere al grado en que un estudiante se percibe a sí mismo para estar participando en una tarea por razones de índole personal, tales como desafío, curiosidad o dominio de la actividad. Al tener una meta de orientación intrínseca el aprendizaje se convierte en un fin no en un medio. A esta sub-escala pertenecen las preguntas 1, 16, 22 y 24.

2. Orientación de meta extrínseca

La orientación de meta extrínseca complementa la orientación de meta intrínseca, y se refiere al grado en el cual los estudiantes perciben que están realizando una tarea por razones externas (obtener una recompensa) tales como calificaciones, premios, el rendimiento, la evaluación por otros, y la competencia. Un alto puntaje en esta sub-escala indica que el estudiante percibe el aprendizaje como un medio para conseguir un fin. Esta sub-escala está conformado por las preguntas 7, 11, 13 y 30.

3. Valor de la tarea

En el MSLQ, el valor de la tarea se refiere a la percepción de los estudiantes acerca de los materiales del curso en términos de interés, importancia, y utilidad, es decir el grado en que el estudiante tiene una creencia de que el material de la clase es interesante y vale la pena aprender, a diferencia de la orientación de metas la cual se refiere a las razones por las cuales el estudiante está participando en una tarea. A esta sub-escala corresponden las preguntas 4, 10, 17, 23, 26 y 27.

2.1.2.2 Componente de Expectativas

La expectativa es el grado en que el alumno confía que posee el control sobre su aprendizaje, y el grado en que considera que gracias a su eficacia personal puede aprender y obtener buenos resultados.

1. Creencias de control sobre el aprendizaje

El control del aprendizaje se refiere a las creencias de los estudiantes de que sus esfuerzos por aprender producirán resultados positivos. Se refiere a la creencia de que los resultados dependen del esfuerzo propio de uno, en contraste con los factores externos tales como el maestro. Si los estudiantes creen que sus esfuerzos para estudiar hacen una diferencia en su aprendizaje, serán más propensos a estudiar de manera más estratégica y eficaz. Es decir, si el estudiante siente que puede controlar su rendimiento académico, es más probable que ponga de manifiesto lo que se necesita de manera estratégica para efectuar los cambios deseados. Las preguntas que conforman esta sub-escalas son 2, 9, 18 y 25.

2. Autoeficacia para el aprendizaje y el desempeño

Esta sub-escala evalúa dos aspectos de expectativa: expectativa para el éxito y la autoeficacia. La expectativa de éxito hace referencia a las expectativas de rendimiento, y se refiere específicamente a la ejecución de tareas. La autoeficacia es una auto-evaluación de la capacidad de un estudiante para dominar una tarea. La autoeficacia incluye juicios sobre la capacidad para realizar una tarea, así como la confianza de un estudiante de sus habilidades para llevar a cabo esa tarea, esta sub-escala está compuesta por las preguntas 5, 6, 12, 15, 20, 21, 29 y 31.

2.1.2.3 Componente afectivo

El MSLQ solo incluye un afecto, el cual es la ansiedad. Se considera que la presencia de niveles altos de ansiedad afecta de forma negativa el rendimiento académico de los alumnos.

1. Ansiedad a examen

En el MSLQ la ansiedad a examen es el grado en que los estudiantes experimentan ansiedad y miedo al presentar exámenes, se considera que la ansiedad es una de las fuentes que mayor influencia negativa tiene en el rendimiento, sin embargo se puede instruir al estudiante para que por medio de estrategias de control y de aprendizaje logre reducir la ansiedad. Las preguntas correspondientes de esta sub-escala son 3, 8, 14, 19 y 28.

2.2 Resolución de Problemas

La *resolución de problemas* es el proceso de construcción y aplicación de representaciones mentales -de los problemas- para encontrar soluciones a los problemas en casi todos los contextos (Jonassen & Hung, 2012).

Modelos tradicionales de resolución de problemas sugieren que todos los problemas pueden ser resueltos si (1) se identifica el problema, (2) se generan soluciones alternativas, (3) se evalúan esas soluciones, (4) se implementa la solución seleccionada y (5) se evalúa la efectividad de la solución (Jonassen, 2012).

Una de las metodologías más utilizadas en la resolución de problemas, y en la que se basa el tratamiento diseñado, es la formulada por Polya, citado en (López García & others, 2014), la cual consta de cuatro etapas que coinciden en la elaboración de programas de computadoras, estas son: entender el problema, elaborar un plan, ejecutar el plan (resolver) y revisar, dichas etapas se muestran en la Figura 6.

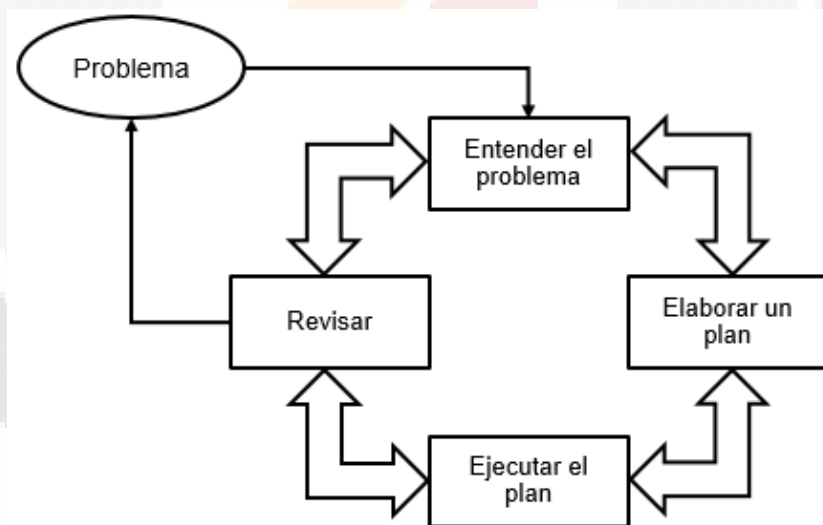


Figura 6. Modelo de resolución de problemas de Polya. Fuente: “Synthesis of Research on Problem Solving”, s/f

Entender el problema. En esta etapa el estudiante debe entender el problema, pero no solo debe entenderlo si no también debe desear su solución. El estudiante debe ser capaz de señalar las partes principales del problema, lo que se desconoce de él, sus datos y la condición así como si es posible satisfacer dicha condición para solucionarlo. Se debe dar

respuesta a las preguntas ¿Qué se desconoce? y ¿Qué se quiere demostrar? (Polya, 1945; Shirali, 2014).

Elaborar un plan. De acuerdo a Polya, la mejor forma de llegar a una solución, es concebir un plan, por lo tanto, el estudiante debe trazar un plan para llegar a la solución de un problema, pero no es tarea sencilla, implica que el estudiante debe conocer qué cálculos, cómputos o construcciones se deben llevar a cabo con el fin de obtener lo que se desconoce, es decir, la solución.

Ejecutar el plan. Consiste en que el estudiante lleve a cabo los pasos especificados en la etapa anterior, es decir, ejecute el plan elaborado que llevará a la solución del problema. Es una de las etapas menos difíciles en el proceso de resolución de problemas. Se debe asegurar que los detalles del problema encajen en el plan trazado y se deben examinar uno tras otro “pacientemente” hasta que todo esté perfectamente claro evitando así que el plan elaborado contenga errores sin identificar (Polya, 1945).

Revisar. Mediante la revisión de la solución completa, reconsiderando y reexaminando el resultado y el plan que llevó a él, los estudiantes pueden consolidar su conocimiento y desarrollar su habilidad para resolver problemas. Además de contestar a la pregunta ¿Cómo se puede mejorar el plan? (Polya, 1945; Shirali, 2014).

2.3 Lenguajes de programación educativos (LPE)

De acuerdo a (Brusilovsky, 1994) un lenguaje de programación educativo (LPE) es un conjunto de herramientas que apoyan al estudiante en el proceso de aprendizaje de la programación introductoria.

La selección de un lenguaje de programación, teniendo en cuenta el nivel cognitivo de los estudiantes y el propósito de aprender programación, es considerada como un factor clave para aumentar la eficacia de la enseñanza de programación.

De acuerdo a (Yoon, Kim, & Lee, 2016) un LPE facilita el proceso de programación, mediante ambientes de programación visuales y una sintaxis de programación intuitiva, más que un lenguaje de programación de propósito general.

Un LPE permite a un programador novato programar de forma rápida y fácil mediante la reducción de la dificultad de la programación en términos de enseñanza de la programación

(Yoon et al., 2016). Al hacer uso de un LPE, los estudiantes pueden poner en práctica rápidamente los conceptos de programación aprendidos ya que un LPE consiste de instrucciones intuitivas y de una estructura de sintaxis fácil (Yoo et al., 2006).

Scratch, Alice, Dolittle, y Squeak son algunos lenguajes de programación que han sido sugeridos como LPE. Scratch, es un lenguaje de programación visual que soporta instrucciones intuitivas basadas en bloques y herramientas multimedia, este LPE ha sido adoptado por el plan nacional de estudios de informática para el aprendizaje de programación tanto en Corea así como en la India e Israel (Yoon et al., 2016).

En la siguiente subsección se habla del LPE Scratch, el cual fue seleccionado para llevar a cabo la presente investigación.

2.3.1 Scratch

De acuerdo al MIT, Scratch es un lenguaje de programación interactivo que hace que sea fácil crear historias interactivas, animaciones, juegos, música y arte que se pueden compartir en la web (“Scratch - Imagine, Program, Share”, s/f). El objetivo principal de Scratch es introducir la programación a los que no tienen experiencia previa en ella (John Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010; Resnick et al., 2009).

Scratch está escrito en Squeak, la cual es una implementación de código abierto del lenguaje Smalltalk-80.

El diseño original de Scratch fue motivado por las necesidades e intereses de jóvenes en los centros de computación después de clases, como el Intel Computer Clubhouse. Scratch inicialmente se utilizó en entornos de aprendizaje informales como centros comunitarios, clubes después de la escuela, bibliotecas y casas, pero cada vez más es utilizado en escuelas.

Scratch se utiliza para la creación de proyectos que contienen media y scripts. Las imágenes y sonidos pueden ser importados o creados con un sistema de herramientas incorporado de dibujo y grabación de sonido, la programación se realiza mediante la unión de bloques de comandos para controlar objetos gráficos, en dos dimensiones, llamados scripts (Resnick et al., 2009). El enfoque de bloques de construcción surgió de la investigación previa sobre LogoBlocks, utilizado como base para el lenguaje de

programación usado en MindStorms de LEGO y Etoys de Squeak, que han sido utilizados con éxito en los salones de clase (J. Maloney et al., 2004; John Maloney et al., 2010)

2.3.1.1 Principales características

Entre las principales características de Scratch mencionadas en (J. Maloney et al., 2004) se encuentran:

Programación en bloques de construcción. Scratch se basa en una metáfora de bloques de construcción, en el que los alumnos construyen secuencias de comandos mediante la unión de bloques gráficos parecidos a las piezas de un rompecabezas. Tanto los comandos así como los tipos de datos están representados por bloques de diferentes formas, con piezas de montaje que encajan unas con otras (J. Maloney et al., 2004). Este enfoque elimina los errores de sintaxis (que han demostrado ser un obstáculo importante para el aprendizaje de lenguajes de programación basados en texto), lo que permite a los jóvenes centrarse en los problemas que se quieren resolver, no en la mecánica de la programación (J. Maloney et al., 2004; Resnick et al., 2009). El usuario de Scratch arrastra bloques de comandos de una paleta para crear "pilas" (procedimientos) que rigen el comportamiento de sus objetos. El multiprocesamiento se integra sin problemas en Scratch ya que diferentes pilas de bloques pueden ejecutarse en paralelo (J. Maloney et al., 2004). La Figura 7 corresponde a la interfaz de usuario de Scratch.

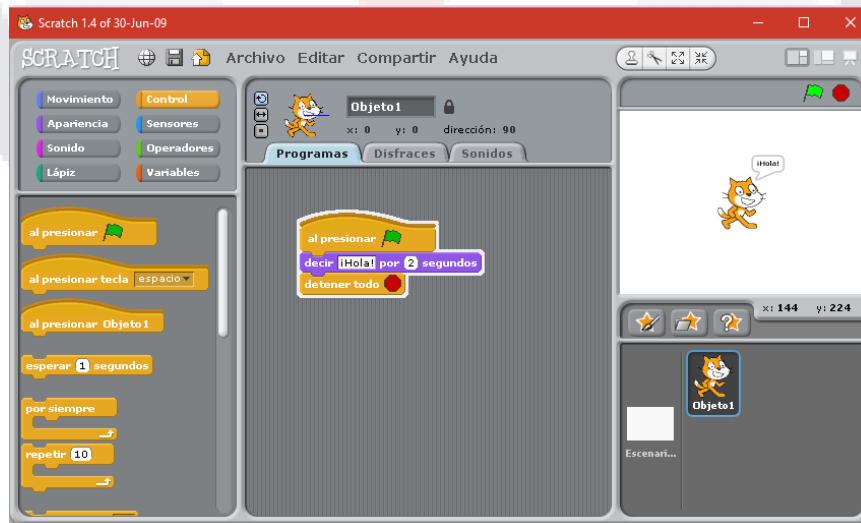


Figura 7. Interfaz de Scratch: Fuente: Scratch

Manipulación de media. Scratch permite la manipulación de contenido media como lo son imágenes, animaciones, películas y sonido, lo cual es de mayor interés para los jóvenes.

Distribución de contenido. La arquitectura de objetos de Scratch apoya lo que los autores llaman "distribución profunda". La juventud será capaz de exportar desde personajes animados individuales hasta proyectos completos, e intercambiarlos con amigos que ejecuten Scratch en gran variedad de dispositivos como son computadoras de escritorio y portátiles, tabletas e incluso teléfonos móviles.

Integración con el mundo físico. Se pueden utilizar entradas de sensores físicos tales como interruptores, reguladores, sensores de distancia, detectores de movimiento, sensores de sonido, para controlar el comportamiento de las creaciones en Scratch.

Soporte para múltiples idiomas. Para apoyar la colaboración y el intercambio en este contexto, es esencial para Scratch ser un entorno multi-lenguaje y multi-cultural. Scratch le permitirá al usuario cambiar entre idiomas de forma dinámica, fomentando la colaboración entre culturas, tanto dentro de una misma ubicación geográfica así como de cualquier parte del mundo (J. Maloney et al., 2004).

2.4 Justificación del uso de Scratch

Scratch es un entorno que trata de ofrecer características de innovación como apoyo a la enseñanza de la programación de computadoras (Begosso & Rodrigues da Silva, 2013), debido a la característica del LPE Scratch, este es adecuado para el aprendizaje de programación básica (Malan & Leitner, 2007; X. Wang & Zhou, 2011) puesto que aumenta la participación de los estudiantes (Resnick et al., 2009). Malan y Leitner proponen el uso de Scratch como primer lenguaje en los cursos introductorios de programación como un puente hacia el uso de Java, además los autores argumentan que Scratch permite a los estudiantes no solo dominar la programación antes de la sintaxis sino también les permite enfocarse en la lógica de los problemas antes de enfocarse en una sintaxis y familiarizarse con los fundamentos de la programación (Malan & Leitner, 2007).

2.5 Contexto del Cuasi-Experimento

El presente cuasi-experimento se realizó en el bachillerato perteneciente a la Universidad Autónoma de Aguascalientes (BachUAA), en los dos planteles con los que cuenta la institución antes mencionada, la cual se localiza en la ciudad capital del estado de Aguascalientes, ubicada en la región centro de México, colindante con los estados de Zacatecas y Jalisco, y que de acuerdo al censo del año 2012 cuenta con una población de 968,119 habitantes (González Villarreal, 2014).



CAPÍTULO III – METODOLOGÍA

En este capítulo se especifica el tipo de investigación y se describe el procedimiento que se llevó a cabo para la realización del experimento, así como el procedimiento para llegar al cumplimiento de los objetivos, específicos y el objetivo general de la investigación, listados en el Capítulo I.

3.1 Tipo y diseño de investigación

El diseño de la investigación fue *cuasi-experimental*. En la Figura 8 se ilustra el diseño de la investigación. G1 (Grupo experimental), O1 (Pre prueba de motivación y pre prueba de programación básica del G1), Tratamiento experimental (Curso de programación básica mediante un lenguaje de programación educativo), O2 (Post prueba de motivación y post prueba de programación básica del G1).

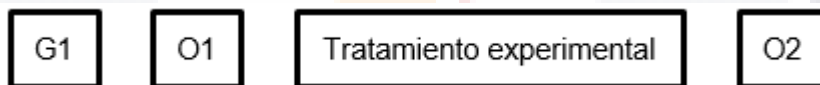


Figura 8. Diseño de la investigación. Fuente: Elaboración propia

3.2 Diseño del Cuasi-Experimento

Como se mencionó anteriormente, se consideró un diseño cuasi-experimental de pre-prueba, post-prueba, con un grupo homogéneo experimental perteneciente al BachUAA. De acuerdo a (Harris et al., 2006), este tipo de diseño es de uso común, consiste en tomar solo una medida antes (pre-prueba) de que se lleve a cabo un tratamiento, y tomar una medida después (post-test) de dicho tratamiento, al mismo grupo experimental. En este caso la pre-prueba sirve como el periodo de “control”, proporcionando información acerca de los resultados que se habrían tenido de no haber aplicado un tratamiento experimental. Se optó por este tipo de diseño debido a la dificultad de crear grupos aleatorios, dada la poca disponibilidad de los participantes fuera de horarios escolares.

La realización de este cuasi-experimento se llevó a cabo en dos fases:

- La aplicación de la prueba piloto en noviembre de 2015. El tamaño de la muestra fue $n=111$. El tratamiento experimental fue aplicado dentro de la materia *Exploración de Software y Sistemas Multimedia*, la cual tiene como objetivo desarrollar

habilidades de edición de imagen, edición de video y animación, para este último se utilizó el tratamiento.

- La aplicación de la réplica en marzo de 2016, para este caso el tamaño de la muestra fue $n=67$. El tratamiento experimental se impartió dentro de la materia *Lógica Computacional y Programación*, cuyo objetivo es introducir la programación básica mediante la resolución de problemas.

Para la medición de la motivación se utilizó el instrumento MSLQ (descrito en el Capítulo II) y se diseñó un curso de programación básica en Scratch (ver **ANEXO A**, dicho anexo solo es una parte del tratamiento), de acuerdo a especificaciones recomendadas por el personal académico de la institución en la que se realizó la investigación, además del diseño de las pre y post pruebas para la medición de conocimientos de programación básica.

El grupo experimental recibió un tratamiento, en este caso, el curso de programación básica, durante una semana, utilizando el lenguaje de programación educativo Scratch. En la Figura 9 se muestra gráficamente el procedimiento llevado a cabo durante la prueba piloto.

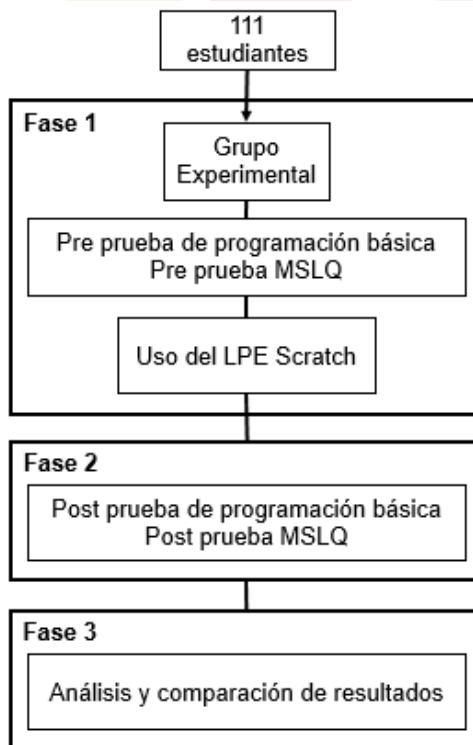


Figura 9. Metodología durante la prueba piloto. Fuente: Elaboración propia

3.3 Población objetivo

El cuasi-experimento se aplicó a alumnos de quinto semestre del bachillerato físico – matemático, en específico a alumnos que cursaban la materia de *Exploración de Software y Sistemas Multimedia* en los dos planteles del BachUAA, un grupo en el plantel central y dos grupos en el plantel oriente. Se eligió a estos grupos, como grupo experimental, debido a que es durante el quinto semestre que los alumnos cursan la primera materia referente a la programación de computadoras, solo como una pequeña introducción sobre este contexto, sin entrar en detalle con la implementación de un lenguaje de programación específico, esto con la intención de que los alumnos desarrollen la lógica de programación.

3.4 Prueba piloto noviembre 2015

La prueba piloto del cuasi-experimento se llevó a cabo a finales del semestre académico agosto-diciembre 2015 a fin de introducir la programación básica a los alumnos de quinto semestre del BachUAA, con el propósito de no interferir en el programa académico de la institución. En dicha prueba piloto participaron 111 estudiantes.

El objetivo de esta prueba piloto fue calibrar los diferentes instrumentos de medición utilizados en la investigación, los cuales incluyen la validación de la adaptación de la pre y post prueba de motivación, así como la validez y calibración del tratamiento (curso de programación básica en Scratch) y de las pre y post pruebas de programación básica.

Tanto las pre pruebas como las post pruebas inician con la solicitud de información demográfica del estudiante, como lo es el ID de estudiante, edad, experiencia en programación, entre otras, esto con el fin de llevar el control de los estudiantes participantes en el cuasi-experimento. Dicha solicitud de información se muestra en el **ANEXO B**.

Las pre y post pruebas fueron implementadas en *LimeSurvey*, la cual es una aplicación open source para la realización de encuestas en línea, con el fin de facilitar la recolección de las respuestas de las pre pruebas necesarias para el cuasi-experimento.

3.4.1 Pre prueba y post prueba de motivación

En lo que respecta al instrumento de medición de la motivación de los estudiantes, el MSLQ, se tradujo y se adaptó al contexto del aprendizaje de la programación básica tomando como

referencia el manual de uso del MSLQ (Paul R. Pintrich & others, 1991) y una traducción del mismo ya validada para la población de Chile (Sánchez Abarca & Burgos Castillo, 2012) adaptándolo al contexto mexicano.

Dicha traducción del cuestionario de motivación se adaptó individualmente tanto para la pre prueba así como la post prueba, de acuerdo al contexto, es decir antes y después de la aplicación del tratamiento experimental. La pre prueba y post prueba empleadas en la prueba piloto se muestran en el **ANEXO C** y **ANEXO D** respectivamente.

3.4.2 Pre prueba y post prueba de programación básica

La pre prueba y post prueba respecto al aprendizaje de la programación básica fueron diseñadas conforme a problemas básicos con la complejidad adecuada al nivel educativo en la que fueron aplicadas y a las habilidades de los estudiantes, cuyas soluciones requerían la correcta aplicación de diversos fundamentos algorítmicos como lo es la secuencia de instrucciones, uso de variables, operadores, condicionales y ciclos.

Al diseñar la pre y post prueba de habilidades algorítmicas, se identificó el reto de cómo evaluar a los estudiantes, ya que para la mayoría de ellos era la primera experiencia en materias de programación, por lo que se optó por diseñar las pre y post pruebas con opción múltiple. Es decir, el estudiante no tuvo que programar directamente. Esto debido a que en BachUAA es hasta quinto semestre y justo en la materia de *Exploración de Software y Sistemas Multimedia* que los estudiantes tienen su primera experiencia de programación básica, por lo que no era viable hacer que ellos programaran. De tal suerte, el instrumento se diseñó de la misma manera para que ambas pruebas no variasen en formato ni en complejidad.

Las pruebas consistieron en cuatro ejercicios, en los cuales al estudiante se le daba el problema a resolver y se le mostraban cuatro posibles respuestas en forma de algoritmo y código en Scratch para la pre prueba y post prueba respectivamente (Ver Figura 10). Los estudiantes debían leer las especificaciones del problema y los requerimientos para llegar a la solución, además de analizar cada una de las opciones de respuesta y elegir la que fuera la opción correcta que solucionara cada uno de los problemas dados. Para la evaluación de las pruebas, solo se revisaron las respuestas de cada estudiante verificando cuántos ejercicios contestó correctamente y así obtener una calificación.

La pre prueba y post prueba utilizadas en la prueba piloto consistieron de cuatro ejercicios con valor de 25 cada uno de ellos, dichas pruebas se muestran en el **ANEXO E** y **ANEXO F** respectivamente.

Pre-prueba

Ejercicio 4: Tabla de multiplicar.

Algoritmo en el que:

- Se pida un número y de acuerdo a este.
- Se imprima su tabla de multiplicar.

Algoritmo 1:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Preguntar numero
6. Introducir respuesta
7. Fijar numero a respuesta
8. Fijar contador a 1
9. Repetir por 10 veces
 - 9.1. Fijar resultado a (numero * contador)
 - 9.2. Decir numero X contador = resultado
 - 9.3. Incrementar valor de contador en 1 (contador + 1)
10. Fin del ciclo repetir
11. Fin del programa

Algoritmo 2:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Preguntar numero
6. Introducir respuesta
7. Fijar numero a respuesta
8. Fijar contador a 1
9. Repetir por 10 veces
 - 9.1. Fijar resultado a (numero * numero)
 - 9.2. Decir numero X numero = resultado
 - 9.3. Incrementar valor de numero en 1 (numero + 1)
10. Fin del ciclo repetir
11. Fin del programa

Algoritmo 3:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Introducir respuesta
6. Preguntar numero
7. Fijar numero a respuesta
8. Fijar contador a 10
9. Repetir por contador veces
 - 9.1. Fijar resultado a (numero * contador)
 - 9.2. Decir numero X contador = resultado
 - 9.3. Incrementar valor de contador en 1 (contador + 1)
10. Fin del ciclo repetir
11. Fin del programa

Algoritmo 4:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Introducir respuesta
6. Preguntar numero
7. Fijar numero a respuesta
8. Fijar contador a 10
9. Repetir por contador veces
 - 9.1. Fijar resultado a (numero * numero)
 - 9.2. Decir numero X numero = resultado
 - 9.3. Incrementar valor de numero en 1 (numero + 1)
10. Fin del ciclo repetir
11. Fin del programa

Post-prueba

Ejercicio 2: Promedio de 3 calificaciones.

Programa en el que un personaje:

- Pida 3 calificaciones, las cuales deben ser introducidas mediante el teclado.
- Calcule el promedio de las 3 calificaciones
- Muestre el promedio, así como si el promedio es aprobatorio o no (el promedio mínimo aprobatorio es 7).

Programa 1:

Programa 2:

Programa 3:

Programa 4:

Figura 10. Ejemplo pre-prueba y post-prueba de programación básica durante la prueba piloto. Fuente: Elaboración propia

3.4.3 Selección de participantes

La selección de los participantes fue no-probabilístico ya que la población de interés para la realización de esta investigación es a nivel bachillerato en específico el BachUAA, con el fin de reducir el riesgo de mortandad experimental y así obtener una mayor validez y fiabilidad en la réplica del cuasi-experimento al realizarlo de acuerdo al contenido y tiempo especificados por la institución.

De tal suerte, se seleccionaron tres grupos de quinto semestre, de los planteles central y oriente, con orientación a carreras informáticas en específico orientadas a programación. El grupo C del platel central, con un total de 33 alumnos y los grupos A y B del plantel oriente con un total de 42 y 36 estudiantes respectivamente, resultando en un grupo experimental de 111 estudiantes.

3.4.4 Aplicación de las pre pruebas de motivación y de programación básica

Una vez acordadas las fechas para la aplicación de las dos pre pruebas (motivación y programación básica), se pidió apoyo a los profesores que imparten la materia de *Exploración de software y Sistemas Multimedia*, con el fin de que se diera la oportunidad de utilizar un espacio en dicha clase para que ellos mismos solicitaran a los estudiantes la realización de ambas pre pruebas.

En primer lugar, los estudiantes contestaron la pre prueba de programación básica, la cual tuvo lugar los días 29 de octubre y 3 de noviembre de 2015, esto dependiendo del día disponible de cada grupo para aplicarla. Posteriormente los estudiantes llenaron la pre prueba de motivación, la cual se llevó a cabo los días 9 y 11 de noviembre de 2015.

3.4.5 Tratamiento experimental

El tratamiento consistió en un curso de programación básica implementando el LPE Scratch, el cual fue programado en cinco sesiones de una hora, correspondientes a una semana, del 9 al 13 de noviembre de 2015, a petición de la institución. Las sesiones se llevaron a cabo durante la hora de clase de la materia de *Exploración de software y Sistemas Multimedia*, puesto que es parte del programa curricular y así no se interfirió en él.

El curso fue impartido entre los días 9 y 17 de noviembre de 2015 en el plantel oriente y los días 10 y 18 de noviembre de 2015 en el plantel central, esto debido al calendario escolar y contratiempos de los maestros y grupos participantes y a fallas eléctricas, las cuales impidieron el término del curso en la semana acordada.

Las sesiones fueron impartidas por los maestros del BachUAA que imparten la materia de la que formó parte el curso diseñado, esto con la finalidad de evitar sesgos al participar el investigador en la impartición del curso y para reducir la inquietud y/o desinterés de los estudiantes participantes. Dichas sesiones tuvieron lugar en los laboratorios de cómputo de ambos planteles, en las que cada estudiante tuvo acceso a un equipo de cómputo con el LPE Scratch instalado. La presentación del material del curso se realizó mediante la utilización de un proyector.

El curso consistió en una introducción a la programación básica, haciendo hincapié en la secuencia de instrucciones, uso de variables, operadores, así como el uso de condicionales y ciclos. En la Tabla 2 se especifica la estructura del curso.

Tabla 2. Estructura del curso diseñado. Fuente: Elaboración propia

Sesión	Contenido	Objetivo
1	Resolución de problemas Pasos en la resolución de problemas Introducción a los algoritmos ¿Qué es Scratch? Interfaz de Scratch Conociendo los conjuntos de instrucciones Primeros pasos en Scratch	Que el estudiante sea capaz de realizar algoritmos para la resolución de problemas en el contexto de la programación, conozca la interfaz de Scratch así como sus conjuntos de instrucciones o bloques para lograr ponerlos en práctica en la realización de programas básicos.
2	Variables Números aleatorios	Introducir el concepto de variables en Scratch. Que el estudiante sea capaz de crear variables e implementarlas cuando sea necesario el uso de éstas así como la generación e implementación de números aleatorios.
3	Operadores aritméticos Operadores de comparación	Que el estudiante conozca los operadores tanto aritméticos como de comparación y sea capaz de implementarlos para realizar cálculos matemáticos y comparación de valores.
4	Condicionales: Condicionales simple (Si) Condicionales doble (Si ... Si no)	Que el estudiante conozca los condicionales básicos y sea capaz de implementarlos adecuadamente en la resolución de problemas.
5	Ciclos: Por siempre Repetir por...	Que el estudiante conozca los ciclos básicos y sea capaz de implementarlos adecuadamente en la resolución de problemas.

El curso se diseñó haciendo énfasis en la creación de algoritmos, en base a la metodología de resolución de problemas propuesta por Polya (descrita en el Capítulo II) para la resolución de los ejercicios o problemas dados.

3.4.6 Aplicación de las post pruebas de motivación y de programación básica

La post prueba de programación básica se aplicó los días 18, 19 y 26 de noviembre de 2015, de acuerdo a los tiempos de cada grupo participante y permitir que alumnos

rezagados la aplicaran, mientras que la post prueba de motivación se aplicó los días 19 y 26 de noviembre de 2015.

3.5 Replica marzo 2016

La réplica calibrada se llevó a cabo en el mes de marzo de 2016, en ella participó un grupo con características similares al que participó en la prueba piloto, dicho grupo fue conformado por los grupos A y F del plantel oriente del BachUAA, con 25 y 42 estudiantes respectivamente, dando un total de 67 estudiantes participantes. Cabe mencionar que los profesores que impartieron el curso fueron diferentes a los que participaron en la prueba piloto.

Al igual que en la prueba piloto, las pre pruebas así como las post pruebas inician con la solicitud de información demográfica del estudiante, como lo es el ID de estudiante, edad, experiencia en programación, entre otras, esto con el fin de llevar el control de los estudiantes participantes en el cuasi-experimento. Dicha solicitud de información se muestra en el **ANEXO B**. La recolección de los datos de las pre y post pruebas se llevaron a cabo haciendo uso de *LimeSurvey*, herramienta utilizada durante la prueba piloto.

3.5.1 Calibración del experimento

Para la realización de la réplica solo fue necesario calibrar el instrumento de medición de habilidades de programación básica, en específico la post prueba.

La calibración necesaria que se realizó a la post prueba de programación fue incluir un ejemplo de ejecución de la solución de los cuatro problemas que conforman la prueba, puesto que este detalle se incluyó en la pre prueba y fue omitido en la post prueba durante la prueba piloto, viendo la necesidad de realizar esa calibración con el fin de que no existiera una diferencia considerable entre las pre y post prueba de programación básica. Dicho instrumento calibrado se muestra en el **ANEXO G**.

3.5.2 Aplicación de las pre pruebas de motivación y de programación básica

Nuevamente, como en la prueba piloto, se pidió a los profesores su apoyo para utilizar un espacio de su clase para que solicitaran a los estudiantes participantes la realización de ambas pre pruebas.

El orden de la realización de ambas pre pruebas fue el mismo que se siguió durante la prueba piloto, la pre prueba de programación básica se llevó a cabo los días 2 y 7 de marzo de 2016. Posteriormente el día 7 de marzo de 2016 los estudiantes llenaron la pre prueba de motivación.

3.5.3 Tratamiento experimental

Las sesiones del experimento se llevaron a cabo siguiendo el mismo formato y contenido llevado a cabo en la prueba piloto. Fueron cinco sesiones de una hora cada una de ellas, dichas sesiones se impartieron del 7 al 11 de marzo de 2016.

3.5.4 Aplicación de las post pruebas de motivación y de programación básica

Las post pruebas tanto de programación básica como de motivación se llevaron a cabo los días 28 y 29 de marzo de 2016 respectivamente dentro del horario de clases de los estudiantes participantes.

Como es notorio, la aplicación de las post pruebas se realizó fuera de la fecha en la que se pretendía, esto debido a que no fue posible un control total del experimento, ya que en los días en que este se llevó a cabo, los estudiantes tuvieron que atender actividades extracurriculares y se les pidió que realizaran las post pruebas fuera del horario de clases, dicha petición fue ignorada.

3.6 Técnicas estadísticas

Para el análisis de los datos recolectados, con la finalidad de cumplir el tercer y cuarto objetivo específico de esta investigación, se utilizó la técnica estadística *prueba t*, además del cálculo del *alfa de Cronbach* para la confiabilidad de las sub-escalas de motivación.

3.6.1 Alfa de Cronbach

Como se menciona en (Ledesma, Molina, & Valero, 2002) dentro de la Teoría Clásica de los Tests (TCT) el método de consistencia interna es el procedimiento más usual para estimar la fiabilidad de pruebas, escalas o test, cuando se utilizan conjuntos de elementos o reactivos que se espera midan el mismo atributo o campo de contenido.

Cuando los elementos se utilizan para formar una escala, estos necesitan tener consistencia interna. Los elementos deben medir lo mismo, así que deben estar correlacionados entre sí. Un coeficiente de utilidad para evaluar la consistencia interna es el alfa de Cronbach (Cronbach, 1951).

El coeficiente alfa fue descrito en 1951 por Lee J. Cronbach. Es un índice usado para medir la confiabilidad del tipo consistencia interna de una escala, es decir, para evaluar la magnitud en que los elementos de un instrumento están correlacionados (Oviedo & Campo-Arias, 2005). En otras palabras, el alfa de Cronbach es el promedio de las correlaciones entre los elementos que forman parte de un instrumento (Streiner, 2003a). La fórmula para obtener el alfa de Cronbach es la siguiente:

$$\alpha = \left(\frac{n}{n-1} \right) \left(1 - \frac{\sum S_i^2}{S_{sum}^2} \right)$$

Donde n es el número de elementos de la prueba, S_i^2 es la varianza de los elementos desde 1 hasta i mientras que S_{sum}^2 es la varianza total de la suma de todos los elementos.

Para que el valor de la consistencia interna de una escala sea aceptable, este debe ser mínimo de 0.70, por debajo de este valor, la consistencia interna de una escala se considera baja, mientras que el valor máximo esperado es 0.90, si se obtiene un valor mayor de 0.90 se considera que hay redundancia o duplicación, es decir, varios elementos de la escala están midiendo lo mismo dentro de un constructo; por lo tanto, los elementos redundantes deben eliminarse. Los valores preferibles de alfa se encuentran entre 0.80 y 0.90 (Streiner, 2003b).

3.6.2 Prueba t para muestras relacionadas o dependientes

La *prueba t* es una prueba estadística utilizada para comparar los resultados de una pre prueba con los resultados de una post prueba en un contexto experimental. Se comparan las medias y las varianzas del grupo en dos momentos diferentes. La hipótesis de investigación propone que existen diferencias significativas entre los grupos, por el contrario la hipótesis nula propone que no existen diferencias significativas.

La comparación se realiza sobre una variable, si hay n variables diferentes, se realizan n pruebas t , una por cada variable (Sampieri, Collado, Lucio, & Pérez, 1998).

Existen tres fórmulas diferentes para la prueba t: prueba t general, prueba t para muestras independientes y prueba t para muestras relacionadas o dependientes. El uso de la primera fórmula, es decir prueba t general, es apropiado cuando se compara el rendimiento de una variable dependiente de una muestra de participantes con el rendimiento de la población de la que se extrajo la muestra. En el caso de la fórmula para muestras independientes se compara el rendimiento de dos grupos distintos en la misma variable dependiente. La fórmula de prueba t para muestras relacionados o dependientes se usa cuando se desea comparar el desempeño de las mismas personas en la misma variable dependiente, como en un diseño pre prueba – post prueba, (Turner, 2014) esta última es la utilizada en la investigación.

La prueba t para muestras relacionadas o dependiente compara dos medias de una misma persona, objeto o unidades relacionadas. Las dos medias representan dos momentos diferentes (ej. pre prueba, post prueba) o dos condiciones o unidades (ej. Ojo izquierdo y derecho) diferentes pero relacionadas.

El objetivo de esta prueba es determinar si existe evidencia estadística de que la diferencia media entre los pares de observaciones en un resultado particular es significativamente diferente de cero. Este tipo de prueba es paramétrica y también se le conoce como prueba t parida y prueba t de medidas repetidas (Yeager, s/f).

El primer paso es establecer las hipótesis (nula y alterna), una vez establecidas las hipótesis de la investigación, se establece el valor de significancia para la aceptación o rechazo de la hipótesis nula, en la mayoría de los casos se utiliza un valor de significancia de 5% (0.05). El siguiente paso es la obtención de t, el cual se obtiene mediante la siguiente formula:

$$t = \frac{\bar{X}_{diff} - 0}{S_{\bar{x}}}$$

Donde:

$$S_{\bar{x}} = \frac{S_{diff}}{\sqrt{n}}$$

Donde \bar{X}_{diff} es la media de las diferencias, n es el número de observaciones, S_{diff} corresponde a la desviación estándar de las diferencias y $S_{\bar{x}}$ es el error estándar de la media.

Después de obtener t , este se compara con el valor t crítico con $gl = n - 1$ (gl = grados de libertad) a partir de la tabla de distribución t (ver **ANEXO H**) para el nivel de confianza escogido (0.05). Si el valor t calculado es mayor que el valor crítico t , entonces se rechaza la hipótesis nula.



CAPÍTULO IV – RESULTADOS

En este capítulo se reportan los resultados del análisis estadístico realizado tanto de la prueba piloto así como de la réplica, las cuales fueron realizadas en noviembre de 2015 y marzo de 2016 respectivamente. Todos los análisis se realizaron con el programa SPSS versión 19 en un entorno Windows.

4.1 Prueba piloto noviembre 2015

Como se mencionó en el Capítulo III, en la prueba piloto participaron en total 111 estudiantes, de los cuales solo 79 participaron en las dos fases de la prueba piloto y solo estos fueron tomados en cuenta para realizar el análisis estadístico debido a las necesidades de la investigación, teniendo un 28% de mortandad experimental.

4.1.1 Resultados de información demográfica de los participantes

Los resultados de la información demográfica se muestran en la Tabla 3, en la cual se muestra el total de participantes así como la edad promedio de los mismos y su desviación estándar, además del porcentaje de experiencia en programación y el porcentaje de hombres y mujeres.

Tabla 3. Resultados información demográfica durante la prueba piloto

Criterio	Total	
Participantes	79	
Sexo	Hombres	Mujeres
	54 % (43)	46 % (36)
Edad promedio	17	
Desviación estándar de edad	0.41	
Porcentaje experiencia previa en programación	Si	No
	67 % (53)	33 % (26)

4.1.2 Resultados pre prueba de aprendizaje de la programación

En la Figura 11 se ilustran los resultados correspondientes a las calificaciones obtenidas durante la pre prueba además del número de alumnos que obtuvo cada una de ellas. Las calificaciones posibles son 0, 25, 50, 75 y 100 ya que la pre prueba de aprendizaje de la programación consistió de 4 ejercicios con valor de 25 cada uno de ellos.

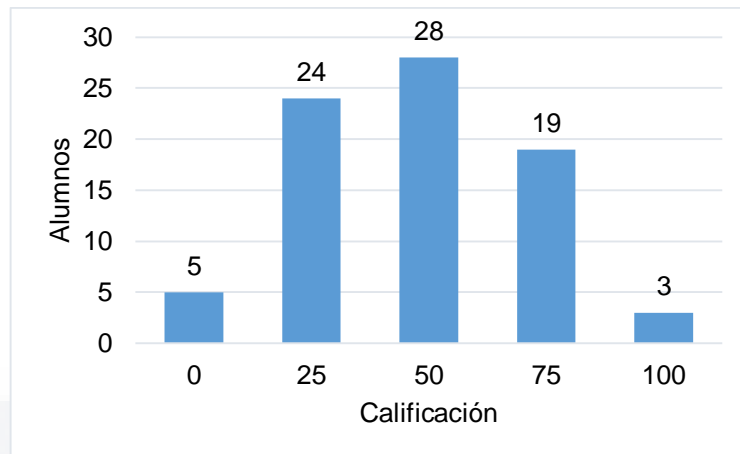


Figura 11. Calificaciones de la pre prueba durante la prueba piloto

Como se observa en la Figura 11 la calificación más recurrente es 50 con 28 alumnos mientras que la menos recurrente es 100 con tan solo 3 alumnos, además se observa que la mayoría de los alumnos obtuvo calificaciones de 25 y 50. La **media** de las calificaciones de la pre prueba durante la prueba piloto es de **47.15**.

4.1.3 Resultados post prueba de aprendizaje de la programación

En la Figura 12 se ilustran las calificaciones obtenidas durante la post prueba y el número de alumnos que obtuvo cada una de ellas. Como se puede observar, al igual que en la pre prueba, la calificación más recurrente es 50 con 31 alumnos y la menos recurrente en este caso es 0 con solo 3 alumnos además se observa que la mayoría de los alumnos obtuvieron 50 y 75 de calificación.

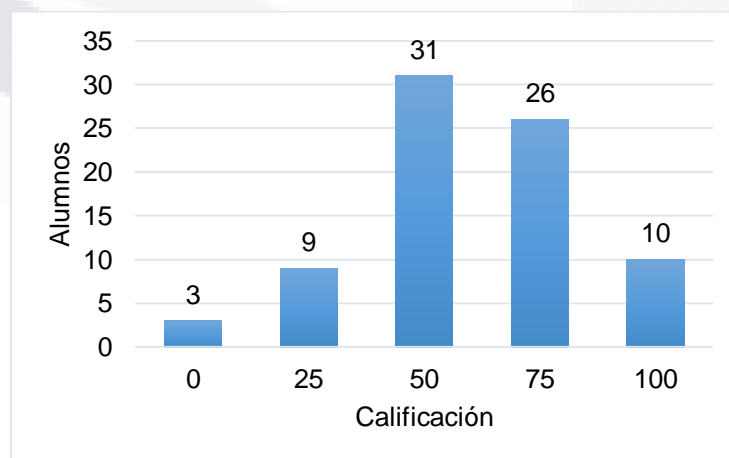


Figura 12. Calificaciones de la post prueba durante la prueba piloto

La **media** de las calificaciones de la post prueba durante la prueba piloto es de **59.81**.

En la Tabla 4 se muestra el contraste entre la pre prueba y la post prueba de aprendizaje de la programación de cada calificación posible y en la Figura 13 se muestra de manera gráfica la diferencia entre ellas.

Tabla 4. Contraste post prueba - pre prueba de calificaciones durante la prueba piloto

Calificación	Contraste Post prueba – pre prueba
0	-2
25	-15
50	3
75	7
100	7

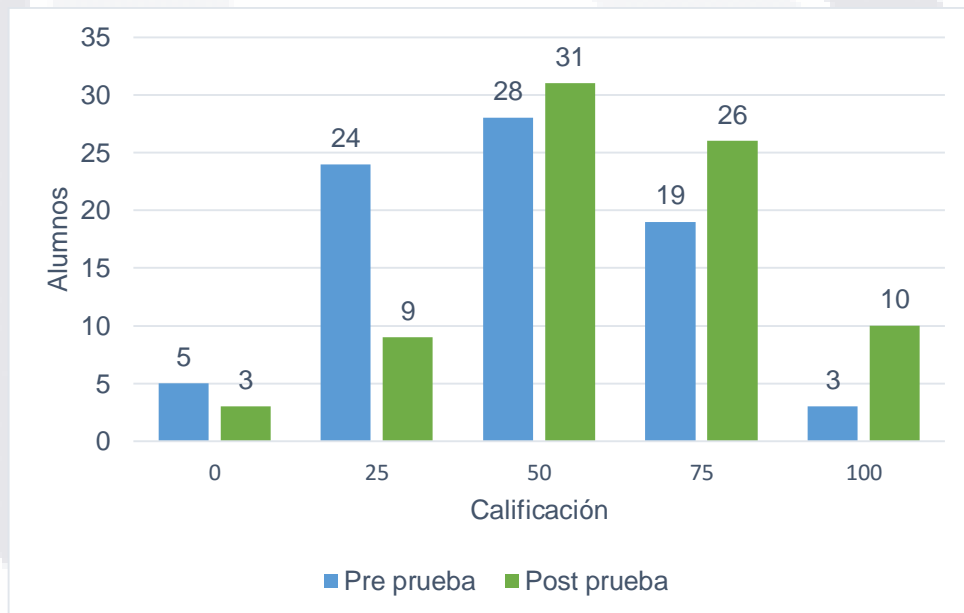


Figura 13. Calificaciones de las pre y post pruebas durante la prueba piloto

Como se observa en la Figura 13, durante la post prueba disminuyó el número de alumnos que obtuvieron 0 y 25 como calificación, lo cual a simple vista indica que hubo un aumento positivo en cuanto a la calificación de los alumnos al hacer uso del LPE Scratch, más adelante se analiza si la diferencia entre la pre y post prueba es estadísticamente significativa.

4.1.4 Resultados pre prueba de motivación

Como se mencionó anteriormente, para la investigación solo se tomó en cuenta la escala de motivación del MSLQ la cual consta de 6 sub-escalas. En la Tabla 5 se muestra el resultado al calcular la confiabilidad de cada una de las sub-escalas al obtener el alfa de Cronbach de la pre prueba de motivación durante la prueba piloto, en ella se observa que la confiabilidad de cada una de las sub-escalas es aceptable.

Tabla 5. Alfa de Cronbach por sub-escala en la pre prueba durante la prueba piloto

Sub-escala	Alfa de Cronbach	Alfa de Cronbach estandarizado
Orientación Intrínseca	0.818	0.816
Orientación Extrínseca	0.721	0.722
Valor de la tarea	0.896	0.895
Creencia de control	0.690	0.724
Autoeficacia	0.910	0.915
Ansiedad a examen	0.803	0.803

En la Tabla 6 se visualiza la media y desviación estándar de cada una de las sub-escalas de motivación en la pre prueba durante la prueba piloto, como se observa, las primeras cinco sub-escalas obtuvieron 5 como respuesta de acuerdo a la escala Likert del cuestionario, lo cual significa que los alumnos se identifican lo suficiente con cada una de las preguntas realizadas, ya que un puntaje de 4 a 7 es mejor que un puntaje de 1 a 3 para estas cinco sub-escalas. La escala que obtuvo una mayor puntuación es Autoeficacia, concluyendo que los alumnos confían en que tienen las habilidades necesarias para desempeñarse satisfactoriamente.

Tabla 6. Media y desviación estándar por sub-escala en la pre prueba durante la prueba piloto

Sub-escala	Media	Desviación estándar
Orientación intrínseca	5.51	1.04
Orientación extrínseca	5.36	1.17
Valor de la tarea	5.50	1.16
Creencia de control	5.38	1.06
Autoeficacia	5.58	0.96
Ansiedad a examen	4.20	1.31

En la Tabla 6 también se observa que la sub-escala Ansiedad a examen tuvo un puntaje menor a las sub-escalas anteriores con tan solo 4 puntos, pero en esta escala, a diferencia de las demás, mientras el puntaje sea mayor, mayor será el grado de ansiedad, por lo tanto

se deduce que los alumnos participantes no se identifican demasiado con las preguntas de la subescala Ansiedad a examen de acuerdo al valor medio obtenido.

4.1.5 Resultados post prueba de motivación

En la Tabla 7 se muestra el alfa de Cronbach obtenido en la post prueba durante la prueba piloto en la cual se observa, al igual que en la pre prueba, que es aceptable la confiabilidad de cada una de las sub-escalas.

Tabla 7. Alfa de Cronbach por sub-escala en la post prueba durante la prueba piloto

Sub-escala	Alfa de Cronbach	Alfa de Cronbach estandarizado
Orientación Intrínseca	0.844	0.848
Orientación Extrínseca	0.721	0.730
Valor de la tarea	0.924	0.924
Creencia de control	0.655	0.686
Autoeficacia	0.916	0.921
Ansiedad a examen	0.863	0.862

En la Tabla 8 se muestran los resultados obtenidos en la post prueba de motivación durante la prueba piloto, en este caso las medias por sub-escala y la desviación estándar.

Tabla 8. Media y desviación estándar por sub-escala en la post prueba durante la prueba piloto

Sub-escala	Media	Desviación estándar
Orientación Intrínseca	5.77	1.02
Orientación Extrínseca	5.62	1.07
Valor de la tarea	5.72	1.13
Creencia de control	5.67	0.92
Autoeficacia	5.84	0.87
Ansiedad a examen	4.30	1.43

Al igual que en la pre prueba, las primeras cinco sub-escalas obtuvieron 5 como respuesta y la sub-escala con mayor puntuación es Autoeficacia y la sub-escala Ansiedad a examen es la que cuenta con la menor puntuación.

En la Figura 14 se muestra gráficamente la comparación de los resultados obtenidos en la pre prueba y post prueba de motivación. Como en ella se observa, las seis sub-escalas tuvieron una pequeña diferencia “positiva” en la post prueba en comparación con la pre prueba durante la prueba piloto a excepción de la sub-escala Ansiedad a examen, ya que aunque aumento un poco su valor, no es un aumento positivo puesto que a mayor valor en

esta sub-escala, mayor es el grado de ansiedad a exámenes de los alumnos. Más adelante se analiza si las diferencias encontradas son estadísticamente significativas.

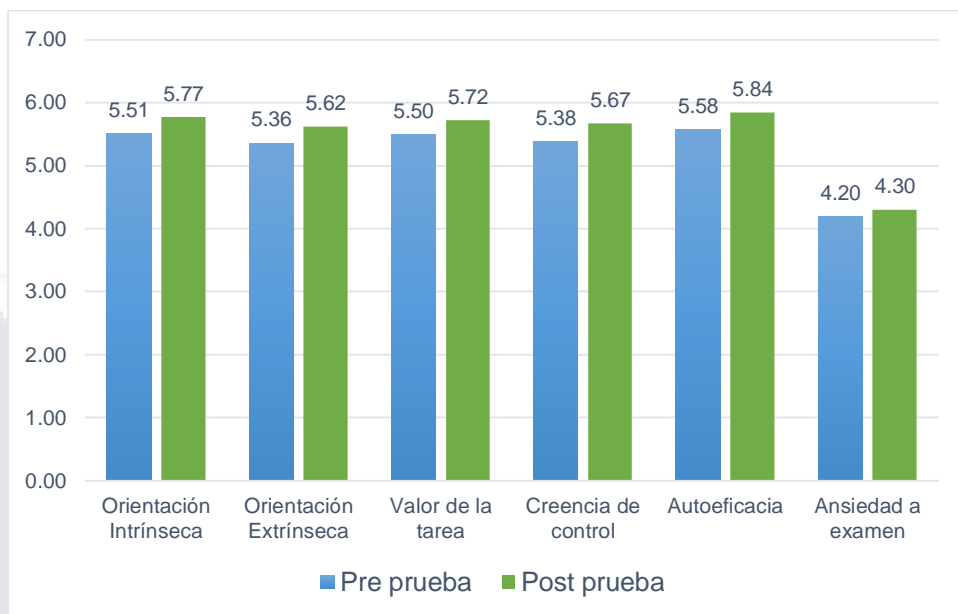


Figura 14. Resultados de las pre y post pruebas de motivación durante la prueba piloto

4.2 Réplica 2016

En la réplica llevada a cabo en Marzo de 2016 participaron 67 estudiantes, de los cuales solo 33 participaron en las dos fases y solo estos fueron tomados en cuenta para realizar el análisis estadístico debido a las necesidades de la investigación, teniendo un 49% de mortandad experimental, debido a que los alumnos tuvieron que atender algunas otras actividades requeridas por el BachUAA, fuera de este, en los días en que se realizó la réplica.

4.2.1 Resultados de información demográfica de los participantes

La información demográfica de la réplica se muestran en la Tabla 9, en la cual se muestra el total de participantes así como la edad promedio de los mismos y su desviación estándar, además del porcentaje de experiencia en programación y el porcentaje del sexo de los participantes.

Tabla 9. Resultados información demográfica durante la réplica

Criterio	Total
----------	-------

Participantes	33	
Sexo	Hombres	Mujeres
	82 % (27)	18 % (6)
Edad promedio	17	
Desviación estándar de edad	0.71	
Porcentaje experiencia previa en programación	Si	No
	24 % (8)	76 % (25)

4.2.2 Resultados pre prueba de aprendizaje de la programación

En la Figura 15 se ilustran los resultados correspondientes a las calificaciones obtenidas durante la pre prueba y el número de alumnos que obtuvo cada una de ellas. Al igual que en la prueba piloto la pre prueba de programación consistió de cuatro ejercicios, con valor de 25 cada uno de ellos, por lo tanto las calificaciones posibles son 0, 25, 50, 75 y 100.

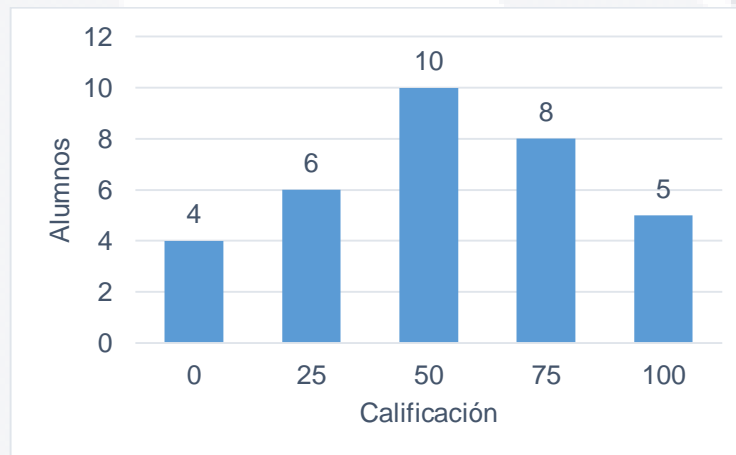


Figura 15. Calificaciones de la pre prueba durante la réplica

Como se muestra en la Figura 15 la calificación más recurrente es 50 con 10 alumnos mientras que la menos recurrente es 0 con tan solo 4 alumnos, al igual se observa que la mayoría de los alumnos obtuvo calificaciones de 50 y 75. La **media** de las calificaciones de la pre prueba durante la réplica es de **53.03**.

4.2.3 Resultados post prueba de aprendizaje de la programación

En la Figura 16 se muestran los resultados obtenidos, como se observa a diferencia de la pre prueba, la calificación más recurrente es 75 con 12 alumnos y la menos recurrente es 0 con solo 1 alumno, además se observa que la mayoría de los alumnos obtuvieron 50 y 75 de calificación al igual que en la pre prueba. La **media** de las calificaciones de la post prueba durante la réplica es de **59.85**.

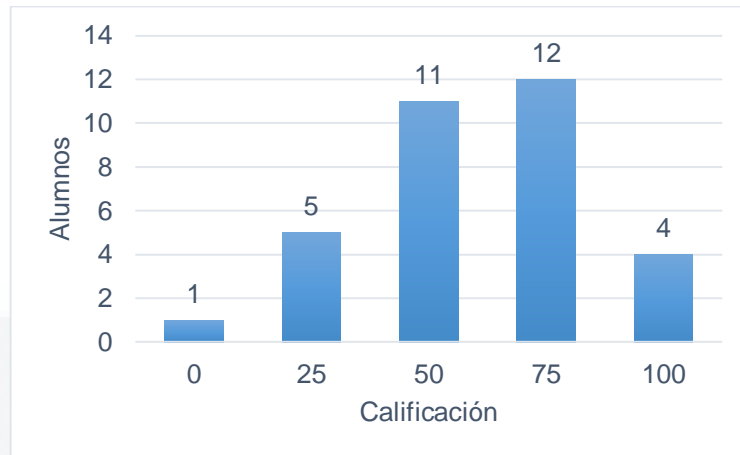


Figura 16. Calificaciones de la post prueba durante la réplica

En la Tabla 10 se muestra el contraste entre la pre prueba y la post prueba de aprendizaje de la programación y en la Figura 17 se muestra de manera gráfica la diferencia entre ellas.

Tabla 10. Contraste post prueba - pre prueba de calificaciones durante la réplica

Calificación	Contraste Post prueba – pre prueba
0	-3
25	-1
50	1
75	4
100	-1

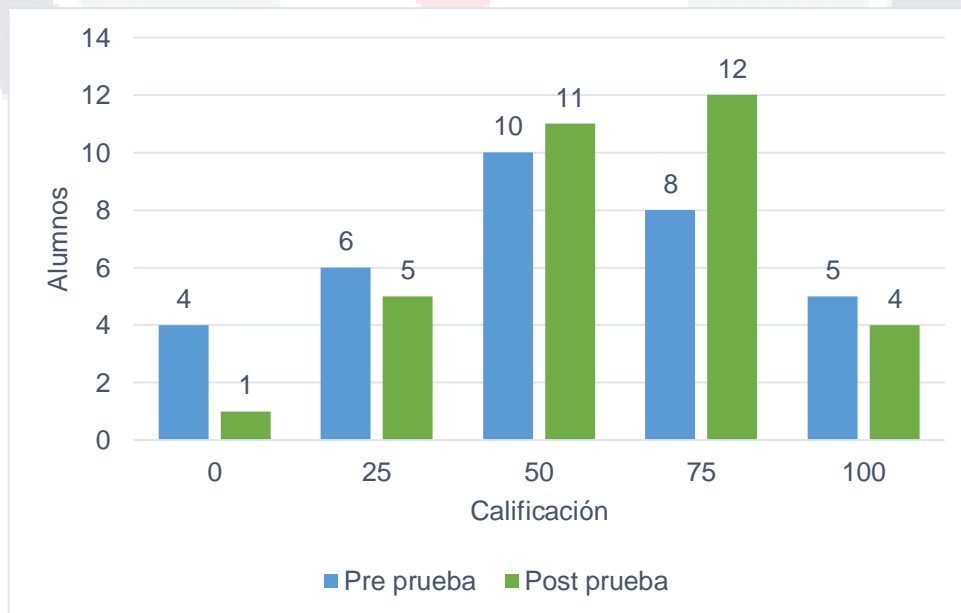


Figura 17. Calificaciones de las pre y post pruebas durante la prueba piloto

Como se observa en la Figura 17, durante la post prueba disminuyó el número de alumnos que obtuvieron 0 y 25 como calificación, a pesar de que el número de alumnos que obtuvo 100 como calificación también disminuyó (-1) aumento el número de alumnos que obtuvo 50 y 75, lo cual a simple vista indica que hubo un aumento positivo en cuanto a la calificación de los alumnos al hacer uso del LPE Scratch, más adelante se analiza si la diferencia entre la pre y post prueba es estadísticamente significativa.

4.2.4 Resultados pre prueba de motivación

En la Tabla 11 se muestra el resultado al calcular la confiabilidad de cada una de las sub-escalas al obtener el alfa de Cronbach de la pre prueba de motivación durante la réplica, en la cual se observa que la confiabilidad de las sub-escalas Orientación intrínseca, Orientación extrínseca y Creencias de control es cuestionable, mientras que la confiabilidad de las sub-escalas Valor de la tarea, Autoeficacia y Ansiedad a examen es de aceptable a excelente.

Tabla 11. Alfa de Cronbach por sub-escala en la pre prueba durante la réplica.

Sub-escala	Alfa de Cronbach	Alfa de Cronbach estandarizado
Orientación intrínseca	0.540	0.593
Orientación extrínseca	0.624	0.652
Valor de la tarea	0.905	0.907
Creencia de control	0.593	0.579
Autoeficacia	0.874	0.878
Ansiedad a examen	0.750	0.757

En la Tabla 12 se muestra la media y desviación estándar de cada sub-escala de motivación de la pre prueba durante la réplica, se observa que las primeras cinco sub-escalas obtuvieron valor de 5 como respuesta de acuerdo a la escala Likert del cuestionario, la cual va de 1 a 7. La escala que obtuvo la puntuación mayor es Valor de la tarea, concluyendo que los alumnos consideran interesante, importante y útil el material del curso de programación impartido antes del tratamiento.

Tabla 12. Media y desviación estándar por sub-escala en la pre prueba durante la réplica

Sub-escala	Media	Desviación estándar
Orientación intrínseca	5.57	0.79
Orientación extrínseca	5.15	1.12
Valor de la tarea	5.86	0.98
Creencia de control	5.38	0.98
Autoeficacia	5.59	0.77

Ansiedad a examen	4.12	1.15
-------------------	------	------

En la Tabla 12 asimismo se visualiza que la sub-escala con la menor puntuación es, al igual que en la prueba piloto, Ansiedad a examen, concluyendo que los alumnos se identifican menos con las preguntas de esta sub-escala, la cual obtuvo el menor valor medio (4.12).

4.2.5 Resultados post prueba de motivación

El alfa de Cronbach correspondiente a cada una de las sub-escalas de motivación de la post prueba durante la réplica se muestra en la Tabla 13, en ella se observa que la confiabilidad de la mayoría de las sub-escalas es de aceptable a excelente a excepción de la sub-escala Orientación extrínseca, la cual obtuvo un valor menor a 0.7.

Tabla 13. Alfa de Cronbach por sub-escala en la post prueba durante la réplica

Sub-escala	Alfa de Cronbach	Alfa de Cronbach estandarizado
Orientación intrínseca	0.749	0.746
Orientación extrínseca	0.686	0.700
Valor de la tarea	0.937	0.942
Creencia de control	0.845	0.853
Autoeficacia	0.948	0.948
Ansiedad a examen	0.834	0.836

La media y desviación estándar de cada una de las sub-escalas se muestra en la Tabla 14. Al igual que en la pre prueba durante la réplica las primeras cinco sub-escalas obtuvieron valor de 5 como respuesta y la sub-escala con mayor puntuación es Valor de la tarea mientras que la subescala con menor puntuación es Ansiedad a examen.

Tabla 14. Media y desviación estándar por sub-escala en la post prueba durante la réplica

Sub-escala	Media	Desviación estándar
Orientación intrínseca	5.37	1.02
Orientación extrínseca	5.03	1.24
Valor de la tarea	5.67	1.16
Creencia de control	5.36	1.18
Autoeficacia	5.47	1.09
Ansiedad a examen	4.23	1.22

La Figura 18 corresponde a la comparación grafica de los resultados obtenidos en la pre prueba y post prueba de motivación durante la réplica, en ella se observa que las seis sub-escalas tuvieron una pequeña diferencia, en este caso, a diferencia de la prueba piloto, disminuyeron su valor en la post prueba en comparación con la pre prueba durante la

réplica. Más adelante se analiza si las diferencias encontradas son estadísticamente significativas.

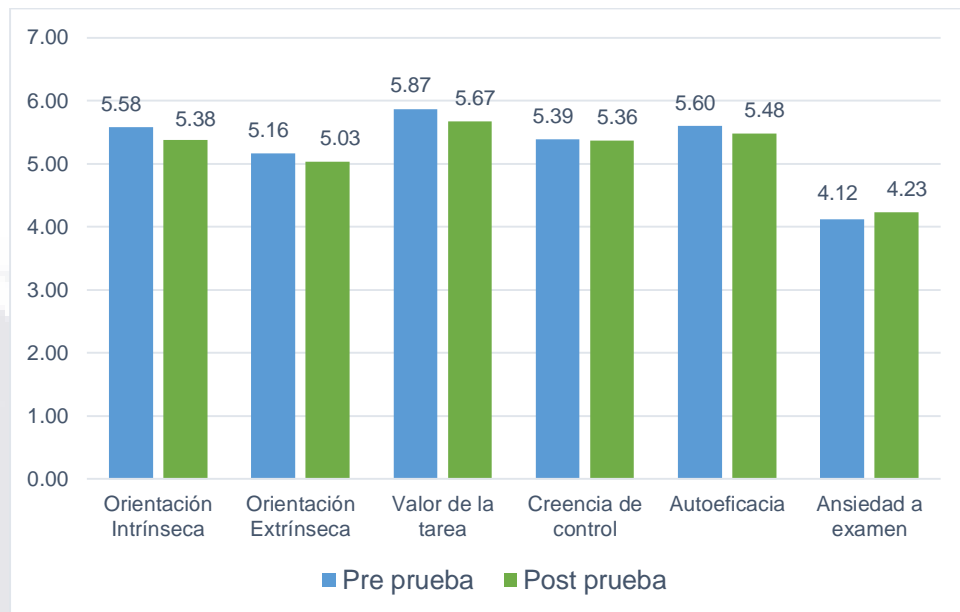


Figura 18. Resultados de las pre y post pruebas de motivación durante la réplica

4.3 Análisis estadístico de los resultados

En esta sección se presenta el análisis estadístico tanto de las pre pruebas así como de las post pruebas de aprendizaje de la programación así como de motivación durante la prueba piloto y la réplica.

4.3.1 Prueba piloto 2015

En este apartado se especifica el análisis estadístico de las pre y post pruebas durante la prueba piloto con el fin de identificar si existen diferencias estadísticamente significativas antes y después del tratamiento haciendo uso de Scratch.

4.3.1.1 Prueba t para muestras relacionadas de la variable *aprendizaje de la programación*

La prueba estadística aplicada a los datos recolectados durante la prueba piloto es la *prueba t*. Las evaluaciones de las muestras duplicadas para la *prueba t* se llevaron a cabo con la finalidad de identificar si existen diferencias estadísticamente significativas antes y

después de la aplicación de un tratamiento haciendo uso del LPE Scratch en el contexto del aprendizaje de la programación básica.

Los datos estadísticos obtenidos se presentan en la Tabla 15, el dato a destacar en esta tabla es la media antes y después de la aplicación del tratamiento, la cual es de 47.15 y 59.81 respectivamente.

Tabla 15. Datos estadísticos pre y post pruebas de aprendizaje de la programación durante la prueba piloto

	Media	N	Desviación estándar
Antes	47.15	79	24.34
Después	59.81	79	24.48

Los resultados de la *prueba t* para muestras relacionadas de la pre y post pruebas se presentan en la Tabla 16 en la que se describen la diferencia media, la desviación estándar de las diferencias y la prueba *t*, la cual incluye el valor de *t*, los grados de libertad y el valor *p* resultante, con un intervalo de confianza de 95%.

Tabla 16. Resultados prueba *t* pre y post pruebas de aprendizaje de la programación durante la prueba piloto

	Media	Des. estándar	t	Grados de libertad	P
Antes - Después	-12.658	31.99	-3.517	78	0.001

Para verificar si las diferencias encontradas entre la pre prueba y post prueba son estadísticamente significativas debe observarse el valor de probabilidad asociado al estadístico, es decir, el valor *p*. Al estar por debajo de 0.05, concretamente en este caso es igual a 0.001, la hipótesis nula tiene poca probabilidad de ocurrencia y debe rechazarse, por lo tanto las diferencias entre la pre prueba ($\bar{X} = 47.15$) y la post prueba ($\bar{X} = 59.81$) de aprendizaje de programación durante la prueba piloto son significativas, siendo más altas en esta última, por lo cual se concluye que el tratamiento en Scratch mejoró el nivel de conocimientos sobre programación básica ($t(78) = -3.517, p < 0.05$) (véase Figura 19).

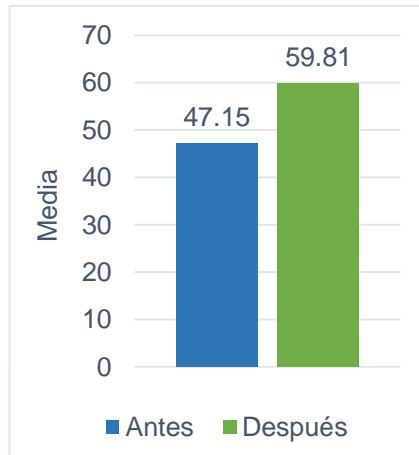


Figura 19. Medias pre y post pruebas de aprendizaje de la programación durante la prueba piloto

4.3.1.2 Prueba t para muestras relacionadas de la variable motivación

La prueba estadística aplicada a los datos recolectados, durante la prueba piloto, de la variable motivación al igual que en el aprendizaje de la programación es la *prueba t*. Las evaluaciones de las muestras duplicadas para la *prueba t* se llevaron a cabo con la finalidad de identificar si existen diferencias estadísticamente significativas en la motivación de los estudiantes antes y después de la aplicación de un tratamiento, en el contexto del aprendizaje de la programación básica, haciendo uso del LPE Scratch.

Los datos estadísticos obtenidos, media y desviación estándar, de cada una de las sub-escalas de motivación se presentan en la Tabla 17, en ella se observa que la media de las sub-escalas para el caso de la pre prueba oscilan entre 4.20 y 5.58 siendo estos los valores para las sub-escalas Ansiedad a examen y Autoeficacia respectivamente. Para el caso de la post prueba, el valor mínimo y máximo corresponden, al igual que en la pre prueba, a las sub-escalas Ansiedad a examen con 4.30 y Autoeficacia con 5.84, de igual manera se observa que todas las sub-escalas tuvieron un incremento después del tratamiento en Scratch.

Tabla 17. Datos estadísticos pre y post pruebas de motivación durante la prueba piloto

Sub-escala	Pre prueba (N=79)		Post prueba (N=79)	
	Media	Desviación estándar	Media	Desviación estándar
Orientación Intrínseca	5.51	1.04	5.77	1.02
Orientación Extrínseca	5.36	1.17	5.62	1.07

Valor de la tarea	5.50	1.16	5.72	1.13
Creencia de control	5.38	1.06	5.67	0.92
Autoeficacia	5.58	0.96	5.84	0.87
Ansiedad a examen	4.20	1.31	4.30	1.43

Los resultados de la *prueba t* para muestras relacionadas de la pre y post pruebas de motivación se muestran en la Tabla 18 en la cual se presenta la diferencia media, la desviación estándar de las diferencias y la prueba *t*, la cual incluye el valor de *t*, los grados de libertad y el valor *p* resultante, con un intervalo de confianza de 95% para cada una de las sub-escalas.

Tabla 18. Resultados prueba *t* pre y post pruebas de motivación durante la prueba piloto

Sub-escala	Media	Des. estándar	t	Grados de libertad	p
Orientación Intrínseca	-0.259	0.832	-2.771	78	0.007
Orientación Extrínseca	-0.259	0.957	-2.408	78	0.018
Valor de la tarea	-0.213	0.852	-2.221	78	0.029
Creencia de control	-0.287	0.984	-2.599	78	0.011
Autoeficacia	-0.262	0.870	-2.683	78	0.009
Ansiedad a examen	-0.093	1.384	-0.602	78	0.549

Para verificar si las diferencias encontradas entre la pre prueba y post prueba son estadísticamente significativas es necesario comparar el valor de probabilidad asociado al estadístico, es decir, el valor *p* para cada una de las sub-escalas. Al estar por debajo de 0.05 la hipótesis nula tiene poca probabilidad de ocurrencia y debe rechazarse, por lo tanto, en la Tabla 18 se observa que se encontró un aumento estadísticamente significativo en las sub-escalas Orientación Intrínseca, Orientación Extrínseca, Valor de la tarea, Creencia de control y Autoeficacia puesto que el valor de *p* en todos los casos es menor que 0.05. No se encontraron diferencias estadísticamente significativas antes y después del tratamiento en la sub-escala Ansiedad a examen.

4.3.2 Replica 2016

En este apartado se especifica el análisis estadístico de las pre y post pruebas durante la réplica con el fin de identificar si existen diferencias estadísticamente significativas antes y después del tratamiento haciendo uso del LPE Scratch.

4.3.2.1 Prueba t para muestras relacionadas de la variable aprendizaje de la programación

Al igual que en la prueba piloto, la prueba estadística aplicada a los datos recolectados durante la réplica es la *prueba t*. Esta prueba se llevó a cabo con la finalidad de identificar si existen o no diferencias estadísticamente significativas antes y después de la aplicación de un tratamiento haciendo uso de Scratch en el contexto del aprendizaje de la programación básica.

Los datos estadísticos obtenidos se muestran en la Tabla 19 en la cual se observa que la media obtenida antes y después de la aplicación del tratamiento es 53.03 y 59.85 respectivamente.

Tabla 19. Datos estadísticos pre y post pruebas de aprendizaje de la programación durante la réplica

	Media	N	Desviación estándar
Antes	53.03	33	31.09
Después	59.85	33	24.95

En la Tabla 20 se muestran los resultados de la *prueba t*, en ella se especifica la diferencia media así como la desviación estándar de las medias y la prueba t, la cual incluye el valor de *t*, los grados de libertad y el valor *p*, con un intervalo de confianza de 95%.

Tabla 20. Resultados prueba t pre y post pruebas de aprendizaje de la programación durante la réplica

	Media	Des. estándar	T	Grados de libertad	P
Antes - Después	-6.81	33.23	-1.179	32	0.247

Para determinar si las diferencias encontradas entre la pre prueba y post pruebas de aprendizaje de la programación básica son estadísticamente significativas, se observa el valor de probabilidad asociado al estadístico, el valor *p*, al estar por debajo de 0.05, la hipótesis nula debe rechazarse, en este caso el valor *p* es mayor que 0.05 siendo este 0.247, por lo tanto las diferencias entre la pre prueba ($\bar{X} = 53.03$) y la post prueba ($\bar{X} = 59.85$) de aprendizaje de programación durante la prueba piloto no son estadísticamente significativas, por lo tanto no se rechaza la hipótesis nula, por lo que el tratamiento en Scratch no mejoró significativamente el desempeño de los alumnos en el contexto de la programación básica.

4.3.2.2 Prueba t para muestras relacionadas de la variable motivación

Se llevaron a cabo evaluaciones de las muestras duplicadas para la *prueba t* con la finalidad de identificar si existen diferencias estadísticamente significativas en la motivación de los estudiantes antes y después de la aplicación de un tratamiento, en el contexto del aprendizaje de la programación básica, haciendo uso del LPE Scratch.

En la Tabla 21 se muestran los datos estadísticos obtenidos, media y desviación estándar correspondientes a cada una de las sub-escalas, se observa que las medias de las sub-escalas de la pre prueba se encuentran entre 4.12 y 5.86 dichos valores pertenece a las sub-escalas Ansiedad a examen y Valor de la tarea respectivamente. En lo que respecta a la post prueba, los valores mínimo y máximo corresponden al igual que en la pre prueba a las sub-escalas Ansiedad a examen y Valor de la tarea con valores de 4.23 y 5.67 respectivamente, además se observa que todas las sub-escalas, a excepción de la sub-escala Ansiedad a examen, reportaron un valor menor en la post prueba en comparación a la pre prueba durante la réplica después del tratamiento en Scratch, lo cual no es un resultado esperado.

Tabla 21. Datos estadísticos pre y post pruebas de motivación durante la réplica

Sub-escala	Pre prueba (N=33)		Post prueba (N=33)	
	Media	Desviación estándar	Media	Desviación estándar
Orientación intrínseca	5.57	0.79	5.37	1.02
Orientación extrínseca	5.15	1.12	5.03	1.24
Valor de la tarea	5.86	0.98	5.67	1.16
Creencia de control	5.38	0.98	5.36	1.18
Autoeficacia	5.59	0.77	5.47	1.09
Ansiedad a examen	4.12	1.15	4.23	1.22

En la Tabla 22 se muestran los resultados de la *prueba t*, en ella se especifica la diferencia media así como la desviación estándar de las medias y la prueba t, la cual incluye el valor de t, los grados de libertad y el valor p, con un intervalo de confianza de 95%.

Tabla 22. Resultados prueba t pre y post pruebas de motivación durante la réplica

Sub-escala	Media	Des. estándar	t	Grados de libertad	p
Orientación Intrínseca	0.196	0.751	1.505	32	0.142
Orientación Extrínseca	0.128	0.8007	0.916	32	0.367
Valor de la tarea	0.197	0.624	1.818	32	0.078
Creencia de control	0.022	1.120	0.117	32	0.908

Autoeficacia	0.121	0.717	0.971	32	0.339
Ansiedad a examen	-0.109	1.212	-0.517	32	0.609

Para verificar si las diferencias encontradas entre la pre prueba y post prueba son estadísticamente significativas se comparó el valor de probabilidad asociado al estadístico, es decir, el valor p para cada una de las sub-escalas. Al estar por debajo de 0.05 la hipótesis nula tiene poca probabilidad de ocurrencia y debe rechazarse, por lo tanto, en la Tabla 22 se observa que no existen diferencias estadísticamente significativas en la motivación de los estudiantes antes y después del tratamiento en Scratch, ya que el valor de p para todas las subescalas es mayor que 0.05.



CONCLUSIONES

La investigación consistió en un cuasi-experimento con el objetivo de identificar si el uso de un lenguaje de programación educativo (LPE) visual produce diferencias estadísticamente significativas en la motivación y en la adquisición de habilidades de resolución de problemas en el ámbito del aprendizaje de la programación básica en estudiantes de educación media superior. Para ello, cumpliendo con el objetivo específico número 1 se identificaron estudios relacionados para determinar cuál LPE utilizar eligiendo el LPE Scratch debido a sus características lúdicas y a la variedad de estudios que han hecho uso de él como apoyo en materias STEM además del aprendizaje de la programación básica.

Para el cumplimiento del objetivo específico número 2, se realizó una revisión de la literatura para conocer qué instrumentos de medición de la motivación existen, encontrando dos posibles, el IMI (Intrinsic Motivation Inventory) y el MSLQ, descartando el primero por ser muy extenso y seleccionando el cuestionario MSLQ, ya que fue el que mejor se adaptaba a las necesidades de la investigación, utilizando solo la escala que mide dicha variable.

En cuanto al cumplimiento del objetivo específico número 3 se diseñó un curso en Scratch con el fin de dar a conocer a los participantes los conceptos básicos de programación básica, haciendo énfasis en la creación de algoritmos basados en el modelo de Polya descrito en el Capítulo II y de acuerdo a lo recomendado por el personal académico del BACHUAA. Para el cumplimiento del objetivo específico número 4 se diseñaron las pre y post prueba de aprendizaje de la programación de acuerdo a especificaciones por el personal académico del BACHUAA con el grado de dificultad necesario de acuerdo al nivel educativo en el que se aplicaron.

Con respecto al objetivo general de la investigación e incluyendo los objetivos específicos 5 y 6 al realizar el análisis estadístico llevado a cabo se identificó que durante la prueba piloto existen diferencias estadísticamente significativas en la adquisición de habilidades de resolución de problemas así como en la motivación de los estudiantes al hacer uso del LPE Scratch en el contexto del aprendizaje de la programación básica en educación media superior; en contraste con los resultados obtenidos de acuerdo al análisis estadístico durante la réplica, se identificó que no existen diferencias estadísticamente significativas en la motivación de los estudiantes ni en la adquisición de habilidades de resolución de problemas al hacer uso del LPE Scratch en el contexto del aprendizaje de la programación

básica a nivel bachillerato, concluyendo que este resultado pudo derivarse a condiciones presentadas, como lo fue la no disponibilidad de los estudiantes así como la aplicación de las post pruebas fuera de tiempo, que afectaron negativamente la realización de la réplica en tiempo y forma.

Como se especificó anteriormente, el objetivo general y los objetivos específicos se cumplieron satisfactoriamente dando respuesta a las preguntas de investigación planteadas: el uso de un LPE puede motivar a un estudiante de educación media superior en el aprendizaje de la programación básica, así como aumentar significativamente la adquisición de habilidades de resolución de problemas de los mismos, rechazando las hipótesis nulas para el caso de la prueba piloto, en la cual se tuvo un mayor control del experimento al aplicar los post test en tiempo y forma.

Aportaciones

Las aportaciones de la investigación son las siguientes:

- Uno de los primeros estudios a nivel bachillerato que evalúan la efectividad del LPE Scratch en México.
- Importancia de la identificación de manera temprana de algunas de las variables motivacionales que pueden despertar el interés de los jóvenes en la programación, como lo son la orientación a las metas, la autoeficacia y creencia de control de los estudiantes que están por iniciar una carrera universitaria.
- Validez de un diseño experimental de pre prueba – post prueba para identificar el grado de motivación y la adquisición de habilidades de resolución de problemas al hacer uso de un LPE, además de los métodos estadísticos para el análisis de los datos recabados.
- Instrumento de medición de la motivación adaptado al contexto mexicano y al aprendizaje de la programación básica, así como también el instrumento de medición de habilidades de resolución de problemas haciendo uso de la programación.

Limitaciones de la investigación

La realización del cuasi-experimento tuvo como limitaciones 1) la no aleatorización en la selección de los participantes, 2) la duración del tratamiento, el cual fue de solo cinco

TESIS TESIS TESIS TESIS TESIS

sesiones de una hora cada una, afectando posiblemente los resultados de las pre y post pruebas de motivación así como de aprendizaje de la programación debido al poco tiempo en el que se hizo uso del tratamiento en Scratch y 3) la falta de un grupo de control. Estas limitaciones estuvieron presentes en la prueba piloto y la réplica.

Durante la réplica se presentó un problema mayor de deserción de los participantes (49%) que en el caso de la prueba piloto (29%), debido a que los participantes tuvieron que atender otras actividades en las fechas en las que se llevó a cabo el cuasi-experimento, lo cual provocó un menor control en la realización del mismo generando así los resultados obtenidos durante la réplica.

Recomendaciones para estudios futuros

Para futuras réplicas de experimentos similares se recomienda:

- Contar con un diseño experimental integro con pre prueba – post prueba y grupo de control implementando la aleatorización de la selección de los participantes.
- Contar con mayor tiempo disponible para la realización del experimento, es decir, hacer uso del tratamiento el mayor tiempo posible para que exista una mayor diferencia de tiempo entre las pre pruebas y las post pruebas.
- Llevar a cabo el experimento integrándolo al programa de estudio con la finalidad de disminuir el porcentaje de deserción y el desinterés de los participantes.
- Que las materias en las que se introduzca el tratamiento promuevan habilidades de programación básica.
- Que los profesores que impartan el material, estén familiarizados con el LPE a utilizar, o reciban capacitación previa.

BIBLIOGRAFÍA

Alice.org. (s/f). Recuperado el 4 de mayo de 2016, a partir de <http://www.alice.org/index.php>

An, J., & Park, N. (2011). Computer Application in Elementary Education Bases on Fractal Geometry Theory Using LOGO Programming. En J. J. Park, H. Arabnia, H.-B. Chang, & T. Shon (Eds.), *IT Convergence and Services* (pp. 241–249). Springer Netherlands. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-94-007-2598-0_26

Arévalo Mercado, C. A., Andrade, E. L. M., & Reynoso, J. M. G. (2008). The Effect of Learning Objects on a C++ Programming Lesson. *Proceedings of the 19th Annual International Information Management Association, San Diego, CA, October, 13–15*.

Arévalo Mercado, C. A., & Reynoso, J. M. G. (2010). Un estudio piloto sobre el efecto de los tutores cognitivos para la enseñanza de conceptos básicos de programación. Recuperado el 12 de febrero de 2015, a partir de <http://www.redalyc.org/resumen.oa?id=67413203004>

Arévalo Mercado, C. A., & Solano Romo, L. I. (2013). Patrones de Comportamiento de Estudiantes de Programación al Utilizar una Herramienta de Visualización de Protocolos Verbales. *Conferencias LACLO, 4(1)*. Recuperado a partir de <http://lacro.org/papers/index.php/lacro/article/view/101>

Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *TOJET: The Turkish Online Journal of Educational Technology, 8(1)*. Recuperado a partir de <http://search.proquest.com/openview/0fb4af9542091d6736f909f03e9e2871/1?pq-origsite=gscholar>

Ballester, A. (2002). El aprendizaje significativo en la práctica. *Seminario de Aprendizaje Significativo, 16–21*.

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Educational Technology Research and Development, 59(6)*, 765–782. <http://doi.org/10.1007/s11423-010-9184-z>

Begosso, L. C., & Rodrigues da Silva, P. (2013). Teaching computer programming: A practical review. En *2013 IEEE Frontiers in Education Conference* (pp. 508–510). <http://doi.org/10.1109/FIE.2013.6684875>

Blockly Games. (s/f). Recuperado el 4 de mayo de 2016, a partir de <https://blockly-games.appspot.com/>

Bosch, N., D’Mello, S., & Mills, C. (2013). What Emotions Do Novices Experience during Their First Computer Programming Learning Session? En *Artificial Intelligence in Education* (pp. 11–20). Springer. Recuperado a partir de http://link.springer.com/chapter/10.1007/978-3-642-39112-5_2

Brito, M. A., & Sá-Soares, F. de. (2010). Computer Programming: Fail Fast to Learn Sooner. En M. D. Lytras, P. O. D. Pablos, D. Avison, J. Sipior, Q. Jin, W. Leal, ... D. Horner (Eds.), *Technology Enhanced Learning. Quality of Teaching and Educational Reform* (pp. 223–229). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-13166-0_32

Brusilovsky, P. (1994). Explanatory visualization in an educational programming environment: connecting examples with general knowledge. En *Human-Computer Interaction* (pp. 202–212). Springer. Recuperado a partir de http://link.springer.com/chapter/10.1007/3-540-58648-2_38

Burke, Q., & Kafai, Y. B. (2010). Programming & storytelling: opportunities for learning about coding & composition. En *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 348–351). ACM. Recuperado a partir de <http://dl.acm.org/citation.cfm?id=1810611>

Castillo, A. (2005). *Módulo de autoenseñanza para el examen del sedimento urinario en una plataforma tecnológica*. Universidad de Chile. Recuperado a partir de http://repositorio.uchile.cl/tesis/uchile/2005/castillo_t/sources/castillo_t.pdf

Cearreta-Urbieta, I., & others. (2015). Scratch como recurso didáctico para el desarrollo del Pensamiento Computacional de los alumnos de Secundaria y Bachillerato en la asignatura de Informática y como recurso transversal en el resto de asignaturas. Recuperado a partir de <http://reunir.unir.net/handle/123456789/3150>

Code.org. (2015). Recuperado el 25 de abril de 2016, a partir de <https://code.org/>

Codeweek. (2016). Recuperado el 25 de abril de 2016, a partir de <http://blog.codeweek.eu/?og=1>

codeweek. (2016). Recuperado el 25 de abril de 2016, a partir de <http://codeweek.eu/>

Credé, M., & Phillips, L. A. (2011). A meta-analytic review of the Motivated Strategies for Learning Questionnaire. *Learning and Individual Differences*, 21(4), 337–346.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3), 297–334.

de Kereki, I. F. (2008). Scratch: Applications in Computer Science 1. En *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual* (p. T3B–7–T3B–11). <http://doi.org/10.1109/FIE.2008.4720267>

Dehnadi, S., & Bornat, R. (2006). The camel has two humps (working title). *Middlesex University, UK*. Recuperado a partir de <http://wiki.t-o-f.info/uploads/EDM4600/The%20camel%20has%20two%20humps.pdf>

Doerschuk, P., Liu, J., & Mann, J. (2012). An INSPIRED game programming academy for high school students. En *Frontiers in Education Conference (FIE), 2012* (pp. 1–6). IEEE. Recuperado a partir de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6462240

Duncan, T. G., & McKeachie, W. J. (2005). The Making of the Motivated Strategies for Learning Questionnaire. *Educational Psychologist*, 40(2), 117–128. http://doi.org/10.1207/s15326985ep4002_6

Esquivel Alcocer, L. A., & Roja Cáceres, C. A. (2005). Motivos de Estudiantes de Nuevo Ingreso para Estudiar un Posgrado en Educación. *Revista Iberoamericana de Educación*. Recuperado a partir de <http://www.rieoei.org/investigacion/953Esquivel.PDF>

García, J. C., & Peña, S. (2014). *Actividades del aula con Scratch que favorecen el uso del pensamiento algorítmico. El caso del grado 3° en el INSA*. Universidad ICESI, Cali, Colombia. Recuperado a partir de <http://www.eduteka.org/pdfdir/tesis-juan-carlos-lopez.pdf>

Garneli, B., Giannakos, M. N., Chorianopoulos, K., & Jaccheri, L. (2013). Learning by Playing and Learning by Making. En M. Ma, M. F. Oliveira, S. Petersen, & J. B. Hauge (Eds.), *Serious Games Development and Applications* (pp. 76–85). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-40790-1_8

Giannakos, M. N., Jaccheri, L., & Leftheriotis, I. (2014). Happy Girls Engaging with Technology: Assessing Emotions and Engagement Related to Programming Activities. En P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences* (pp. 398–409). Springer International Publishing. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-319-07482-5_38

González Villarreal, R. J. (2014, junio 11). *EFFECTOS DEL APRENDIZAJE BASADO EN JUEGOS EN EL PENSAMIENTO ALGORÍTMICO DE APRENDICES DE PROGRAMACIÓN*. Universidad Autonoma de Aguascalientes, Aguascalientes, Ags. México.

Harris, A. D., McGregor, J. C., Perencevich, E. N., Furuno, J. P., Zhu, J., Peterson, D. E., & Finkelstein, J. (2006). The Use and Interpretation of Quasi-Experimental Studies in Medical Informatics. *Journal of the American Medical Informatics Association : JAMIA*, 13(1), 16–23. <http://doi.org/10.1197/jamia.M1749>

Hernández, P., & García, L. A. (1991). *Psicología y Enseñanza en el Estudio. Teorías y Técnicas para Potenciar las Habilidades Intelectuales*. Ediciones Piramide.

Høiseth, M., & Jaccheri, L. (2011a). Art and Technology for Young Creators. En J. C. Anacleto, S. Fels, N. Graham, B. Kapralos, M. S. El-Nasr, & K. Stanley (Eds.), *Entertainment Computing – ICEC 2011* (pp. 210–221). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-24500-8_22

Hora de Código - Mexico. (s/f). Recuperado el 4 de mayo de 2016, a partir de <https://hourofcode.com/events/special/mx>

Howey, S. C. (1999). The Relationship between Motivation and Academic Success of Community College Freshmen Orientation Students. Recuperado a partir de <http://eric.ed.gov/?id=ED465391>

Huertas, J. A. (1997). Motivación. *Querer aprender*, 1. Recuperado a partir de http://mateandoconlaciencia.zonalibre.org/TA_Huertas_Unidad_4.pdf

Hui, C., & Tam, F. (2014). Learning Subject Knowledge and Acquiring Twenty-First Century Skills Through Game Development in the Classroom. En Y. Baek, R. Ko, & T. Marsh (Eds.), *Trends and Applications of Serious Gaming and Social Media* (pp. 143–156). Springer Singapore. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-981-4560-26-9_9

Javidi, G., & Sheybani, E. (2009). Digispired: digital inspiration for interactive game design and programming. *Journal of Computing Sciences in Colleges*, 24(3), 144–150.

Jenkins, T. (2002). On the difficulty of learning to program. En *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, pp. 53–58). Recuperado a partir de <http://www.ics.heacademy.ac.uk/Events/conf2002/tjenkins.pdf>

Jonassen, D. D. H. (2012). Problem Typology. En P. D. N. M. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 2683–2686). Springer US. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/referenceworkentry/10.1007/978-1-4419-1428-6_209

Jonassen, D. D. H., & Hung, D. W. (2012). Problem Solving. En P. D. N. M. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 2680–2683). Springer US. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/referenceworkentry/10.1007/978-1-4419-1428-6_208

Kammer, T., Brauner, P., Leonhardt, T., & Schroeder, U. (2011). Simulating LEGO Mindstorms Robots to Facilitate Teaching Computer Programming to School Students. En C. D. Kloos, D. Gillet, R. M. C. García, F. Wild, & M. Wolpers (Eds.), *Towards Ubiquitous Learning* (pp. 196–209). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-23985-4_16

Kelleher, C. (2009). Supporting Storytelling in a Programming Environment for Middle School Children. En I. A. Iurgel, N. Zagalo, & P. Petta (Eds.), *Interactive Storytelling* (pp. 1–4). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-10643-9_1

Kelleher, C., & Pausch, R. (2003). *Lowering the Barriers to Programming: a survey of programming environments and languages for novice programmers*. DTIC Document. Recuperado a partir de <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA457911>

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. En *ACM SIGCSE Bulletin* (Vol. 37, pp. 14–18). ACM. Recuperado a partir de <http://dl.acm.org/citation.cfm?id=1067453>

Ledesma, R., Molina, G., & Valero, P. (2002). Análisis de consistencia interna mediante Alfa de Cronbach: un programa basado en gráficos dinámicos. *Psico-USF*, 7(2), 143–152.

Llera, J. B. (1995). *Psicología de la educación*. Marcombo.

López García, J. C., & others. (2014). Actividades de aula con scratch que favorecen el uso del pensamiento algorítmico. El caso del grado 3° en el INSA. Recuperado a partir de http://bibliotecadigital.icesi.edu.co/biblioteca_digital/handle/10906/76942

Malan, D., & Leitner, H. (2007). Scratch for Budding Computer Scientists.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2014). Educational Games for Teaching Computer Programming. En C. Karagiannidis, P. Politis, & I. Karasavvidis (Eds.), *Research on e-Learning and ICT in Education* (pp. 87–98). Springer New York. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-1-4614-6501-0_7 a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-1-4614-6501-0_7

Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: a sneak preview [education]. En *Second International Conference on Creating, Connecting and Collaborating through Computing, 2004. Proceedings* (pp. 104–109). <http://doi.org/10.1109/C5.2004.1314376>

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16.

Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. *Instructional Science*, 26, 49–63.

McInerney, C. (2010). Having Fun with Computer Programming and Games: Teacher and Student Experiences. En J. Hromkovič, R. Královič, & J. Vahrenhold (Eds.), *Teaching Fundamentals Concepts of Informatics* (pp. 136–142). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-11376-5_13

McKeachie, W. J., & others. (1987). Teaching and Learning in the College Classroom. A Review of the Research Literature (1986) and November 1987 Supplement. Recuperado a partir de <http://eric.ed.gov/?id=ED314999>

Nikou, S. A., & Economides, A. A. (2014). Transition in student motivation during a scratch and an app inventor course. En *2014 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1042–1045). <http://doi.org/10.1109/EDUCON.2014.6826234>

Ogar, O., Shabalina, O., Davtyan, A., & Kizim, A. (2014). Mastering Programming Skills with the Use of Adaptive Learning Games. En A. Kravets, M. Shcherbakov, M. Kultsova, & T. Iijima (Eds.), *Knowledge-Based Software Engineering* (pp. 144–155). Springer International Publishing. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-319-11854-3_14

Oviedo, H. C., & Campo-Arias, A. (2005). Aproximación al uso del coeficiente alfa de Cronbach. *Revista colombiana de psiquiatría*, 34(4), 572–580.

Parham, J., Gugerty, L., & Stevenson, D. E. (2010). Empirical Evidence for the Existence and Uses of Metacognition in Computer Science Problem Solving. En *Proceedings of the 41st ACM technical symposium on Computer science education*. Milwaukee, WI, USA: ACM New York, NY, USA.

Park, C. J., & Hyun, J. S. (2014). Effects of abstract thinking and familiarity with programming languages on computer programming ability in high schools. En *Teaching, Assessment and*

Learning (TALE), 2014 International Conference on (pp. 468–473). IEEE. Recuperado a partir de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7062628

Pereira, M. L. N. (2009). Motivación: perspectivas teóricas y algunas consideraciones de su importancia en el ámbito educativo. *Revista Educación*, 33(2), 153–170.

Pintrich, P. R., & De Groot, E. (1990). Motivational and Self-Regulated Learning Components of Classroom Academic Performance. *Journal of Educational Psychology*, 82, 33–40.

Pintrich, P. R., & others. (1991). A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ). Recuperado a partir de <http://eric.ed.gov/?id=ED338122>

Pintrich, P. R., Smith, D. A. F., García, T., & McKeachie, W. J. (1991). *A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. Ann Arbor: Michigan, National Centre for Research to Improve Postsecondary Teaching and Learning.

Polanco Hernández, A. (2005). La motivación en los estudiantes universitarios. *Revista Electrónica Actualidades Investigativas en Educación*, 5. Recuperado a partir de revista.inie.ucr.ac.cr/index.php/aie/article/download/85/84

Polya, G. (1945). *How to Solve It: A New Aspect of Mathematical Method: A New Aspect of Mathematical Method*. Princeton University Press.

Quevedo Ordoñez, M. V. (2015). Estrategias didácticas basadas en software educativo, para la correcta aplicación de la estructura condicional if else en estudiantes de bachillerato. Recuperado a partir de <http://repositorio.utmachala.edu.ec/handle/48000/4150>

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... others. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60–67.

Rinaudo, M. C., Chiecher, A., & Donolo, D. (2003). Motivación y uso de estrategias en estudiantes universitarios. Su evaluación a partir del Motivated Strategies Learning Questionnaire. *Anales de psicología*, 19(1), 107–119.

Rizvi, M., Humphries, T., Major, D., Jones, M., & Lauzun, H. (2011). A CS0 course using scratch. *Journal of Computing Sciences in Colleges*, 26(3), 19–27.

Roces, C., Tourón, J., & González-Torres, M. C. (1995). Validación preliminar del CEAM II (Cuestionario de estrategias de aprendizaje y motivación II). Recuperado a partir de <http://dadun.unav.edu/handle/10171/19176>

Rodger, S. H., Bashford, M., Dyck, L., Hayes, J., Liang, L., Nelson, D., & Qin, H. (2010). Enhancing K-12 education with alice programming adventures. En *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education* (pp. 234–238). ACM. Recuperado a partir de <http://dl.acm.org/citation.cfm?id=1822156>

Sampieri, R. H., Collado, C. F., Lucio, P. B., & Pérez, M. de la L. C. (1998). *Metodología de la investigación* (Vol. 1). McGraw-Hill. Recuperado a partir de http://www.univo.edu.sv:8081/tesis/021552/021552_Cap3.pdf

Sánchez Abarca, P. S., & Burgos Castillo, E. (2012). Adaptación y validación preliminar del cuestionario de motivación y estrategias de aprendizaje (MSLQ). Recuperado a partir de http://cybertesis.ubiobio.cl/tesis/2012/burgos_e/doc/burgos_e.pdf

Scratch. (2015). Recuperado el 1 de mayo de 2016, a partir de <https://scratch.mit.edu/>

Scratch - Imagine, Program, Share. (s/f). Recuperado el 10 de abril de 2016, a partir de <https://scratch.mit.edu/>

Serafini, G. (2011). Teaching Programming at Primary Schools: Visions, Experiences, and Long-Term Research Prospects. En I. Kalaš & R. T. Mittermeir (Eds.), *Informatics in Schools. Contributing to 21st Century Education* (pp. 143–154). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-24722-4_13

Shannon, L.-J., & Ward, Y. D. (2014). A Case Study: From Game Programming to ICTs. En H. Kaur & X. Tao (Eds.), *ICTs and the Millennium Development Goals* (pp. 237–250). Springer US. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-1-4899-7439-6_14

Shirali, S. A. (2014). George Pólya & problem solving ... An appreciation. *Resonance*, 19(4), 310–322. <http://doi.org/10.1007/s12045-014-0037-7>

Sovic, A., Jagust, T., & Sersic, D. (2014). How to teach basic university-level programming concepts to first graders? En *Integrated STEM Education Conference (ISEC), 2014 IEEE* (pp. 1–6). IEEE. Recuperado a partir de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6891050

Streiner, D. L. (2003a). Being inconsistent about consistency: When coefficient alpha does and doesn't matter. *Journal of personality assessment*, 80(3), 217–222.

Streiner, D. L. (2003b). Starting at the beginning: an introduction to coefficient alpha and internal consistency. *Journal of personality assessment*, 80(1), 99–103.

Suárez, P., Andrés, C., & Sarmiento Porras, R. E. (2015). Estado del arte sobre experiencias de enseñanza de programación a niños y jóvenes para el mejoramiento de las competencias matemáticas en primaria. *Revista mexicana de investigación educativa*, 20(65), 607–641.

Turner, J. L. (2014). *Using Statistics in Small-Scale Language Education Research: Focus on Non-Parametric Data*. Routledge.

Univision. (s/f). Recuperado el 1 de mayo de 2016, a partir de <http://www.univision.com/noticias/educacion/codeorg-cree-que-la-programacion-de-computadoras-debe-ser-materia-esencial>

Wang, L.-C., & Chen, M.-P. (2010). Learning Programming Concepts through Game Design: A PCT Perspective. En *2010 Third IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL)* (pp. 219–221). <http://doi.org/10.1109/DIGITEL.2010.47>

Wang, L.-C., & Chen, M.-P. (2012). Effects of the Sequence of Game-Play and Game-Design on Novices' Motivation, Flow, and Performance. En Z. Pan, A. D. Cheok, W. Müller, M. Chang, & M. Zhang (Eds.), *Transactions on Edutainment VIII* (pp. 46–55). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com.dibpxy.uaa.mx/chapter/10.1007/978-3-642-31439-1_5

Wang, T., Mei, W.-H., Lin, S.-L., Chiu, S.-K., & Lin, J. M.-C. (2009). Teaching programming concepts to high school students with alice. En *Frontiers in Education Conference, 2009*.

FIE'09. 39th IEEE (pp. 1–6). IEEE. Recuperado a partir de http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5350486

Wang, X., & Zhou, Z. (2011). The research of situational teaching mode of programming in high school with Scratch. En *Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International* (Vol. 2, pp. 488–492). <http://doi.org/10.1109/ITAIC.2011.6030380>

Wolz, U., Leitner, H. H., Malan, D. J., & Maloney, J. (2009). Starting with scratch in CS 1. En *ACM SIGCSE Bulletin* (Vol. 41, pp. 2–3). ACM. Recuperado a partir de <http://dl.acm.org/citation.cfm?id=1508869>

Yeager, K. (s/f). LibGuides: SPSS Tutorials: Paired Samples t Test. Recuperado el 13 de abril de 2016, a partir de <http://libguides.library.kent.edu/SPSS/PairedSamplestTest>

Yoo, S., Kim, K.-A., Kim, Y., Yeum, Y., Kanemune, S., & Lee, W. (2006). Empirical study of educational programming language for k12: Between dolittle and visual basic. *IJCSNS*, 6(6), 119.

Yoon, I., Kim, J., & Lee, W. (2016). The analysis and application of an educational programming language (RUR-PLE) for a pre-introductory computer science course. *Cluster Computing*, 1–18.

ANEXOS

ANEXO A: Tratamiento en Scratch

ANEXO B: Información demográfica

ANEXO C: Pre prueba MSLQ

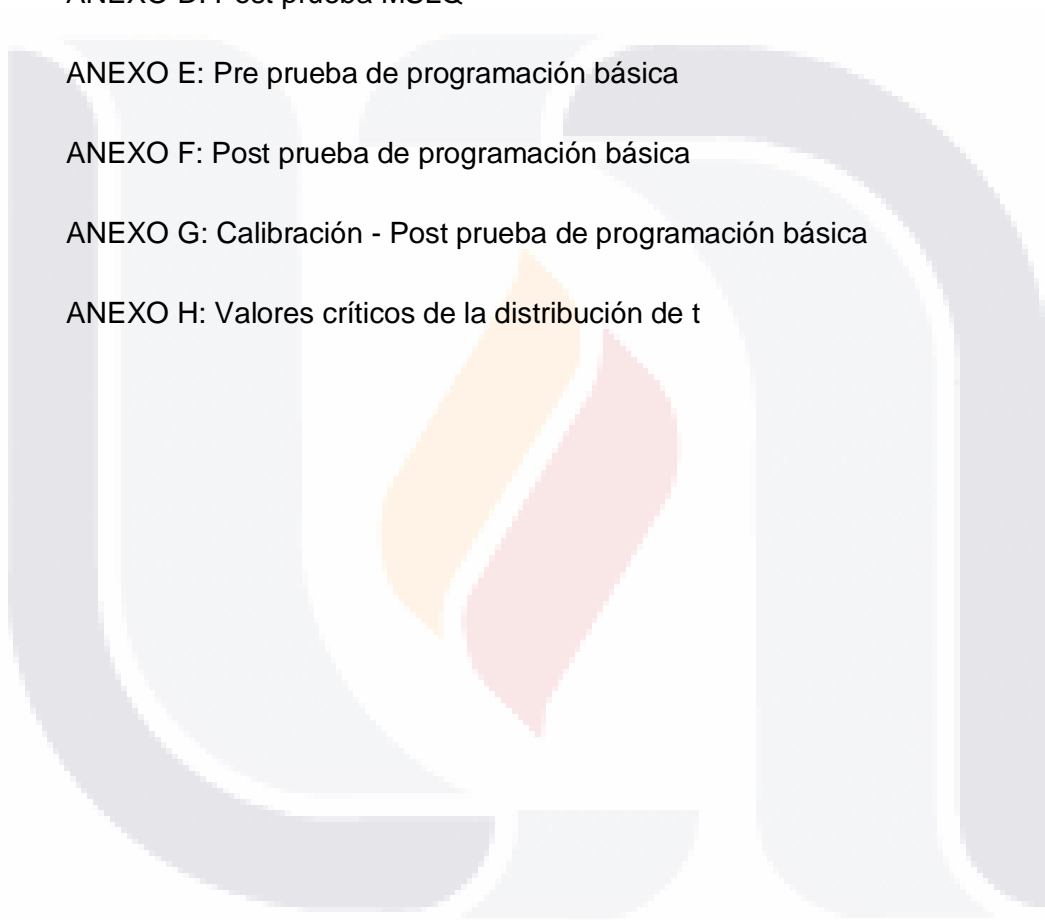
ANEXO D: Post prueba MSLQ

ANEXO E: Pre prueba de programación básica

ANEXO F: Post prueba de programación básica

ANEXO G: Calibración - Post prueba de programación básica

ANEXO H: Valores críticos de la distribución de t



ANEXO A: Tratamiento en Scratch

UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES CONACYT

por siempre
IMAGINAR
CREAR
JUGAR
COMPARTIR
REFLEXIONAR

SCRATCH
Aprendiendo a programar jugando

Sesión 1
Conociendo Scratch
Secuencia de Instrucciones

Autor: L. Néstor Yashira García Ortega
Asesor: Dr. Carlos Argüello Arévalo Mercado
Marzo de 2016, Aguascalientes, Ags.

Contenido

- Resolución de problemas
- Pasos en la resolución de problemas
- Introducción a los algoritmos
- ¿Qué es Scratch?
- Interfaz de Scratch
- Conociendo los conjuntos de instrucciones
- Primeros pasos en Scratch
 - Mi primer programa en Scratch (Mostrar mensaje)
 - Mensajes por lapso de tiempo
 - Cambiar apariencia de los objetos
 - Agregar sonido a los objetos
 - Agregar movimiento a los objetos
 - Agregar fondo de escenario y más objetos e interacción entre ellos

Pasos en la Resolución de Problemas

Una de las metodologías más utilizadas en la resolución de problemas es la formulada por Polya, la cual consta de cuatro etapas que coinciden en la elaboración de programas de computadoras.

Problema

Entender el problema

Elaborar un Plan

Ejecutar el Plan

Revisar

Creación del Algoritmo

Pasos para resolver un problema

Problema

Entender el Problema

Revisar

Elaborar un plan (Algoritmo)

Ejecutar el plan

Hacer que el "gato" de Scratch muestre el mensaje: Bienvenidos al mundo de Scratch!

¿Cuál es el problema? (Entendiendo lo que debo resolver?)

Ejemplo Ejecución en Scratch:

1. Iniciar el programa
2. Mostrar mensaje "Bienvenidos al mundo de Scratch!"
3. Terminar el programa

¿Qué es Scratch?

Es un **lenguaje de programación educativo** que te permite crear tus propias historias interactivas, animaciones, juegos, música y arte.

Esta constituido por símbolos iconográficos denominados **"bloques"**. Hace que la programación sea más atractiva y accesible.

Prácticamente todo se lleva a cabo arrastrando y soltando elementos con el ratón, y modificando con el teclado únicamente valores numéricos, textos.

Ejercicio 1 Mostrar Mensajes

Ejercicio 1: Hacer que el "gato" muestre el siguiente mensaje:

- Bienvenido al curso de Scratch!

Elaborar el plan (Algoritmo)

1. Iniciar el programa
2. Mostrar mensaje "Bienvenido al curso de Scratch!"
3. Terminar el programa

Primeros pasos en Scratch

Curso de Programación

SCRATCH

Aprendiendo a programar jugando

Sesión 2 Parte I

Variables y Operadores en Scratch

Autor: L. Neira Yadhira García Ortega
Asesor: Dr. Carlos Aguilar Arriola Mercado
Marzo de 2016, Aguascalientes, Ags.

Contenido

- Variables en Scratch
 - Variables simples
 - Variables como respuesta a preguntas
 - Variables numéricas
- Números aleatorios en Scratch

Variables

Las **variables** son herramientas que nos permiten recordar datos, en otras palabras, son un espacio en memoria que nos permiten almacenar información para posteriormente utilizarla dentro de un programa.

Se encuentran dentro del conjunto **Variables**

Creación de variables

Para crear variables en Scratch:

- Se selecciona el objeto al que pertenece dicha variable/información **Variables**
- Se selecciona el conjunto de instrucciones **Nueva variable**
- Se le da nombre a la variable y clic en **Aceptar**

Mostrar valor de variable en un mensaje

Para ello, haremos uso del bloque **unir: hola mundo** que se encuentra dentro del conjunto **Operadores**

Y se utiliza dentro del bloque **decir: Hola por 2 segundos** del conjunto **Apariencia**

Ejercicio 8

Mostrar valor de variable en un mensaje

Ejercicio 8: Guarda el Ejercicio 7 como Ejercicio 8. Haz las siguientes modificaciones para que el objeto 1 (gato) muestre el valor de las variables en un mensaje:

- Después de fijar el valor a las variables, utiliza el bloque **unir: hola mundo** para que el gato muestre su valor, "concatenando", es decir uniendo cadenas de texto

Elaborar el plan (Algoritmo)

1. Iniciar el programa
2. Crear variable nombre
3. Crear variable edad
4. Fijar valor de variable nombre
5. Fijar valor de variable edad
6. Decir "hola nombre" por dos segundos
7. Decir "tienes edad" por dos segundos
8. Terminar el programa

Ejercicio 8

Mostrar valor de variable en un mensaje

Ahora modifica tu algoritmo y tus bloques para que el resultado sea el siguiente:

Elaborar el plan (Algoritmo)

6. Decir "hola nombre" por dos segundos
7. Decir "tienes edad años" por dos segundos

Variable como respuesta a una pregunta

Este tipo de variable, toma el valor, que se da como **respuesta** a una pregunta, el cual es introducido **mediante el teclado**.

Dicha respuesta puede ser **texto** o **valores numéricos**.

Curso de Programación

SCRATCH
Aprendiendo a programar jugando

Sesión 2 Parte II
Operadores en Scratch

Autor: L.1 Nelia Yashira García Ortega
Asesor: Dr. Carlos Angelito Arribas Mercado
Octubre de 2015, Aguascalientes, Ags.

Contenido

- Operadores aritméticos en Scratch
- Operadores de comparación en Scratch

Operadores aritméticos

Los *operadores aritméticos* son los que se utilizan para realizar cálculos matemáticos.

Se encuentran dentro del conjunto **Operadores**

Operador	Descripción
$+$	Calcula la suma de dos números
$-$	Calcula la resta de dos números
$*$	Calcula la multiplicación de dos números
$/$	Calcula la división entre dos números

Ejercicio 12 Realizar la suma de dos números

Ejercicio 12: Crea un nuevo proyecto y guárdalo como **Ejercicio 12**.

Realiza un programa que pida dos números y obtenga la suma de ellos. Para ello:

- Crea dos variables para pedir los números, *numero 1* y *numero 2*
- Crea una variable llamada *suma* y en ella guarda el **resultado** de la suma de los dos números
- Haz que el personaje muestre el resultado.

Ejercicio 12 Realizar la suma de dos números

Elaborar el plan (Algoritmo)

1. Iniciar el programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Crear variable suma
5. Preguntar valor de numero 1
6. Fijar valor de numero 1
7. Preguntar valor de numero 2
8. Fijar valor de numero 2
9. Fijar valor suma = numero 1 + numero 2
10. Decir numero 1 + numero 2 = suma
11. Terminar el programa

Ejercicio 12 Realizar la suma de dos números

Ejercicio 13 Realizar la suma, resta, multiplicación y división de dos números

Ejercicio 13: Guarda el Ejercicio 12 como **Ejercicio 13**.

Ahora, antes de terminar el programa, agrega las demás operaciones, es decir la resta, multiplicación y división. El personaje deberá mostrar cada uno de los resultados.

Nota: crea una variable más por cada operación, para almacenar en ellas cada uno de los resultados (resta, multiplicación, división).

Ejercicio 14 Agregar interacción, al Ejercicio 13, entre dos personajes

Ejercicio 14: Guarda el Ejercicio 13 como **Ejercicio 14**.

Ahora, cambia el fondo del escenario y agrega otro personaje.

Haz las **modificaciones** correspondientes para que el *personaje 1* pregunte al *personaje 2* los dos números, los cuales deben ser generados aleatoriamente, y realiza la suma de estos.

Tip: Bástate en el **Ejercicio 6_1** para determinar los turnos de los personajes.




Curso de Programación



SOBRATOH

Aprendiendo a programar jugando

Sesión 3

Condicionales en Scratch

Autor: L. I. Neira Yuliana García Ortega
Asesor: Dr. Carlos Angélio Arévalo Mercado
Octubre de 2015, Aguascalientes, Ags.



Contenido

- **Condicionales en Scratch**
 - Condicional simple (Si)
 - Condicional doble (Si ... Si no)

Condicionales en Scratch

Los **condicionales** son aquellos bloques que evalúan una **condición lógica**, es decir, son preguntas con respuestas de la forma "sí" o "no".


Permiten ejecutar una instrucción o conjunto de instrucciones, siempre y cuando se **cumpla con la condición** dada, si no se cumple, las instrucciones son ignoradas.

Se encuentran dentro del conjunto Control

Condicional simple

En caso del **condicional simple**, si la respuesta es "sí", es decir, si se cumple con la condición especificada, se realiza el conjunto de instrucciones dentro del bloque y sigue con la secuencia normal del programa.

De lo contrario, si la respuesta es "no", es decir, si la condición no se cumple, se ignoran las instrucciones dentro del bloque, siguiendo con la secuencia del programa.



Ejercicio 15 condicional simple

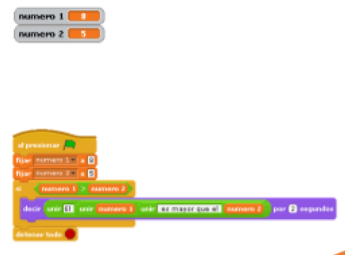
Ejercicio 15: Genera un nuevo proyecto y guárdalo con el nombre **Ejercicio 15**.

Realizar un programa que mediante el uso del **condicional simple** determine si el número 1 es mayor que el número 2.

Si el **numero 1** es mayor que el **numero 2**, el personaje debe mostrar el mensaje "el número 1 es mayor que el número 2", de lo contrario termina el programa.

Elaborar el plan (Algoritmo)

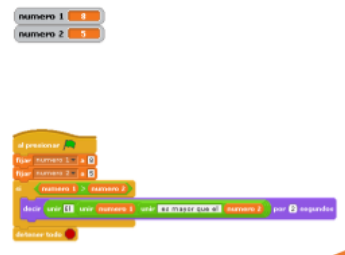
1. Iniciar el programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Fijar valor de variable numero 1
5. Fijar valor de variable numero 2
6. Si numero 1 mayor que numero 2 (numero 1 > numero 2)
 - 6.1. Decir: el número 1 es mayor que el número 2
7. Fin del condicional "si"
8. Terminar el programa



Ejercicio 15 condicional simple

Si numero 1 > numero 2

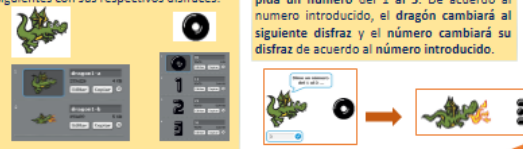
El 1 es mayor que el 2



Ejercicio 17 condicional simple --- agregando animación a personajes y objetos

Ejercicio 17:

1. Genera un nuevo proyecto y guárdalo con el nombre **Ejercicio 17**.
2. Borra todos los objetos y agrega los siguientes con sus respectivos disfraces:
3. Realiza un programa, en el cual el dragón pida un número del 1 al 3. De acuerdo al número introducido, el dragón cambiará al siguiente disfraz y el número cambiará su disfraz de acuerdo al número introducido.



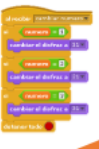
Ejercicio 17 condicional simple --- agregando animación a personajes y objetos

Elaborar el plan (Algoritmo)

1. Iniciar el programa
2. Crear variable numero
3. Dragón pregunta valor de numero
4. Introducir valor de numero
5. Fijar valor de numero
6. Cambiar disfraz al dragón
7. Dragón envía mensaje "Cambiar numero"

Elaborar el plan (Algoritmo)

1. Al recibir "Cambiar numero"
2. Si numero = 1
 - 6.1. Cambiar al disfraz a 1
7. Fin del condicional "si numero = 1"
8. Si numero = 2
 - 8.1. Cambiar al disfraz a 2
9. Fin del condicional "si numero = 2"
10. Si numero = 3
 - 11.1. Cambiar al disfraz a 3
9. Fin del condicional "si numero = 3"
7. Terminar el programa





Curso de Programación

SCRATCH
Aprendiendo a programar jugando

Sesión 4
Ciclos en Scratch

Autor: L. I. Neira Yashira García Ortega
Asesor: Dr. Carlos Angello Arévalo Mercado
Octubre de 2015, Aguascalientes, Ags.

Contenido

- Ciclos en Scratch
 - Por siempre
 - Repetir por...


Ciclos en Scratch

Los **ciclos** son bloques que repiten un conjunto de bloques o instrucciones dentro de ellos, hasta que la condición especificada deje de cumplirse.

Se utilizan para repetir una acción sin tener que escribir varias veces el mismo código, ahorrando tiempo.

Se encuentran dentro del conjunto **Control**

Ciclo "por siempre..."



Ejecuta continuamente el conjunto de bloques dentro de él, es decir, los repite.


Ejercicio 21 Ciclo "por siempre"

Ejercicio 21: Genera un nuevo proyecto y guárdalo con el nombre **Ejercicio 21**.

Realizar un programa, en el cual, un personaje se mueva 10 pasos, por siempre, dentro del escenario, además cambie de disfraz.

Elaborar el plan (Algoritmo)

1. Iniciar el programa
2. Por siempre
 - 2.1 Cambiar disfraz (1)
 - 2.2 Mover 10 pasos
 - 2.3 Cambiar disfraz (2)
3. Fin del ciclo "por siempre"
4. Terminar el programa




Ejercicio 22 Ciclo "por siempre"

Ejercicio 22: Guarda el **Ejercicio 21** con el nombre **Ejercicio 22**.


Modifica tu algoritmo y tus bloques para que tu personaje al tocar un borde del escenario, rebote.

Elaborar el plan (Algoritmo)

1. Iniciar el programa
2. Por siempre
 - 2.1 Rebotar si está tocando un borde
 - 2.2 Siguiente disfraz
 - 2.3 Mover 10 pasos
3. Fin del ciclo "por siempre"



Ejercicio 22 Ciclo "por siempre"



Ciclo "repetir por ..."



Ejecuta, un número específico de veces, los bloques dentro de él, es decir, los repite.

ANEXO B: Información demográfica

Información personal	
Esta sección consiste en preguntas básicas de acuerdo a tu nivel y plantel educativo así como tu experiencia en programación.	
* ID de Alumno	
<input type="text"/>	
<i>Solo números deben ser ingresados en esta campo.</i>	
* Edad	
<input type="text"/>	
<i>Solo números deben ser ingresados en esta campo.</i>	
* Sexo	
<input type="radio"/> Mujer <input type="radio"/> Hombre	
* Grupo Elija una de las siguientes opciones.	
<input type="text" value="Elige una respuesta"/>	
* Semestre Elija una de las siguientes opciones.	
<input type="text" value="Elige una respuesta"/>	
* Plantel Educativo Elija una de las siguientes opciones.	
<input type="text" value="Elige una respuesta"/>	
* ¿Has cursado materias de programación anteriormente?	
<input type="radio"/> Sí <input type="radio"/> No	

ANEXO C: Pre prueba MSLQ

Inicio de la pre prueba MSLQ en la herramienta LimeSurvey

Pre MSLQ Bachuaa

El objetivo de este cuestionario es ayudarte a conocer la motivación que tienes al iniciar el **APRENDIZAJE DE CONCEPTOS BÁSICOS DE PROGRAMACIÓN** con la finalidad de que desarrolles habilidades de resolución de problemas mediante la elaboración de algoritmos para su resolución.

Los resultados permitirán que las autoridades de tu facultad, profesores y tutores diseñen estrategias educativas para apoyarte durante tu trayectoria académica. **Recuerda que no hay respuestas correctas o incorrectas, solo responde tan precisamente como puedas** de manera que tu respuesta refleje tu situación. Te pedimos que respondas con honestidad.

0% 100%

Preguntas sobre Motivación

Las siguientes preguntas indagan respecto a la motivación y actitudes

Escoge una de las respuestas de abajo para contestar las preguntas. Si piensas que un enunciado te describe totalmente, marca el 7; si el enunciado no te describe en absoluto, marca el 1. Si el enunciado te describe mas o menos, escoge un número entre el 1 y el 7 que mejor te describa.

• INSTRUCCIONES: Lee cuidadosamente cada una de las afirmaciones y usa la escala, que se encuentra debajo de este párrafo para responder a la preguntas. Selecciona la opción que corresponda a tu respuesta que puede ir desde No tiene en absoluto nada que ver conmigo (1) hasta Me describe totalmente (7).

Esta pregunta es de respuesta obligatoria. Por favor complete todas las partes.

	(1) No tiene en absoluto nada que ver conmigo	(2) Me describe muy poco	(3) Me describe moderadamente	(4) No estoy seguro(a)	(5) Me describe suficientemente	(6) Me describe mucho	(7) Me describe totalmente
Para el aprendizaje de la programación básica, prefiero material didáctico que sea un reto para mí, para aprender cosas nuevas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Cada una de las preguntas del cuestionario tiene las mismas opciones de respuesta. En la siguiente tabla se especifica cada una de las preguntas de motivación.

No.	Pregunta
1	Para el aprendizaje de la programación básica, prefiero material didáctico que sea un reto para mí, para aprender cosas nuevas
2	Si estudio de manera adecuada, seré capaz de aprender programación básica
3	Cuando presento un examen de programación básica, pienso que mi desempeño es deficiente comparado con el de mis compañeros
4	Pienso que seré capaz de usar lo que aprenda en programación básica en otras materias
5	Creo que recibiré una excelente calificación en programación básica
6	Estoy seguro que puedo entender las lecturas más difíciles de programación básica
7	Obtener una buena calificación en programación básica es lo más satisfactorio para mí ahora
8	Cuando presento un examen de programación básica pienso en las preguntas que no he podido contestar
9	Es mi culpa si no aprendo el material o contenido de programación básica

-
- 10 Es importante para mí aprender programación básica
 - 11 Lo más importante para mi ahora es mejorar mi promedio general, por lo que mi principal interés en programación básica es obtener una buena calificación
 - 12 Confío en que puedo aprender los conceptos básicos enseñados en programación básica
 - 13 Si me lo propongo, puedo conseguir las mejores calificaciones en programación básica y superar a mis compañeros
 - 14 Cuando presento un examen de programación básica pienso en las consecuencias de un fracaso
 - 15 Confío en que puedo entender el material más complejo, de programación básica, presentado por el profesor
 - 16 En el aprendizaje de la programación básica, prefiero que el material o contenido aliente mi curiosidad, aun si es difícil de aprender
 - 17 Estoy muy interesado en el contenido de programación básica
 - 18 Si me esfuerzo lo suficiente, entenderé el contenido de programación básica
 - 19 Cuando presento un examen de programación básica, experimento una sensación desagradable como de "angustia"
 - 20 Confío en que puedo hacer un excelente trabajo respecto a las tareas y exámenes de programación básica
 - 21 Espero que mi desempeño en el aprendizaje de la programación básica sea bueno
 - 22 Lo más satisfactorio para mí en programación básica es tratar de entender el contenido tan completamente como sea posible
 - 23 Considero útil el contenido de esta clase de programación básica porque me permite aprender
 - 24 Cuando tengo la oportunidad escojo las actividades de programación básica en las cuales pueda aprender, a pesar de no estar seguro si obtendré una buena calificación
 - 25 Si no entiendo el contenido de programación básica es porque no me esfuerzo lo necesario
 - 26 Me gusta la programación básica
 - 27 Entender programación básica es muy importante para mí
 - 28 Cuando presento un examen de programación básica, siento que mi corazón se acelera
 - 29 Estoy seguro de que puedo dominar las habilidades que se enseñan en programación básica
 - 30 Quiero desempeñarme bien en programación básica porque es importante para mi demostrar mi habilidad a mi familia, amigos, maestros u otros
 - 31 Considerando al profesor, mis habilidades y la dificultad de la programación básica; pienso que saldré bien en el resultado final
-

ANEXO D: Post prueba MSLQ

Inicio de la post prueba MSLQ en la herramienta LimeSurvey

uando el constructi...

Post MSLQ Bachuaa

El objetivo de este cuestionario es ayudarte a conocer la motivación que tienes **AL TERMINAR EL CURSO DE PROGRAMACIÓN BÁSICA EN SCRATCH** el cual tenía la finalidad de que desarrolles habilidades de resolución de problemas mediante la elaboración de algoritmos y programas para su resolución.

Los resultados permitirán que las autoridades de tu facultad, profesores y tutores diseñen estrategias educativas para apoyarte durante tu trayectoria académica. **Recuerda que no hay respuestas correctas o incorrectas, solo responde tan precisamente como puedas** de manera que tu respuesta refleje tu situación. Te pedimos que respondas con honestidad.

0% 100%

Preguntas sobre Motivación

Las siguientes preguntas indagan respecto a la motivación y actitudes de estudio para este curso.

Escoge una de las respuestas de abajo para contestar las preguntas. Si piensas que un enunciado te describe totalmente, marca el 7; si el enunciado no te describe en absoluto, marca el 1. Si el enunciado te describe mas o menos, escoge un número entre el 1 y el 7 que mejor te describa.

• INSTRUCCIONES: Lee cuidadosamente cada una de las afirmaciones y usa la escala, que se encuentra debajo de este párrafo para responder a la preguntas. Selecciona la opción que corresponda a tu respuesta que puede ir desde No tiene en absoluto nada que ver conmigo (1) hasta Me describe totalmente (7).

Recuerda que este cuestionario solo hace referencia al curso impartido en Scratch.

Esta pregunta es de respuesta obligatoria. Por favor complete todas las partes.

	(1) No tiene en absoluto nada que ver conmigo	(2) Me describe muy poco	(3) Me describe moderadamente	(4) No estoy seguro(a)	(5) Me describe suficientemente	(6) Me describe mucho	(7) Me describe totalmente
En un curso como este, prefiero material didáctico que sea un reto para mí, para aprender cosas nuevas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Cada una de las preguntas del cuestionario tiene las mismas opciones de respuesta. En la siguiente tabla se especifica cada una de las preguntas de motivación.

No.	Pregunta
1	En un curso como este, prefiero material didáctico que sea un reto para mí, para aprender cosas nuevas
2	Si estudio de manera adecuada, seré capaz de aprender el material de este curso
3	Cuando presento un examen en este curso, pienso que mi desempeño es deficiente comparado con el de mis compañeros
4	Pienso que seré capaz de usar lo que aprenda en este curso en otras materias
5	Creo que recibiré una excelente calificación en este curso
6	Estoy seguro que puedo entender las lecturas más difíciles de este curso
7	Obtener una buena calificación en este curso es lo más satisfactorio para mí ahora
8	Cuando presento un examen en este curso, pienso en las preguntas que no he podido contestar

-
- 9 Es mi culpa si no aprendo el material o contenido de este curso
 - 10 Es importante para mí aprender el material de este curso
 - 11 Lo más importante para mí ahora es mejorar mi promedio general, por lo que mi principal interés en este curso es conseguir una buena calificación
 - 12 Confío en que puedo aprender los conceptos básicos enseñados en este curso
 - 13 Si me lo propongo, puedo conseguir las mejores calificaciones en este curso y superar a mis compañeros
 - 14 Cuando presento un examen en este curso pienso en las consecuencias de un fracaso
 - 15 Confío en que puedo entender el material más complejo de este curso presentado por el profesor
 - 16 En un curso como este, prefiero que el material o contenido aliente mi curiosidad, aun si es difícil de aprender
 - 17 Estoy muy interesado en el contenido de este curso
 - 18 Si me esfuerzo lo suficiente, entenderé el contenido del curso
 - 19 Cuando presento un examen este curso, experimento una sensación desagradable como de "angustia"
 - 20 Confío en que puedo hacer un excelente trabajo respecto a las tareas y exámenes en este curso
 - 21 Espero que mi desempeño en este curso sea bueno
 - 22 Lo más satisfactorio para mí en este curso es tratar de entender el contenido tan completamente como sea posible
 - 23 Considero útil el contenido de este curso porque me permite aprender
 - 24 Cuando tengo la oportunidad escojo las actividades de este curso en las cuales pueda aprender, aun si ello no me garantiza una buena calificación
 - 25 Si no entiendo el contenido del curso es porque no me esfuerzo lo necesario
 - 26 Me gusta este curso
 - 27 Entender este curso es muy importante para mí
 - 28 Cuando presento un examen en este curso, siento que mi corazón se acelera
 - 29 Estoy seguro de que puedo dominar las habilidades que se enseñan en este curso
 - 30 Quiero desempeñarme bien en este curso porque es importante para mi demostrar mi habilidad a mi familia, amigos, maestros u otros
 - 31 Considerando al profesor, mis habilidades y la dificultad de este curso; pienso que saldré bien en el resultado final
-

ANEXO E: Pre prueba de programación básica

Instrucciones: Lee cuidadosamente el ejercicio, identificando qué pasos debes seguir para llegar a la solución. Lee cada uno de los algoritmos y selecciona el correcto, es decir, el que llega a la solución del ejercicio.

Ejercicio 1: Número mayor.

Algoritmo en el que se:

- Pida 2 números enteros, los cuales deben ser introducidos mediante el teclado.
- Determine cuál número es mayor.
- Muestre en un mensaje el número mayor.



Algoritmo 1:

1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Preguntar **numero 1**
5. Introducir respuesta
6. Fijar **numero 1** a respuesta
7. Preguntar **numero 2**
8. Introducir respuesta
9. Fijar **numero 2** a respuesta
10. Si **numero 2** es mayor que **numero 1**:
 - 10.1. Decir "el número mayor es **numero 1**"
11. Si no
 - 11.1. Decir "el número mayor es **numero 2**"
12. Fin del programa

Algoritmo 2:

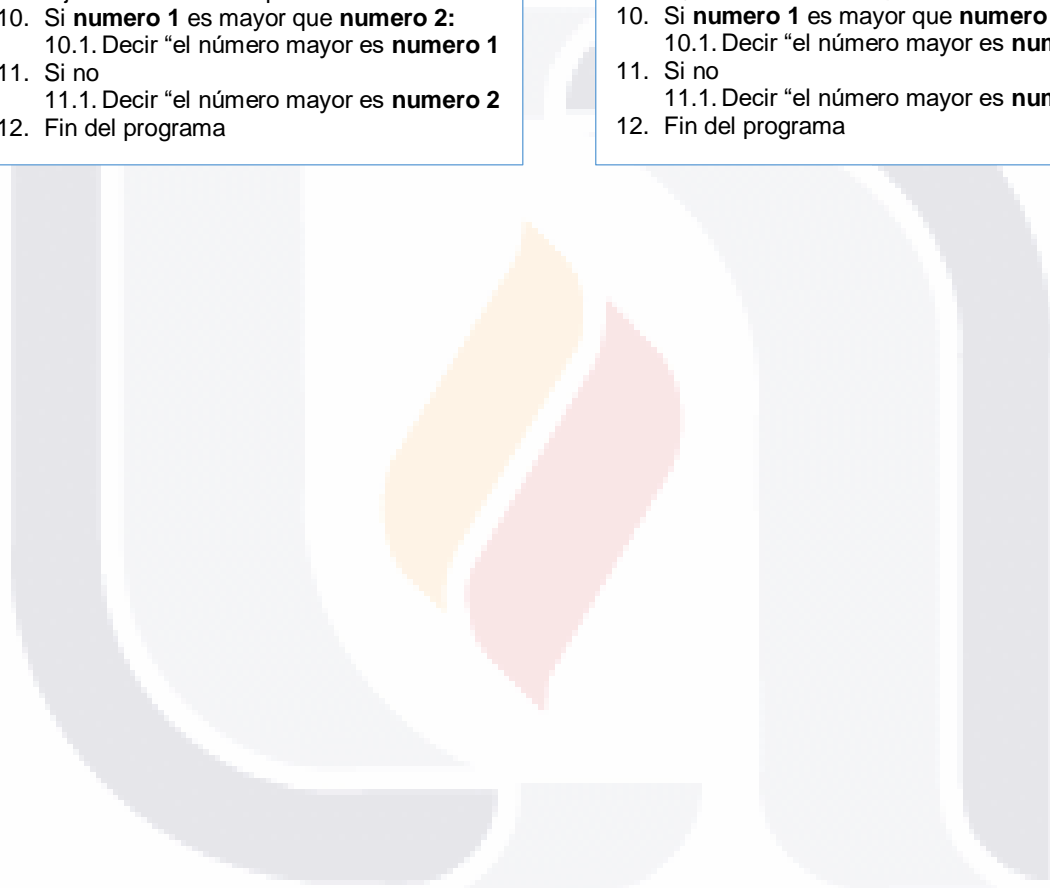
1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Decir "hola"
5. Preguntar **numero 1**
6. Introducir respuesta
7. Fijar **numero 1** a respuesta
8. Preguntar **numero 2**
9. Introducir respuesta
10. Si **numero 1** es mayor que **numero 2**:
 - 10.1. Decir "el número mayor es **numero 1**"
11. Si no
 - 11.1. Decir "el número mayor es **numero 2**"
12. Fin del programa

Algoritmo 3:

1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Decir "hola"
5. Preguntar **numero 1**
6. Introducir respuesta
7. Fijar **numero 1** a respuesta
8. Preguntar **numero 2**
9. Fijar **numero 2** a respuesta
10. Si **numero 1** es mayor que **numero 2**:
 - 10.1. Decir "el número mayor es **numero 1**"
11. Si no
 - 11.1. Decir "el número mayor es **numero 2**"
12. Fin del programa

Algoritmo 4:

1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Preguntar **numero 1**
5. Introducir respuesta
6. Fijar **numero 1** a respuesta
7. Preguntar **numero 2**
8. Introducir respuesta
9. Fijar **numero 2** a respuesta
10. Si **numero 1** es mayor que **numero 2**:
 - 10.1. Decir "el número mayor es **numero 1**"
11. Si no
 - 11.1. Decir "el número mayor es **numero 2**"
12. Fin del programa



Ejercicio 2: Promedio de 3 números.

Algoritmo en el que se:

- Pida 3 números, los cuales deben ser introducidos mediante el teclado.
- Calcule el promedio de los 3 números.
- Muestre el promedio, así como si el promedio es mayor o menor que 10.



Algoritmo 1:

1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Crear variable numero 3
5. Crear variable promedio
6. Preguntar **numero 1**
7. Introducir respuesta
8. Fijar **numero 1** a respuesta
9. Preguntar **numero 2**
10. Introducir respuesta
11. Fijar **numero 2** a respuesta
12. Preguntar **numero 3**
13. Introducir respuesta
14. Fijar **numero 3** a respuesta
15. Fijar **promedio** a **(numero 1 + numero 2 + numero 3)**
16. Decir "el promedio es **promedio**"
17. Si **promedio** es menor que **10**:
 - 17.1. Decir "el **promedio** es menor que 10"
18. Si no
 - 18.1. Decir "el **promedio** es mayor que 10"
19. Fin del programa

Algoritmo 2:

1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Crear variable numero 3
5. Crear variable promedio
6. Preguntar **numero 1**
7. Introducir respuesta
8. Fijar **numero 1** a respuesta
9. Preguntar **numero 2**
10. Introducir respuesta
11. Fijar **numero 2** a respuesta
12. Preguntar **numero 3**
13. Introducir respuesta
14. Fijar **numero 3** a respuesta
15. Fijar **promedio** a **(numero 1 + numero 2 + numero 3) / 3**
16. Decir "el promedio es **promedio**"
17. Si **promedio** es menor que **10**:
 - 17.1. Decir "el **promedio** es menor que 10"
18. Si no
 - 18.1. Decir "el **promedio** es mayor que 10"
19. Fin del programa

Algoritmo 3:

1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Crear variable numero 3
5. Preguntar **numero 1**
6. Introducir respuesta
7. Fijar **numero 1** a respuesta
8. Preguntar **numero 2**
9. Introducir respuesta
10. Fijar **numero 2** a respuesta
11. Preguntar **numero 3**
12. Introducir respuesta
13. Fijar **numero 3** a respuesta
14. Preguntar **promedio**
15. Introducir respuesta
16. Fijar **promedio** a respuesta
17. Fijar **promedio** a $(\text{numero 1} + \text{numero 2} + \text{numero 3}) / 3$
18. Decir "el **promedio** es **promedio** menor que 10"
19. Fin del programa

Algoritmo 4:

1. Iniciar programa
2. Crear variable numero 1
3. Crear variable numero 2
4. Crear variable numero 3
5. Preguntar **numero 1**
6. Introducir respuesta
7. Fijar **numero 1** a respuesta
8. Preguntar **numero 2**
9. Introducir respuesta
10. Fijar **numero 2** a respuesta
11. Preguntar **numero 3**
12. Introducir respuesta
13. Fijar **numero 3** a respuesta
14. Preguntar **promedio**
15. Introducir respuesta
16. Fijar **promedio** a respuesta
17. Fijar **promedio** a $(\text{numero 1} + \text{numero 2} + \text{numero 3})$
18. Decir "el **promedio** es **promedio** mayor que 10"
19. Fin del programa

Ejercicio 3: Números impares

Algoritmo en el que:

- Los números impares del 1 al 5 se muestren individualmente, es decir, el número debe cambiar de disfraz según el valor del número impar.



Algoritmo 1:

1. Iniciar programa
2. Crear variable numero
3. Fijar **numero** a 1
4. Repetir por 5 veces
 - 4.1. Si **numero** es impar
 - 4.1.1. Cambiar el disfraz a **numero**
 - 4.1.2. **Esperar 1 segundo**
 - 4.2. Incrementar valor de **numero** en 1 (numero + 1)
5. Fin ciclo repetir
6. Fin del programa

Algoritmo 2:

1. Iniciar programa
2. Crear variable numero
3. Fijar **numero** a 5
4. Incrementar valor de **numero** en 1 (numero + 1)
5. Repetir por 3 veces
 - 5.1. Si **numero** es impar
 - 5.1.1. Cambiar el disfraz a **numero**
 - 5.1.2. Esperar 1 segundo
6. Fin ciclo repetir
7. Fin del programa

Algoritmo 3:

1. Iniciar programa
2. Crear variable numero
3. Fijar **numero** a 1
4. Repetir por 3 veces
5. Si **numero** es impar
 - 5.1. Cambiar el disfraz a **numero**
 - 5.2. Esperar 1 segundo
6. Fin ciclo repetir
7. Fin del programa

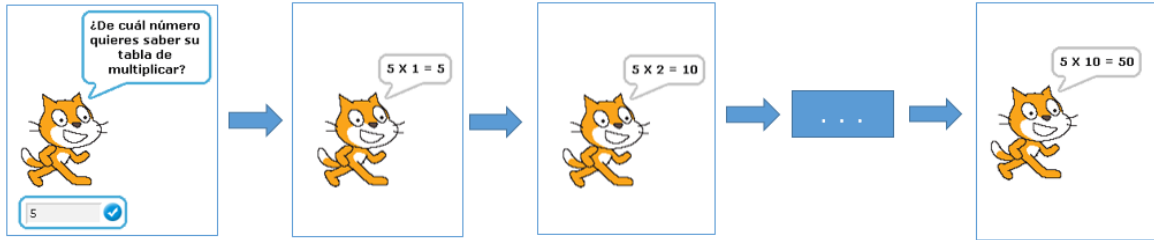
Algoritmo 4:

1. Iniciar programa
2. Crear variable numero
3. Fijar **numero** a 5
4. Repetir por 5 veces
5. Si **numero** es impar
 - 5.1. Cambiar el disfraz a **numero**
 - 5.2. Esperar 1 segundo
6. Incrementar valor de **numero** en 1 (numero + 1)
7. Fin del programa

Ejercicio 4: Tabla de multiplicar.

Algoritmo en el que:

- Se pida un número y de acuerdo a este,
- Se imprima su tabla de multiplicar.



Algoritmo 1:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Preguntar numero
6. Introducir respuesta
7. Fijar **numero** a respuesta
8. Fijar **contador** a 1
9. Repetir por 10 veces
 - 9.1. Fijar **resultado** a (**numero** * **contador**)
 - 9.2. Decir **numero X contador = resultado**
 - 9.3. Incrementar valor de **contador** en 1 (**contador** + 1)
10. Fin ciclo repetir
11. Fin del programa

Algoritmo 2:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Preguntar numero
6. Introducir respuesta
7. Fijar **numero** a respuesta
8. Fijar **contador** a 1
9. Repetir por 10 veces
 - 9.1. Fijar **resultado** a (**numero** * **numero**)
 - 9.2. Decir **numero X numero = resultado**
 - 9.3. Incrementar valor de **numero** en 1 (**numero** + 1)
10. Fin ciclo repetir
11. Fin del programa

Algoritmo 3:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Introducir respuesta
6. Preguntar numero
7. Fijar **numero** a respuesta
8. Fijar **contador** a 10
9. Repetir por **contador** veces
 - 9.1. Fijar **resultado** a (**numero** * **contador**)
 - 9.2. Decir **numero X contador = resultado**
 - 9.3. Incrementar valor de **contador** en 1 (**contador** + 1)
10. Fin ciclo repetir
11. Fin del programa

Algoritmo 4:

1. Iniciar programa
2. Crear variable numero
3. Crear variable contador
4. Crear variable resultado
5. Introducir respuesta
6. Preguntar numero
7. Fijar **numero** a respuesta
8. Fijar **contador** a 10
9. Repetir por **contador** veces
 - 9.1. Fijar **resultado** a (**numero** * **numero**)
 - 9.2. Decir **numero X numero = resultado**
 - 9.3. Incrementar valor de **numero** en 1 (**numero** + 1)
10. Fin ciclo repetir
11. Fin del programa

ANEXO F: Post prueba de programación básica

Instrucciones: Lee cuidadosamente el ejercicio, identificando qué pasos debes seguir para llegar a la solución. Lee cada uno de los programas y selecciona el correcto, es decir, el que llega a la solución del ejercicio.

Ejercicio 1: ¿Quién es mayor?

Programa en el que se:

- Pregunte la edad de dos personas
- Determine cuál persona es mayor.
- Muestre en un mensaje quién es mayor que quién.

Programa 1:



```
al presionar bandera verde clicada
preguntar ¿Cuál es la edad de Luis? y esperar
fijar edad 1 a respuesta
preguntar ¿Cuál es la edad de José? y esperar
fijar edad 2 a edad 1
si edad 1 > edad 2
  decir José es mayor que Luis por 2 segundos
si no
  decir Luis es mayor que José por 2 segundos
detener todo
```

Programa 2:




```
al presionar bandera verde clicada
preguntar ¿Cuál es la edad de Luis? y esperar
fijar edad 1 a respuesta
preguntar ¿Cuál es la edad de José? y esperar
fijar edad 2 a respuesta
si edad 1 > edad 2
  decir José es mayor que Luis por 2 segundos
si no
  decir Luis es mayor que José por 2 segundos
detener todo
```

Programa 3:



```
al presionar bandera verde clicada
preguntar ¿Cuál es la edad de José? y esperar
fijar edad 1 a respuesta
preguntar ¿Cuál es la edad de Luis? y esperar
fijar edad 2 a respuesta
si edad 1 > edad 2
  decir José es mayor que Luis por 2 segundos
si no
  decir Luis es mayor que José por 2 segundos
detener todo
```

Programa 4:



```
al presionar bandera verde clicada
preguntar ¿Cuál es la edad de Luis? y esperar
fijar edad 1 a edad 1
preguntar ¿Cuál es la edad de José? y esperar
fijar edad 2 a edad 2
si edad 1 > edad 2
  decir José es mayor que Luis por 2 segundos
si no
  decir Luis es mayor que José por 2 segundos
detener todo
```


Ejercicio 2: Promedio de 3 calificaciones.

Programa en el que un personaje:

- Pida 3 calificaciones, las cuales deben ser introducidas mediante el teclado.
- Calcule el promedio de las 3 calificaciones
- Muestre el promedio, así como si el promedio es aprobatorio o no (el promedio mínimo aprobatorio es 7).

Programa 1:

```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a respuesta
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a respuesta
fijar promedio a (calificación 1 + calificación 2 + calificación 3) / 3
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
si no
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Programa 2:

```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a respuesta
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a respuesta
fijar promedio a (calificación 1 + calificación 2 + numero 3)
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
si no
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Programa 3:

```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a calificación 2
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a calificación 3
fijar promedio a (calificación 1 + calificación 2 + calificación 3) / 3
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Programa 4:

```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a calificación 2
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a calificación 3
fijar promedio a (calificación 1 + calificación 2 + calificación 3) / 7
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Ejercicio 3: Números pares

Programa en el que:

- Los números pares del 1 al 10 se muestren individualmente cada uno, es decir, el número debe cambiar de disfraz según el valor del número par.

Programa 1:

```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 10
    si numero mod 2 = 0
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
      cambiar numero por 1
  detener todo
  
```

Programa 2:

```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 1
    si numero mod 2 = 0
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
      cambiar numero por 10
  detener todo
  
```

Programa 3:

```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 10
    si no numero mod 2 = 10
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
  cambiar numero por 10
  detener todo
  
```

Programa 4:

```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 1
    si no numero mod 2 = 10
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
      cambiar numero por 1
  detener todo
  
```

Ejercicio 4: Múltiplos de un número dado.

Programa en el que:

- Se pida un número entero y de acuerdo a este
- Se sume 10 veces ese número mostrando el resultado de cada suma. (Ejemplo: numero = 5, entonces mostrar 5, 10, 15, ..., 50)

Programa 1:

```

al presionar bandera verde clic
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  fijar suma a suma + numero
  decir suma por 2 segundos
detener todo
    
```

Programa 2:

```

al presionar bandera verde clic
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  fijar suma a numero + numero
  decir suma por 2 segundos
detener todo
    
```

Programa 3:

```

al presionar bandera verde clic
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  decir suma por 2 segundos
  fijar suma a suma + numero
detener todo
    
```

Programa 4:

```

al presionar bandera verde clic
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  decir suma por 2 segundos
  fijar suma a numero + numero
detener todo
    
```

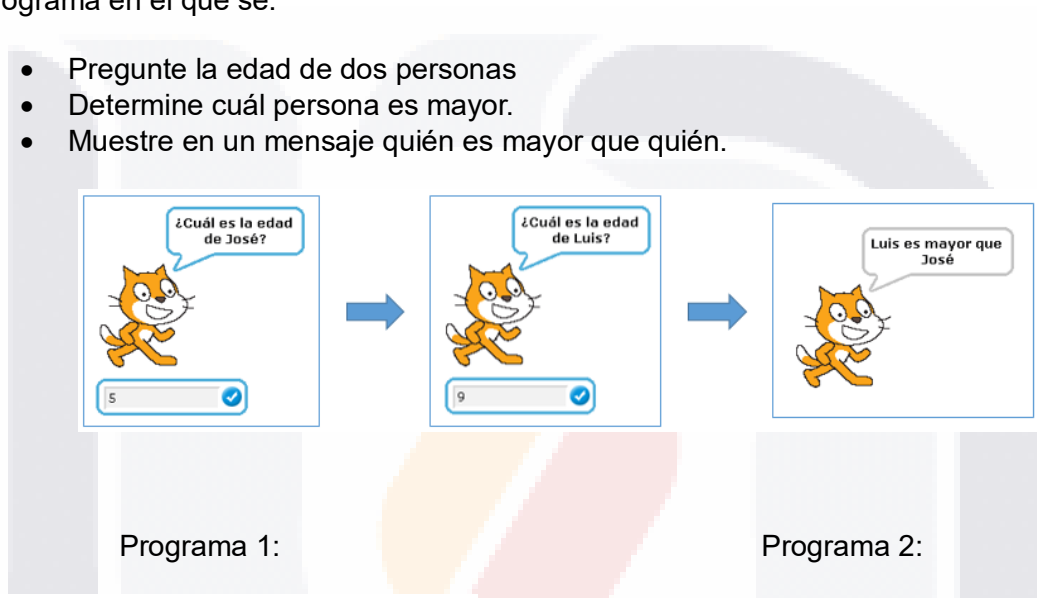
ANEXO G: Calibración - Post prueba de programación básica

Instrucciones: Lee cuidadosamente el ejercicio, identificando qué pasos debes seguir para llegar a la solución. Lee cada uno de los programas y selecciona el correcto, es decir, el que llega a la solución del ejercicio.

Ejercicio 1: ¿Quién es mayor?

Programa en el que se:

- Pregunte la edad de dos personas
- Determine cuál persona es mayor.
- Muestre en un mensaje quién es mayor que quién.



```

al presionar [bandera]
preguntar ¿Cuál es la edad de Luis? y esperar
fijar edad 1 a respuesta
preguntar ¿Cuál es la edad de José? y esperar
fijar edad 2 a edad 1
si edad 1 > edad 2
  decir José es mayor que Luis por 2 segundos
si no
  decir Luis es mayor que José por 2 segundos
detener todo
  
```

```

al presionar [bandera]
preguntar ¿Cuál es la edad de Luis? y esperar
fijar edad 1 a respuesta
preguntar ¿Cuál es la edad de José? y esperar
fijar edad 2 a respuesta
si edad 1 > edad 2
  decir José es mayor que Luis por 2 segundos
si no
  decir Luis es mayor que José por 2 segundos
detener todo
  
```

Programa 3:

```
al presionar   
preguntar ¿Cuál es la edad de José? y esperar  
fijar edad 1 a respuesta  
preguntar ¿Cuál es la edad de Luis? y esperar  
fijar edad 2 a respuesta  
si edad 1 > edad 2  
  decir José es mayor que Luis por 2 segundos  
si no  
  decir Luis es mayor que José por 2 segundos  
detener todo 
```

Programa 4:

```
al presionar   
preguntar ¿Cuál es la edad de Luis? y esperar  
fijar edad 1 a edad 1  
preguntar ¿Cuál es la edad de José? y esperar  
fijar edad 2 a edad 2  
si edad 1 > edad 2  
  decir José es mayor que Luis por 2 segundos  
si no  
  decir Luis es mayor que José por 2 segundos  
detener todo 
```



Ejercicio 2: Promedio de 3 calificaciones.

Programa en el que un personaje:

- Pida 3 calificaciones, las cuales deben ser introducidas mediante el teclado.
- Calcule el promedio de las 3 calificaciones
- Muestre el promedio, así como si el promedio es aprobatorio o no (el promedio mínimo aprobatorio es 7).



Programa 1:

```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a respuesta
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a respuesta
fijar promedio a (calificación 1 + calificación 2 + calificación 3) / 3
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
si no
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Programa 2:

```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a respuesta
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a respuesta
fijar promedio a calificación 1 + calificación 2 + numero 3
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
si no
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Programa 4:

```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a calificación 2
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a calificación 3
fijar promedio a (calificación 1 + calificación 2 + calificación 3) / 3
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Programa 3:

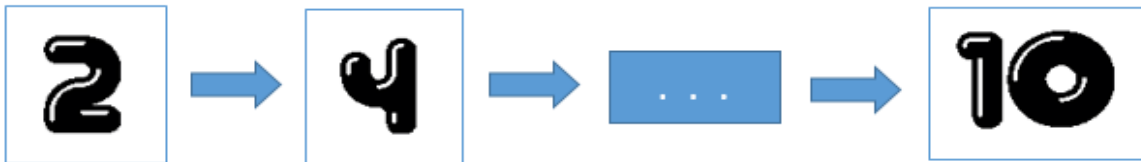
```

al presionar
preguntar ¿Primera calificación? y esperar
fijar calificación 1 a respuesta
preguntar ¿Segunda calificación? y esperar
fijar calificación 2 a calificación 2
preguntar ¿Tercera calificación? y esperar
fijar calificación 3 a calificación 3
fijar promedio a (calificación 1 + calificación 2 + calificación 3) / 7
decir unir El promedio es promedio por 2 segundos
si promedio < 7
decir unir promedio NO es aprobatorio por 2 segundos
decir unir promedio SI es aprobatorio por 2 segundos
detener todo
    
```

Ejercicio 3: Números pares

Programa en el que:

- Los números pares del 1 al 10 se muestren individualmente cada uno, es decir, el número debe cambiar de disfraz según el valor del número par.



Programa 1:

```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 10
    si numero mod 2 = 0
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
    cambiar numero por 1
  detener todo
  
```

Programa 2:

```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 1
    si numero mod 2 = 0
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
    cambiar numero por 10
  detener todo
  
```

Programa 3:

```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 10
    si no numero mod 2 = 10
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
    cambiar numero por 10
  detener todo
  
```

Programa 4:

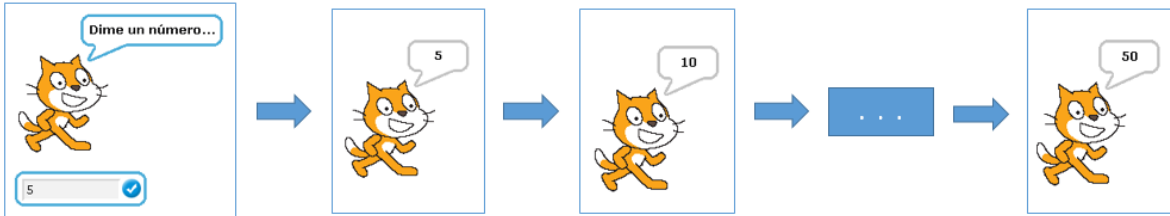
```

al presionar [bandera]
  esconder
  esperar 1 segundos
  fijar numero a 1
  repetir 1
    si no numero mod 2 = 10
      cambiar el disfraz a numero
      mostrar
      esperar 1 segundos
    cambiar numero por 1
  detener todo
  
```

Ejercicio 4: Múltiplos de un número dado.

Programa en el que:

- Se pida un número entero y de acuerdo a este
- Se sume 10 veces ese número mostrando el resultado de cada suma. (Ejemplo: numero = 5, entonces mostrar 5, 10, 15, ..., 50).



Programa 1:

```

al presionar bandera
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  fijar suma a suma + numero
  decir suma por 2 segundos
detener todo
    
```

Programa 2:

```

al presionar bandera
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  fijar suma a numero + numero
  decir suma por 2 segundos
detener todo
    
```

Programa 3:

```

al presionar bandera
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  decir suma por 2 segundos
  fijar suma a suma + numero
detener todo
    
```

Programa 4:

```

al presionar bandera
preguntar Dime un número... y esperar
fijar numero a respuesta
fijar suma a 0
repetir 10
  decir suma por 2 segundos
  fijar suma a numero + numero
detener todo
    
```


ANEXO H: Valores críticos de la distribución de t

A (1 TAIL)	0.05	0.025	0.01	0.005	0.0025	0.001	0.0005
A (2 TAIL)	0.1	0.05	0.02	0.01	0.005	0.002	0.001
GL							
1	6.3138	12.7065	31.8193	63.6551	127.3447	318.4930	636.0450
2	2.9200	4.3026	6.9646	9.9247	14.0887	22.3276	31.5989
3	2.3534	3.1824	4.5407	5.8408	7.4534	10.2145	12.9242
4	2.1319	2.7764	3.7470	4.6041	5.5976	7.1732	8.6103
5	2.0150	2.5706	3.3650	4.0322	4.7734	5.8934	6.8688
6	1.9432	2.4469	3.1426	3.7074	4.3168	5.2076	5.9589
7	1.8946	2.3646	2.9980	3.4995	4.0294	4.7852	5.4079
8	1.8595	2.3060	2.8965	3.3554	3.8325	4.5008	5.0414
9	1.8331	2.2621	2.8214	3.2498	3.6896	4.2969	4.7809
10	1.8124	2.2282	2.7638	3.1693	3.5814	4.1437	4.5869
11	1.7959	2.2010	2.7181	3.1058	3.4966	4.0247	4.4369
12	1.7823	2.1788	2.6810	3.0545	3.4284	3.9296	4.3178
13	1.7709	2.1604	2.6503	3.0123	3.3725	3.8520	4.2208
14	1.7613	2.1448	2.6245	2.9768	3.3257	3.7874	4.1404
15	1.7530	2.1314	2.6025	2.9467	3.2860	3.7328	4.0728
16	1.7459	2.1199	2.5835	2.9208	3.2520	3.6861	4.0150
17	1.7396	2.1098	2.5669	2.8983	3.2224	3.6458	3.9651
18	1.7341	2.1009	2.5524	2.8784	3.1966	3.6105	3.9216
19	1.7291	2.0930	2.5395	2.8609	3.1737	3.5794	3.8834
20	1.7247	2.0860	2.5280	2.8454	3.1534	3.5518	3.8495
21	1.7207	2.0796	2.5176	2.8314	3.1352	3.5272	3.8193
22	1.7172	2.0739	2.5083	2.8188	3.1188	3.5050	3.7921
23	1.7139	2.0686	2.4998	2.8073	3.1040	3.4850	3.7676
24	1.7109	2.0639	2.4922	2.7970	3.0905	3.4668	3.7454
25	1.7081	2.0596	2.4851	2.7874	3.0782	3.4502	3.7251
26	1.7056	2.0555	2.4786	2.7787	3.0669	3.4350	3.7067
27	1.7033	2.0518	2.4727	2.7707	3.0565	3.4211	3.6896
28	1.7011	2.0484	2.4671	2.7633	3.0469	3.4082	3.6739
29	1.6991	2.0452	2.4620	2.7564	3.0380	3.3962	3.6594
30	1.6973	2.0423	2.4572	2.7500	3.0298	3.3852	3.6459
31	1.6955	2.0395	2.4528	2.7440	3.0221	3.3749	3.6334
32	1.6939	2.0369	2.4487	2.7385	3.0150	3.3653	3.6218
33	1.6924	2.0345	2.4448	2.7333	3.0082	3.3563	3.6109
34	1.6909	2.0322	2.4411	2.7284	3.0019	3.3479	3.6008
35	1.6896	2.0301	2.4377	2.7238	2.9961	3.3400	3.5912
36	1.6883	2.0281	2.4345	2.7195	2.9905	3.3326	3.5822
37	1.6871	2.0262	2.4315	2.7154	2.9853	3.3256	3.5737
38	1.6859	2.0244	2.4286	2.7115	2.9803	3.3190	3.5657
39	1.6849	2.0227	2.4258	2.7079	2.9756	3.3128	3.5581
40	1.6839	2.0211	2.4233	2.7045	2.9712	3.3069	3.5510

A (1 TAIL)	0.05	0.025	0.01	0.005	0.0025	0.001	0.0005
A (2 TAIL)	0.1	0.05	0.02	0.01	0.005	0.002	0.001
GL							
41	1.6829	2.0196	2.4208	2.7012	2.9670	3.3013	3.5442
42	1.6820	2.0181	2.4185	2.6981	2.9630	3.2959	3.5378
43	1.6811	2.0167	2.4162	2.6951	2.9591	3.2909	3.5316
44	1.6802	2.0154	2.4142	2.6923	2.9555	3.2861	3.5258
45	1.6794	2.0141	2.4121	2.6896	2.9521	3.2815	3.5202
46	1.6787	2.0129	2.4102	2.6870	2.9488	3.2771	3.5149
47	1.6779	2.0117	2.4083	2.6846	2.9456	3.2729	3.5099
48	1.6772	2.0106	2.4066	2.6822	2.9426	3.2689	3.5051
49	1.6766	2.0096	2.4049	2.6800	2.9397	3.2651	3.5004
50	1.6759	2.0086	2.4033	2.6778	2.9370	3.2614	3.4960
51	1.6753	2.0076	2.4017	2.6757	2.9343	3.2579	3.4917
52	1.6747	2.0066	2.4002	2.6737	2.9318	3.2545	3.4877
53	1.6741	2.0057	2.3988	2.6718	2.9293	3.2513	3.4838
54	1.6736	2.0049	2.3974	2.6700	2.9270	3.2482	3.4800
55	1.6730	2.0041	2.3961	2.6682	2.9247	3.2451	3.4764
56	1.6725	2.0032	2.3948	2.6665	2.9225	3.2423	3.4730
57	1.6720	2.0025	2.3936	2.6649	2.9204	3.2394	3.4696
58	1.6715	2.0017	2.3924	2.6633	2.9184	3.2368	3.4663
59	1.6711	2.0010	2.3912	2.6618	2.9164	3.2342	3.4632
60	1.6706	2.0003	2.3901	2.6603	2.9146	3.2317	3.4602
61	1.6702	1.9996	2.3890	2.6589	2.9127	3.2293	3.4573
62	1.6698	1.9990	2.3880	2.6575	2.9110	3.2269	3.4545
63	1.6694	1.9983	2.3870	2.6561	2.9092	3.2247	3.4518
64	1.6690	1.9977	2.3860	2.6549	2.9076	3.2225	3.4491
65	1.6686	1.9971	2.3851	2.6536	2.9060	3.2204	3.4466
66	1.6683	1.9966	2.3842	2.6524	2.9045	3.2184	3.4441
67	1.6679	1.9960	2.3833	2.6512	2.9030	3.2164	3.4417
68	1.6676	1.9955	2.3824	2.6501	2.9015	3.2144	3.4395
69	1.6673	1.9950	2.3816	2.6490	2.9001	3.2126	3.4372
70	1.6669	1.9944	2.3808	2.6479	2.8987	3.2108	3.4350
71	1.6666	1.9939	2.3800	2.6468	2.8974	3.2090	3.4329
72	1.6663	1.9935	2.3793	2.6459	2.8961	3.2073	3.4308
73	1.6660	1.9930	2.3785	2.6449	2.8948	3.2056	3.4288
74	1.6657	1.9925	2.3778	2.6439	2.8936	3.2040	3.4269
75	1.6654	1.9921	2.3771	2.6430	2.8925	3.2025	3.4250
76	1.6652	1.9917	2.3764	2.6421	2.8913	3.2010	3.4232
77	1.6649	1.9913	2.3758	2.6412	2.8902	3.1995	3.4214
78	1.6646	1.9909	2.3751	2.6404	2.8891	3.1980	3.4197
79	1.6644	1.9904	2.3745	2.6395	2.8880	3.1966	3.4180
80	1.6641	1.9901	2.3739	2.6387	2.8870	3.1953	3.4164