



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD CIENCIAS DE LA INGENIERÍA
CARRERA DE INGENIERÍA EN DISEÑO GRÁFICO Y MULTIMEDIA

Proyecto de Investigación previo a
la obtención del título de Ingeniero
en Diseño Gráfico y Multimedia

Título del Proyecto de Investigación:

**“Evaluación meta-analítica sobre el impacto de scratch en el aprendizaje de la
programación por computadora”**

Autor:

Hablich Proaño Juan José Ricardo

Director del Proyecto de Investigación:

Ph.D. Pavel Novoa Hernández

Quevedo – Los Ríos – Ecuador

2019

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo, Juan José Ricardo Hablich Proaño, declaro que la investigación aquí descrita es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este documento, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Juan José Ricardo Hablich Proaño
C.C. # 120620961-9

CERTIFICACIÓN DE CULMINACION DEL PROYECTO DE INVESTIGACIÓN

El suscrito, **Ph.D. Pavel Novoa Hernández**, docente de la Universidad Técnica Estatal de Quevedo, certifica que el egresado, Hablich Proaño Juan José Ricardo realizó el Proyecto de Investigación Previo a la obtención del Título de Ingeniero en Diseño Gráfico y Multimedia, titulada “**EVALUACIÓN META-ANALÍTICA SOBRE EL IMPACTO DE SCRATCH EN EL APRENDIZAJE DE LA PROGRAMACIÓN POR COMPUTADORA**” bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

.....

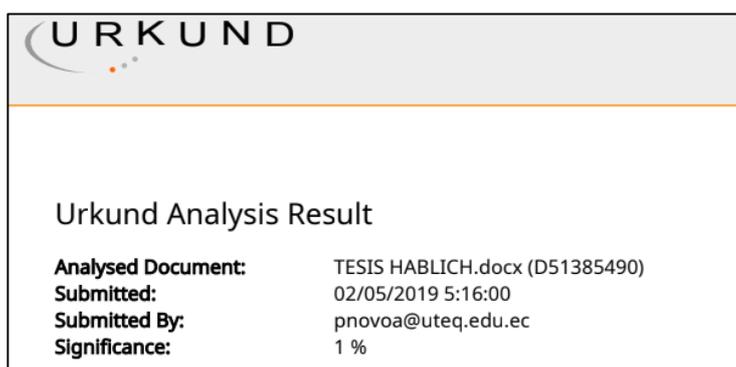
PhD. Pavel Novoa Hernández.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO

El suscrito, Ing. Pavel Novoa Hernández Ph.D., en atención a la Vigésima tercera resolución del Consejo Académico Extraordinario de la Facultad Ciencias de la Ingeniería con fecha 14 de enero del 2019 en la que se me designa Director del Proyecto de Investigación titulado **“EVALUACIÓN META-ANALÍTICA SOBRE EL IMPACTO DE SCRATCH EN EL APRENDIZAJE DE LA PROGRAMACIÓN POR COMPUTADORA”**, me permito exponer lo siguiente:

Que el Sr. Hablich Proaño Juan José Ricardo con cédula de identidad 120620961-9 previo a la obtención del título de Ingeniero en Diseño Gráfico y Multimedia ha cumplido con las correcciones pertinentes del Proyecto de Investigación, de acuerdo a la **ESTRUCTURA Y FORMATO DE PRESENTACIÓN PARA EL PROYECTO DE INVESTIGACIÓN EN LA UNIDAD DE TITULACIÓN ESPECIAL DE LA UTEQ** y respectivamente ingresada al SISTEMA URKUND, tengo a bien certificar la siguiente información, sobre el informe del sistema, el mismo que avala los niveles de originalidad en un noventa y nueve por ciento y de copia un por ciento para los fines de ley.



.....
Ph.D. Pavel Novoa Hernández.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD CIENCIAS DE LA INGENIERÍA
CARRERA DE INGENIERÍA EN DISEÑO GRÁFICO Y MULTIMEDIA

PROYECTO DE INVESTIGACIÓN

Título:

“Evaluación Meta-Analítica sobre el impacto de Scratch en el aprendizaje de la Programación por computadora”

Presentado al Consejo Académico como requisito previo a la obtención del título de Ingeniero en Diseño Gráfico y Multimedia.

Aprobado por:

Ing. Iván F. Jaramillo, MSc.
PRESIDENTE DE TRIBUNAL

Lcdo. Amilkar Puris Cáceres, PhD.
MIEMBRO DEL TRIBUNAL

Ing. Kenya Guerrero Goyes, MSc.
MIEMBRO DEL TRIBUNAL

QUEVEDO – LOS RIOS – ECUADOR
2019

AGRADECIMIENTO

El presente trabajo de tesis primeramente me gustaría agradecer a Dios por bendecirme para llegar hasta donde he llegado, porque hiciste realidad este sueño anhelado. a mi familia por apoyarme en todos estos años de estudio., en especial a mis papas por siempre estar ahí para mi

A mi director de tesis, PhD. Pavel Novoa Hernández por su esfuerzo y dedicación, quien, con sus conocimientos, su experiencia, su paciencia y su motivación ha logrado en mí que pueda terminar el proyecto de investigación.

DEDICATORIA

Dedico esta tesis a Dios

A mi Padre Marcos Proaño y mi Madre María Sánchez especialmente, por haberme criado como una persona de bien, a mis tías Alexandra Proaño, Patricia Proaño y a mi tío Alfredo Proaño que me motivaron durante todo el proceso académico que se sacrificaron por darme una buena educación y me brindaron su apoyo durante toda mi vida.

La dedico también a mi novia Jacqueline por todos los momentos de felicidad que me ha dado desde que la conocí por su apoyo incondicional y su motivación para cumplir todas nuestras metas.

RESUMEN Y PALABRAS CLAVES

El presente proyecto se realizó con el propósito de conocer si el aprendizaje de la programación es significativamente mejorado cuando se emplea el lenguaje de programación visual Scratch. En tal sentido, se desarrolló un meta-análisis basado en los trabajos más relevantes publicados sobre el tema. La metodología empleada sirvió para recopilar toda la información relacionada al tema y para lograr los objetivos propuestos en la investigación. En relación al meta-análisis se procedió a recopilar trabajos relacionados al tema, a través de la consulta a dos bases de datos bibliográficas de alto impacto científico como son el catálogo de Scopus y Web of Science.

Un total de 10 resultados obtenidos a partir de 8 trabajos publicados en el periodo 2008 a 2019, fueron procesados estadísticamente de acuerdo al enfoque de meta-análisis. Se empleó el lenguaje de programación R para el procesamiento estadístico de los datos, lo cual propició la obtención de diversos gráficos de bosque que resumen el impacto de Scratch en el aprendizaje de la programación, así como de otros análisis asociados.

Como resultado, se pudo confirmar que sí, efectivamente Scratch posee un impacto positivo en el aprendizaje de la programación. El tamaño del efecto global, definido como diferencia de media estandarizada, fue significativo e igual a 0,56 con un intervalo de confianza [0,19; 0,92] al 95%. Se pudo comprobar también que este tamaño del efecto no es sensible a la mayoría de las variables moderadoras definidas, excepto por la duración de la intervención. En cuanto al sesgo de publicación, el test de asimetría de Egger mostró que la muestra de estudios seleccionados es heterogénea en cuanto a los resultados y, por tanto, no se puede afirmar la existencia de sesgo.

Palabras claves: Scratch, Meta-Análisis, Lenguaje de Programación Visual, Aprendizaje de la programación.

ABSTRACT AND KEYWORDS

The present project aimed to find out whether programming learning is significantly improved when the visual programming language Scratch is used. In this sense, a meta-analysis was developed based on the most relevant works published on the subject. The methodology used was used to gather all the information related to the subject and to achieve the objectives proposed in the investigation. In relation to the meta-analysis, works related to the subject were collected, through the consultation of two bibliographic databases of high scientific impact, such as the Scopus catalog and the Web of Science.

A total of 10 results obtained from 8 papers published in the period 2008 to 2019, were processed statistically according to the meta-analysis approach. The programming language R was used for the statistical processing of the data, which led to the obtaining of diverse forest graphs that summarize the impact of Scratch on the learning of programming, as well as other associated analyzes.

As a result, it was confirmed that yes, Scratch has a positive impact on learning programming. The size of the global effect, defined as a standardized mean difference, was significant and equal to 0.56 with a confidence interval [0.19; 0.92] to 95%. It was also found that this effect size is not sensitive to most of the moderating variables defined, except for the duration of the intervention. Regarding the publication bias, the Egger asymmetry test showed that the sample of selected studies is heterogeneous in terms of results and, therefore, the existence of bias cannot be affirmed.

Keywords: Scratch, Meta-analysis, Visual Programming Language, Computer programming learning.

TABLA DE CONTENIDOS

Cubierta y portada	i
Declaración de autoría y cesión de derechos.....	ii
Certificación de culminación del proyecto de investigación.....	iii
Certificado del reporte de la herramienta de prevención de coincidencia y/o plagio académico	iv
Certificado de aprobación por tribunal de sustentación	v
Agradecimiento	vi
Dedicatoria.....	vii
Resumen y palabras claves	viii
Abstract and keywords	ix
Código dublin	xv
Introducción.....	1
Capítulo i	2
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN	2
1.1.Problema de Investigación	3
1.1.1. Planteamiento del problema.....	3
1.1.2. Formulación del problema	3
1.1.3. Sistematización	3
1.2. Objetivos	5
1.2.1. General	5
1.2.2. Objetivos Específicos	5
1.3. Justificación.....	6
CAPÍTULO II.....	7
FUNDAMENTACIÓN TEORICA DE LA INVESTIGACIÓN	7
2.1. Marco Conceptual.	8
2.1.1. Leguaje Visual de Programación (LVP)	8
2.1.2. Scratch	11
2.1.3. Fases de un meta-análisis	17
2.2. Marco Referencial	20
CAPITULO III.....	22
METODOLOGÍA DE LA INVESTIGACIÓN	22
3.1. Metodología de la investigación.....	23

3.1.1. Localización.....	23
3.1.2. Tipos de investigación.....	23
3.1.3. Métodos de Investigación.....	24
3.1.4. Fuentes de recopilación de información.....	24
3.1.5. Tratamiento de los datos.....	24
3.1.6. Recursos Humanos, Tecnológicos y Materiales.....	24
3.2. Metodología aplicada al Meta-Análisis.....	25
3.1. Scopus:	25
3.2. Web of Science:.....	25
CAPÍTULO IV	29
RESULTADOS Y DISCUSIÓN	29
4.1. RESULTADOS	30
4.1.1. Características generales de los estudios recopilados	30
4.1.2. Descripción general de los trabajos.....	31
4.1.3. Meta-análisis	34
4.2. Discusión de los resultados	42
CAPÍTULO V.....	44
CONCLUSIONES Y RECOMENDACIONES	44
5.1. Conclusiones	45
5.2. Recomendaciones.....	46
CAPÍTULO VI	47
BIBLIOGRAFÍA	47
BIBLIOGRAFÍA	48
CAPÍTULO VII.....	51
ANEXOS.....	51

INDICE DE TABLAS

Tabla 1 Principales LVP y sus características más importantes	8
Tabla 2 Criterios de inclusión y exclusión.....	26
Tabla 3 Características principales de los estudios seleccionados.....	30
Tabla 4 Descripción de las variables	31

INDICE DE FIGURAS

Figura 1 Escenario de Scratch.....	13
Figura 2 Partes principales del entorno de creación de un proyecto.....	14
Figura 3 Instrucciones para objetos	14
Figura 4 Instrucciones para fondo.....	15
Figura 5 Opciones de la pestaña código	15
Figura 6 Opciones para modificar el fondo	16
Figura 7 Opciones para modificar el sonido	16
Figura 8 Estudios seleccionados	27
Figura 9 Variable moderadora Comparado.Con	32
Figura 10 Variables moderadora Participantes.Universitario	33
Figura 11 Variable moderadora Duracion.Semestral.....	33
Figura 12 Variables moderadora Región y País	33
Figura 13 Gráfico de bosque impacto de Scratch en el aprendizaje de la programación	34
Figura 14 Gráfico de dispersión sobre la influencia de los estudios en la heterogeneidad y el tamaño del efecto.	35
Figura 15 Gráfico de bosque omitiendo estudios de (Wang et al, 2014) y (Cardenas-Cobo et al , 2019)b	35
Figura 16 Gráfico de bosque correspondiente a la variable <i>Comparado.Con</i>	36
Figura 17 Gráfico de bosque correspondiente a la variable <i>Participantes.Universitarios</i>	37
Figura 18 Gráfico de bosque correspondiente a la variable <i>Duracion.Semestral</i>	38
Figura 19 Gráfico de bosque correspondiente a la variable <i>Region</i>	39
Figura 20 Test de Egger con todos los estudios seleccionados	40
Figura 21 Test de Egger sin estudios con mayor valor absoluto	41

ÍNDICE DE ANEXOS

ANEXO 1: Cronograma De Actividades.....	52
---	----

CÓDIGO DUBLIN

Título:	Evaluación Meta-Analítica sobre el impacto de Scratch en el aprendizaje de la programación por computadora.
Autor:	<u>Hablich Proaño Juan José Ricardo</u>
Palabras Clave:	Scratch, Meta-Análisis, Lenguaje de Programación Visual, Aprendizaje de la programación.
Editorial:	Quevedo, UTEQ 2019
Resumen:	Resumen: El presente proyecto se realizó con el propósito de conocer si el aprendizaje de la programación es significativamente mejorado cuando se emplea el lenguaje de programación visual Scratch. En tal sentido, se desarrolló un meta-análisis basado en los trabajos más relevantes publicados sobre el tema. La metodología empleada sirvió para recopilar toda la información relacionada al tema y para lograr los objetivos propuestos en la investigación. En relación al meta-análisis se procedió a recopilar trabajos relacionados al tema, a través de la consulta a dos bases de datos bibliográficas de alto impacto científico como son el catálogo de Scopus y Web of Science. Un total de 10 resultados obtenidos a partir de 8 trabajos publicados en el periodo 2008 a 2019, fueron procesados estadísticamente de acuerdo al enfoque de meta-análisis. Se empleó el lenguaje de programación R para el procesamiento estadístico de los datos, lo cual propició la obtención de diversos gráficos de bosque que resumen el impacto de Scratch en el aprendizaje de la programación, así como de otros análisis asociados. Como resultado, se pudo confirmar que sí, efectivamente Scratch posee un impacto positivo en el aprendizaje de la programación. El tamaño del efecto global, definido como diferencia de media estandarizada, fue significativo e igual a 0.56 con un intervalo de confianza [0,19; 0,92] al 95%. Se pudo comprobar también que este tamaño del efecto no es sensible a la mayoría de las variables moderadoras definidas, excepto por la duración de la intervención. En cuanto al sesgo de publicación, el test de asimetría de Egger mostró que la muestra de estudios

seleccionados es heterogénea en cuanto a los resultados y, por tanto, no se puede afirmar la existencia de sesgo.

Abstract: The present project aimed to find out whether programming learning is significantly improved when the visual programming language Scratch is used. In this sense, a meta-analysis was developed based on the most relevant works published on the subject. The methodology used was used to gather all the information related to the subject and to achieve the objectives proposed in the investigation. In relation to the meta-analysis, works related to the subject was collected, through the consultation of two bibliographic databases of high scientific impact, such as the Scopus catalog and the Web of Science. A total of 10 results obtained from 8 papers published in the period 2008 to 2019, were processed statistically according to the meta-analysis approach. The programming language R was used for the statistical processing of the data, which led to the obtaining of diverse forest graphs that summarize the impact of Scratch on the learning of programming, as well as other associated analyzes. As a result, it was confirmed that yes, Scratch has a positive impact on learning programming. The size of the global effect, defined as a standardized mean difference, was significant and equal to 0.56 with a confidence interval [0.19; 0.92] to 95%. It was also found that this effect size is not sensitive to most of the moderating variables defined, except for the duration of the intervention. Regarding the publication bias, the Egger asymmetry test showed that the sample of selected studies is heterogeneous in terms of results and, therefore, the existence of bias cannot be affirmed.

INTRODUCCIÓN

Durante muchos años se ha programado usando lenguajes textuales de programación haciendo que estos sean la técnica tradicional para programar. Sin embargo, saber programar sigue siendo una habilidad difícil de aprender y enseñar. Un enfoque alternativo que ha surgido en los últimos 20 años son los denominados Lenguajes Visuales de Programación (LVP).

Los LVP son a veces confundidos con los tradicionales lenguajes de programación textuales debido a que algunos de estos cuenta con una interfaz gráfica que permite visualizar lo que se está desarrollando, pero en sí, un LVP es aquel que usa diferentes tipos de gráficos para poder desarrollar un programa.

La programación visual se refiere al desarrollo de programas donde los signos gráficos y elementos de software manejables interactivamente son usados para definir y componer programas. Se puede definir comúnmente a la programación visual como el uso de expresiones visuales (Animaciones, iconos o gráficos) en el desarrollo de un programa, estas son utilizadas para formar sintaxis o la estructura de un programa.

En particular, la programación visual tiene como objetivo general facilitar el aprendizaje de la programación por computadora.

La programación visual está basada en el uso de lenguajes visuales (LVP) estos pueden tener las siguientes características:

- Usan representaciones graficas (Como Dibujos, animaciones o iconos)
- Nos permite programar con expresiones visuales.
- Es un conjunto de arreglos de símbolos y gráficos.

CAPÍTULO I
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1.1. Problema de Investigación

1.1.1. Planteamiento del problema

Durante mucho tiempo la forma de enseñar programación ha sido la misma, esta consiste en enseñar a los estudiantes usando los clásicos lenguajes de programación textuales. Estos consisten en escribir el código fuente ya sea este de un software o de una página web e ir compilando para ver los resultados que vamos obteniendo con el código escrito.

De manera alternativa, la programación visual provee a los estudiantes una herramienta interactiva y atractiva en la que se presta más atención a la semántica del programa que a la sintaxis. Es decir, se emplean elementos gráficos, iconos, dibujos o animaciones haciendo que programar con un lenguaje de programación visual sea mucho más entretenido. Con este grado de motivación se espera por tanto que el estudiante pueda adquirir competencias básicas sobre computación en un tiempo relativamente corto.

En este contexto se encuentra Scratch, un LVP que ha sido ampliamente utilizado en diversos escenarios. Especialmente en el contexto de la enseñanza de computación a niños y adolescentes. Existen además experiencias interesantes en el contexto universitario en los que se emplea Scratch para introducir nociones elementales antes de dominar lenguajes no visuales más profesionales (ej. Java, C#, etc.).

No obstante, aunque esta es una importante herramienta educativa, en la actualidad no se conocen evidencias concretas de su verdadero impacto en la enseñanza de la programación. Las principales investigaciones relacionadas evalúan de forma cualitativa el efecto de Scratch en el aprendizaje, lo cual, aunque importante no brinda fundamentos sólidos sobre el verdadero efecto. En este contexto, se plantea el siguiente problema.

1.1.2. Formulación del problema

¿Cómo medir el impacto del lenguaje de programación visual Scratch en el aprendizaje de programación?

1.1.3. Sistematización

1. ¿Cuáles son los trabajos de la literatura más relevantes sobre intervenciones con Scratch para potenciar el aprendizaje de la programación?

2. ¿Cómo resumir estadísticamente el impacto general de Scratch en el aprendizaje de la programación basado en las experiencias publicadas?
3. ¿Es realmente significativo y positivo el impacto general de Scratch en el aprendizaje de la programación?

1.2. Objetivos

1.2.1. General

- Realizar un meta-análisis sobre el impacto de Scratch en el aprendizaje de programación, basado en los trabajos más relevantes publicados sobre el tema.

1.2.2. Objetivos Específicos

1. Realizar una búsqueda profunda sobre trabajos relacionados con Scratch en escenarios educativos provenientes de catálogos de literatura importantes.
2. Aplicar un meta-análisis sobre los resultados de la búsqueda realizada, para resumir el tamaño del efecto (impacto de Scratch)
3. Discutir crítica y estadísticamente los resultados del meta-análisis desarrollado, exponiendo las principales conclusiones y limitaciones del estudio.

1.3. Justificación

El presente proyecto tiene como objetivo determinar mediante un análisis meta-analítico el impacto del lenguaje visual de programación Scratch en el aprendizaje de la programación.

Desde un punto de vista educativo, aprender a programar siempre ha sido una de las competencias más difíciles de dominar. Esto se debe principalmente a que la programación involucra habilidades relacionadas con la resolución de problemas. A esto se le suma el hecho de que los métodos tradicionales de su enseñanza se basan en lenguajes de programación textuales que emplean sintaxis complejas. Esto último añade una capa de complejidad al proceso de aprendizaje de la programación, pues el estudiante ya no solo debe preocuparse por comprender y resolver algorítmicamente el problema en cuestión, sino también dominar a un nivel de detalle alto, al lenguaje de programación que empleará para implementar su solución.

Scratch es un lenguaje de programación visual diseñado para resolver las dificultades mencionadas anteriormente. Cuando se programa en Scratch el código se va formando en forma de piezas de Tetris o un puzle, lo cual lo hace muy atractivo y divertido al estudiante. Por lo que la sensación de este último es que está inmerso en un juego. No obstante, aún existe incertidumbre si enseñar a programar con Scratch realmente supera al método tradicional o incluso, a otros enfoques alternativos.

Con el desarrollo de la presente investigación se busca dar respuesta a esta cuestión y otras interrogantes que surjan al momento de aplicar o no dentro de un ambiente de aprendizaje, un lenguaje de programación visual como Scratch. Determinar su impacto en el aprendizaje contribuirá a la toma de decisiones de las instituciones que pretendan emplearlo como alternativa de enseñanza de la programación.

CAPÍTULO II

FUNDAMENTACIÓN TEORICA DE LA INVESTIGACIÓN

2.1. Marco Conceptual.

2.1.1. Leguaje Visual de Programación (LVP)

Los lenguajes visuales de programación (LVP) son un tipo especial de lenguaje de programación que permite al ser humano crear programas usando diferentes tipos de gráficos como iconos, animaciones o dibujos en lugar de hacerlo usando texto. Un LVP permite programar con diferentes expresiones visuales, arreglos espaciales de texto y símbolos gráficos, utilizándolos como elementos de sintaxis. Los LVP también son conocidos como programación diagramática. La idea principal de este tipo de programación se basa en cuadros y flechas o piezas de un rompecabezas que se organizan de acuerdo a las acciones que se vayan a programar. [1]

Los LVP también son conocidos como lenguaje de gráficos ejecutables. Esto se debe al empleo de técnicas para diseñar o programar un software en dos o más dimensiones e incluye elementos gráficos, texto, símbolos e iconos dentro de su entorno de programación. [2]

Los LVP no deben confundirse con lenguajes de programación basados en GUI, ya que solo proporcionan servicios de creación de programas gráficos. Sin embargo, su código / contexto es completamente textual. [2]

Existen diferentes tipos de lenguajes visuales entre los que tenemos tanto libres como de pago. En la Tabla 1 se ilustran los principales LVP junto a algunas de sus características.

Tabla 1 Principales LVP y sus características más importantes

LVP	Licencia	Características
Node Red	Open Source - Apache 2.0 license	Los programas en Node-RED se denominan Flujos y se crean a partir de bloques de código prefabricados llamados Nodos. Los usuarios arrastran los nodos a un área de trabajo (o escriben los suyos en un editor de texto usando Javascript), luego los conectan para definir las relaciones lógicas de un flujo. Una extensa base de datos en línea recopila nodos y flujos creados por los usuarios para compartirlos con el resto de la comunidad. [3]

LVP	Licencia	Características
NETLab Toolkit	Licencia gratuita	El kit de herramientas de NETLab, o NTK, muestra los fragmentos de código precompilados como widgets: pequeñas cajas llenas de campos de entrada, controles deslizantes, conmutadores y otros controles en pantalla. Los usuarios eligen widgets adecuados para la tarea en cuestión, establecen los parámetros para cada uno y los conectan en una secuencia lógica para crear un programa. Los programas NTK son compatibles con el hardware de computación integrado basado en Linux original de Arduino y similar. Philip van Allen, profesor en el Art Center College of Design, creó NTK como herramienta de enseñanza y continúa expandiendo la plataforma mediante el lanzamiento de nuevos widgets. [3]
Ardublock	GNU General Public License	"El software Ardublock es un complemento que se agrega al IDE de arduino y que permite al usuario programar en bloques de funciones, un poco como scratch4arduino, o appinventor para Android". [3]
Scratch for Android	Scratch is licensed under GNU General Public License V2 (GPL2)	"S4A es una modificación Scratch que permite la programación simple de la plataforma de hardware de código abierto Arduino. Proporciona nuevos bloques para administrar sensores y actuadores conectados a Arduino". [3]

LVP	Licencia	Características
Scratch	Licencia gratuita	Es un lenguaje de programación visual basado en bloques y una comunidad en línea dirigida principalmente a niños. Los usuarios del sitio pueden crear proyectos en línea usando una interfaz similar a un bloque. El servicio está desarrollado por el MIT Media Lab, se ha traducido a más de 70 idiomas y se utiliza en la mayor parte del mundo. [3]
DG Solution Builder	Demo	"DG Solution Builder, plataforma rápida de desarrollo y visualización de aplicaciones" arrastrar y soltar "permite a los individuos y equipos diseñar aplicaciones y cuadros de mando en tiempo real, sin necesidad de escribir una sola línea de código. Maximiza la eficiencia del análisis y permite una comunicación más rápida a través de cuadros de mando en tiempo real y basados en datos para dispositivos web, de escritorio y móviles ". [3]
Wylidrin	50\$ mensuales	Wylidrin es un servicio basado en web que pretende ser una especie de ventanilla única para la programación sencilla de dispositivos IoT. Los usuarios pueden seleccionar entre una variedad de lenguajes de programación, luego usar la interfaz visual de arrastrar y soltar para construir su código. Al igual que Scratch, los fragmentos de código de Wylidrin se juntan como piezas de un rompecabezas. La plataforma es compatible con muchos tipos populares de hardware de creación de prototipos, incluidos Arduino, Raspberry Pi, Beaglebone, Grove y varios productos Intel. [3]

LVP	Licencia	Características
GraspIO	Licencia gratuita	GraspIO con la función de arrastrar y soltar para programar la electrónica con iPhone, iPad o Android. GraspIO es una placa de desarrollo inteligente que hace que la creación de proyectos de electrónica sea fácil, divertida y accesible. [3]

De la tabla 1 Se puede resaltar que la mayoría de los LVP que se muestran en esta son de licencia gratuita. No obstante, también hay de pago como el Wylidrin y versiones Demo como DG Solution Builder.

Por su particular importancia para la presente investigación, en lo que sigue se describirá en detalle a Scratch, un LVP muy difundido en la actualidad.

2.1.2. Scratch

Scratch es un lenguaje de programación visual y multimedia desarrollado por Media Laboratorios de MIT y por el grupo KIDS de UCLA, Universidad de California, Los Ángeles. Está disponible para todas las plataformas con una licencia gratuita. [4]

Este LVP proporciona un interfaz intuitivo donde se puede emplear diferentes tipos de elementos multimedia como imágenes, sonidos, etc, los cuales permiten desarrollar implementaciones sencillas. En adición a esto, Scratch brinda la opción de publicar el programa que se haya creado en el mismo. [4]

Varios conceptos informáticos y matemáticos pueden ser aprendidos mediante Scratch a través de conceptos visuales. Estos son [4]:

- Los procesos interactivos (bucles),
- Los criterios condicionales (si, entonces, si-no),
- Las coordenadas en un plano,
- Las variables, etc.

Estos conceptos informáticos y matemáticos se comprenden de manera fácil y motivadora debido al contexto en la que se las aprende, ya que no es lo mismo que aprenderlas dentro de un lenguaje de programación tradicional donde solo veremos texto a verlo por medio de un juego que se esté creando dentro de Scratch. [4]

2.1.2.1. Características del Diseño de Scratch

Las características de Scratch son las siguientes:

Bloques: La programación por medio de Scratch se basa en bloques que van formando una construcción estos se asemejan a ladrillos de lego o a las piezas de un rompecabezas. [5]

Elementos multimedia: Esta es una de las características más interesantes de Scratch debido a que permite interactuar con el programa que estemos construyendo de una manera más llamativa además de que este nos da diferentes opciones para editar o manipular el audio las imágenes o cualquier elemento multimedia que tengamos. [5]

Facilidad al compartir: Scratch permite compartir de una forma fácil los proyectos que hayamos creado en el mismo, así como también permite intercambiar *sprites* entre diferentes proyectos. [5]

Estrecha vinculación con el mundo físico: Scratch posibilita la programación con objetos tanto físicos como virtuales esto se puede realizar por medio de sensores físicos como el Scratch board. [5]

Multilinguaje: Scratch puede cambiarse a cualquier idioma que esté disponible incluso en mitad de ejecución, lo cual contribuye a aumentar su usabilidad como software. [5]

2.1.2.2. Interfaz de Scratch

2.1.2.2.1. Escenario

En la Figura 1 Se puede visualizar la pantalla principal de Scratch.

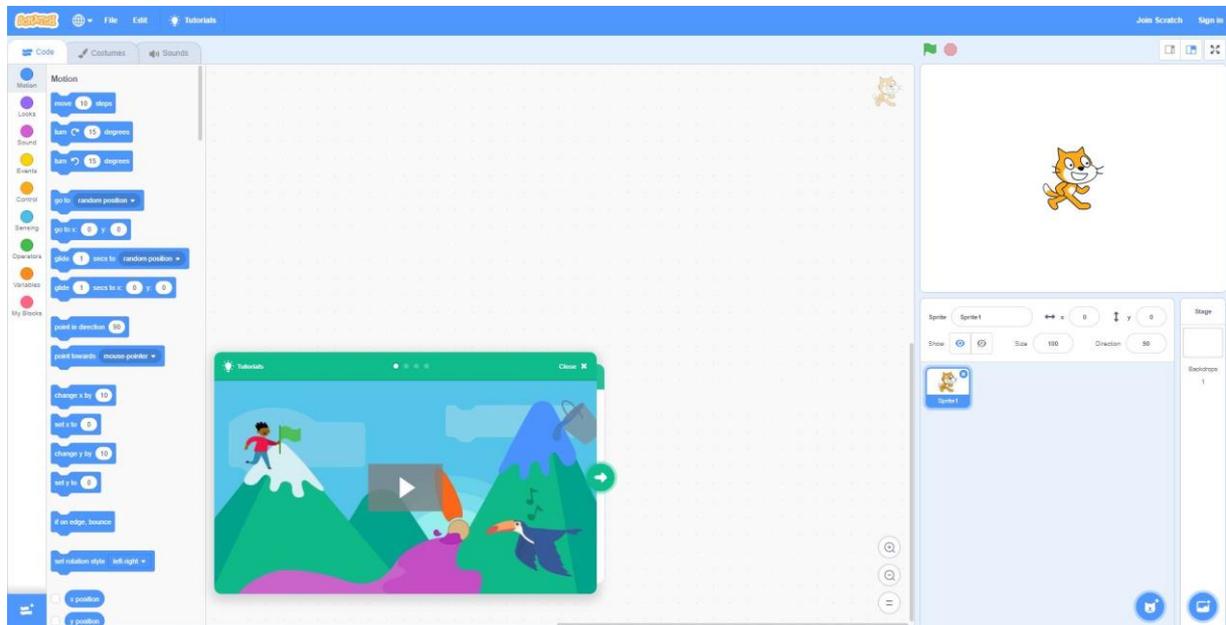


Figura 1 Escenario de Scratch

Las partes principales del entorno de creación de un proyecto se pueden observar en la Figura 2 estas son: [6]

- **Los objetos:** Son los elementos visuales que aparecerán en el proyecto y sobre los que se actuara. [6]
- **El escenario:** Es la zona en la que los objetos trabajarán: moviéndose, girando, etc. [6]
 - El centro del escenario es la coordenada (0,0).
 - La coordenada x varía de -240 a 240.
 - La coordenada y varía desde -180 a 180.
- **El fondo:** Es la imagen o imágenes de fondo del escenario donde irán trabajando los objetos. [6]
- **Todas las instrucciones:** Son todas las posibles instrucciones que se usaran en el proyecto. Las instrucciones también se denominan bloques. [6]

- **Instrucciones del proyecto:** Son las instrucciones que se han elegido para que se ejecuten dentro del proyecto. Las instrucciones del proyecto se consiguen arrastrando instrucciones de la zona “todas las instrucciones”. [6]

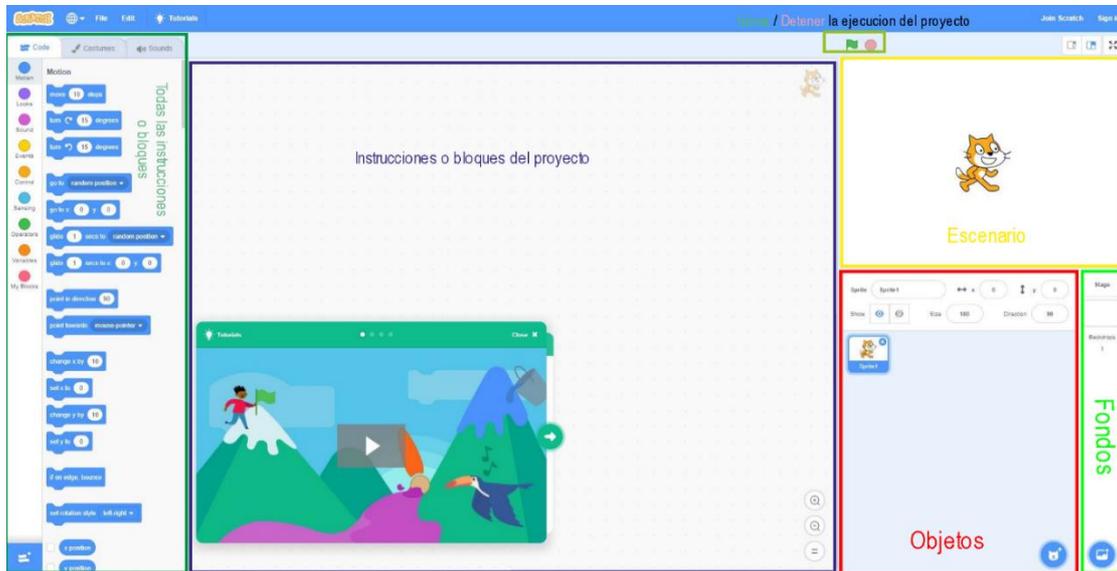


Figura 2 Partes principales del entorno de creación de un proyecto

2.1.2.2.2. Selección de objetos y fondos

Al seleccionar un objeto o al seleccionar un fondo, los bloques que se muestran en la zona "todas las instrucciones" son los que están disponibles para dichos objetos o para los fondos en las Figuras 3 y 4 se puede observar un ejemplo. [6]

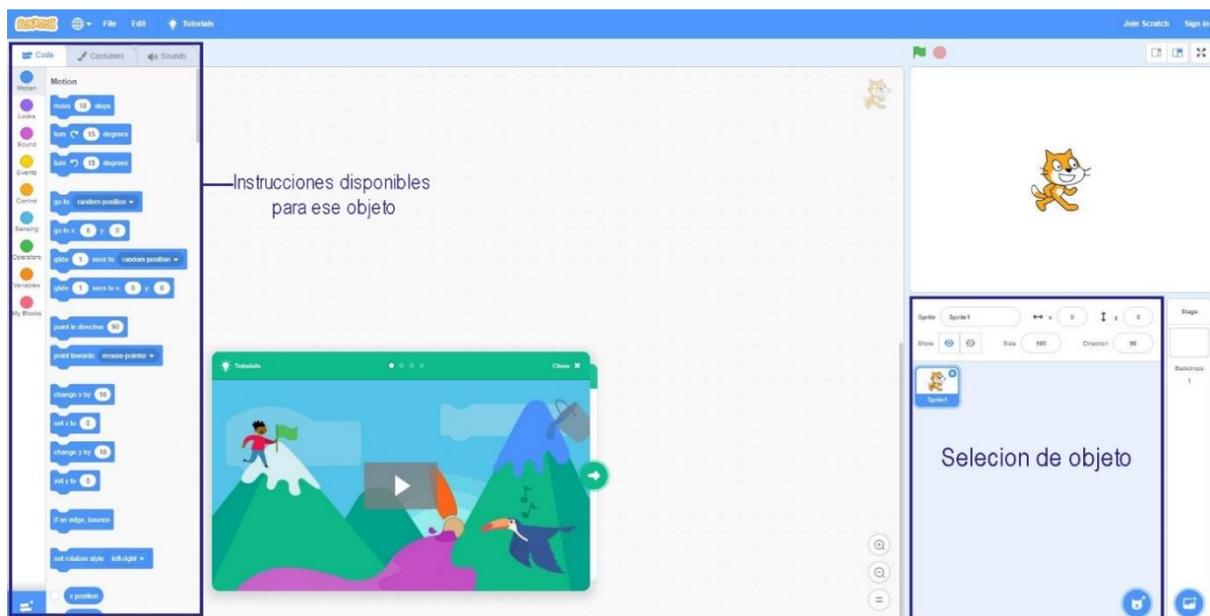


Figura 3 Instrucciones para objetos

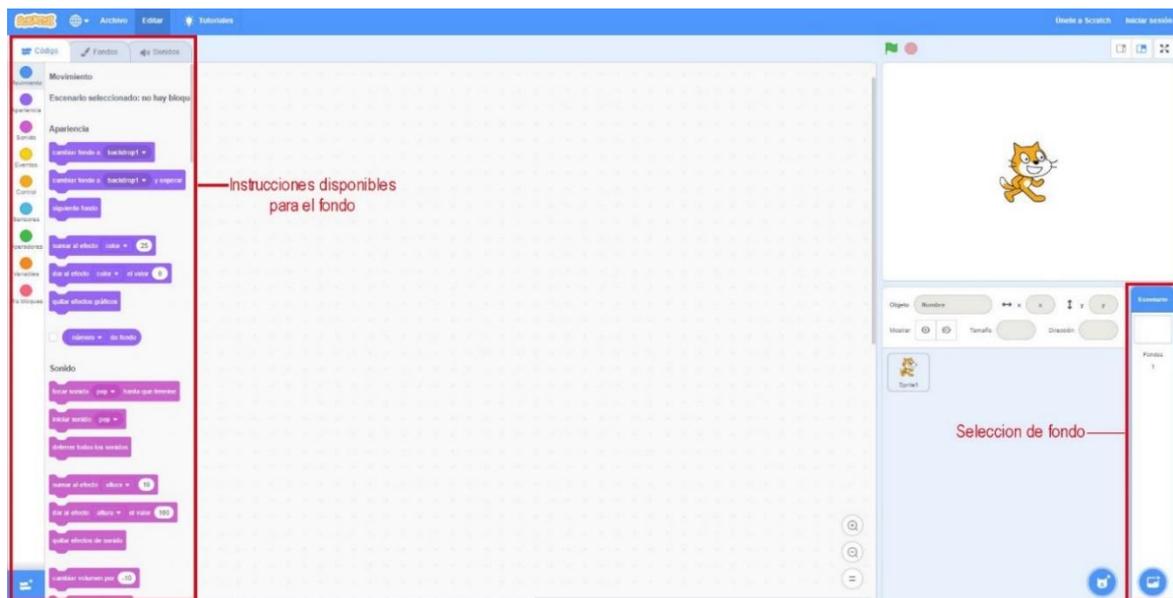


Figura 4 Instrucciones para fondo

Al seleccionar un objeto se pueden realizar las siguientes acciones:

- Definir un código para dicho objeto. Para ello debe estar seleccionada la pestaña "Código" la cual se puede ver en la Figura 5. [6]

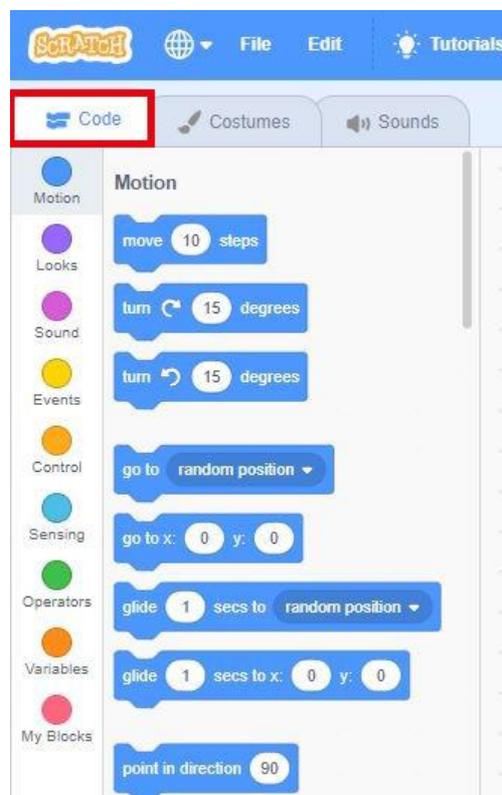


Figura 5 Opciones de la pestaña código

- Modificar el fondo, variando/añadiendo/borrando los fondos. Para ello debe estar seleccionada la pestaña "Fondos". En la Figura 6 se muestra las opciones disponibles para este. [6]

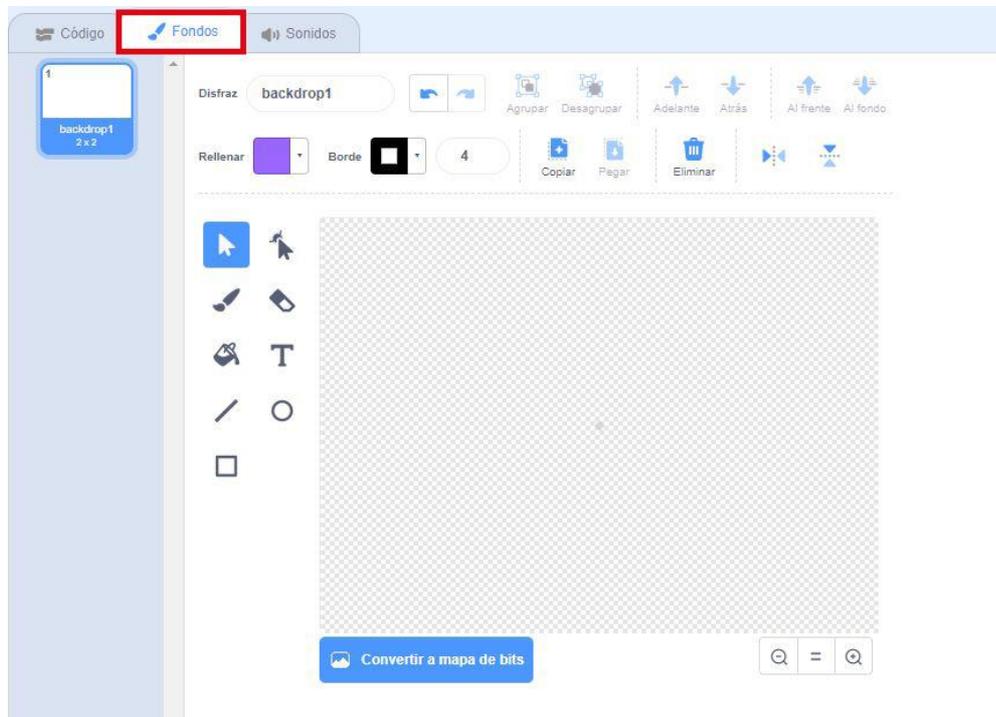


Figura 6 Opciones para modificar el fondo

- Modificar los sonidos asociados al objeto, variando/añadiendo/borrando pistas de audio. Para ello debe estar seleccionada la pestaña "Sonidos". En la Figura 7 se muestran las opciones disponibles para editar los sonidos. [6]

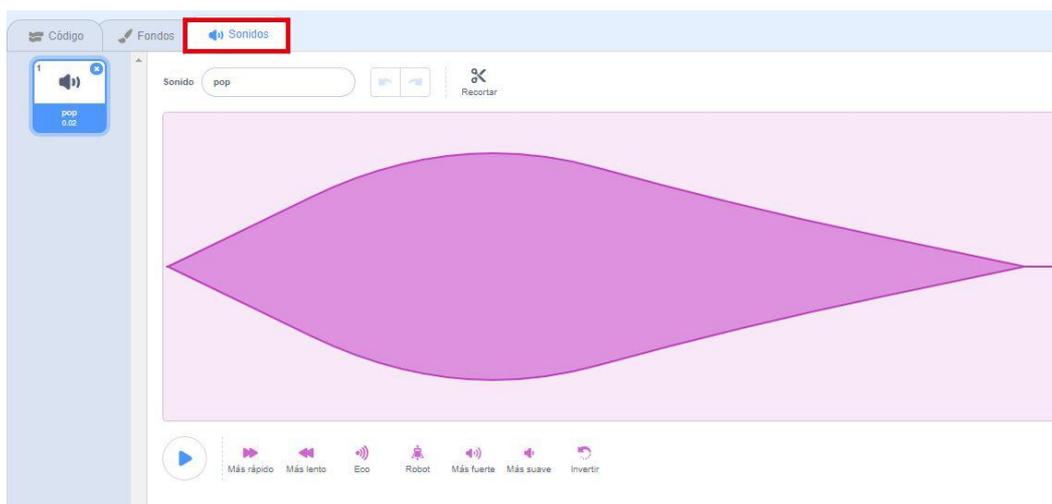


Figura 7 Opciones para modificar el sonido

2.1.2.3. Revisión sistemática.

Una revisión sistemática se puede definir como una investigación científica en la que se revisa los estudios científicos realizados sobre un tema un concreto iniciando con una pregunta formulada de forma clara y objetiva, utilizando métodos sistemáticos para encontrar y seleccionar de manera critica los estudios encontrados referentes a la pregunta formulada, recoger datos e información de dichos estudios con la meta de encontrar conclusiones validas de estos estudios sobre dicho tema [7]. En palabras de Last una revisión sistemática “es la aplicación de estrategias que limitan la comisión de sesgos al integrar, analizar críticamente y sintetizar todos los estudios relevantes sobre un tópico. El meta-análisis puede, aunque no necesariamente, formar parte de este proceso”. [8]

Una revisión sistemática se denomina meta-análisis cuando se logra un índice cuantitativo del impacto que cada uno de los estudios seleccionados para luego aplicar un proceso estadístico. [7] Con esto podemos decir que un meta-análisis es “el análisis estadístico de una gran colección de resultados de trabajos individuales con el propósito de integrar los hallazgos obtenidos” [9] o también como “la síntesis estadística de los datos de estudios diferentes pero similares, es decir, estudios comparables, que proporciona un resumen numérico de los resultados globales” [10].

2.1.3. Fases de un meta-análisis

Para realizar una revisión sistemática o un meta-análisis se deben seguir fases parecidas a las que se realizarían en estudio empírico. Sin embargo, en un meta-análisis el análisis se lo realiza a un estudio empírico mientras que en un estudio empírico el análisis suelen ser los participantes seleccionados de una población, las fases para realizar un meta-análisis tienen ciertas características. En un ámbito general un meta-análisis se puede realizar siguiendo las siguiendo las siguientes fases. [7]

1. Formular el problema
2. Buscar los estudios
3. Análisis de los estudios
4. Calcular el tamaño del efecto
5. Interpretación de los resultados estadísticos
6. Publicación del meta-análisis

2.1.3.1. Formulación del problema

La primera fase del meta-análisis al igual que en una investigación empírica es la formulación del problema que se pretende investigar. Esta fase es igual a la de cualquier tipo de investigación. Para realizar una formulación del problema correcta primero se formula la pregunta que se va a responder. De esta pregunta sacaremos los objetivos que se quieren alcanzar por medio del meta-análisis y la hipótesis que se desea comprobar. [7]

El meta-análisis es muy utilizado dentro del campo de la salud y las ciencias sociales este se usa para responder preguntas de diversos campos. Pero lo más habitual es que se use para valorar la eficacia de programas, tratamientos entre otros, en términos generales podemos decir que sirve para “prevenir, resolver o paliar problemas educativos, psicológicos, de salud o sociales”. [7]

2.1.3.2. Búsqueda de los estudios

Una vez determinados los objetivos del meta-análisis, lo siguiente que se tiene que realizar es la búsqueda de estudios que hayan tratado la pregunta objeto de investigación. Esta fase entra por un proceso donde se filtran los estudios mediante criterios de inclusión y exclusión. Estos criterios son definidos por el objetivo del meta-análisis. Algunos criterios que no pueden faltar son los siguientes: [7]

- Qué tipo de diseño de estudios se aceptarán para el meta-análisis.
- Delimitar que se quiere investigar: tratamientos, programas u otros.
- Definir las características que tendrá la muestra de los estudios.
- Delimitar que datos estadísticos debe contener cada uno de los estudios.
- Definir cómo se va a medir las variables de los resultados.
- Definir el idioma.

Con los criterios de inclusión y exclusión ya definidos podemos comenzar con la búsqueda bibliográfica lo más amplia que sea posible para encontrar la mayor cantidad de estudios que cumplan con los mismos. Dentro de las fuentes que se pueden consultar se encuentran bases bibliográficas electrónicas como: [7]

- ERIC (<https://eric.ed.gov/>)
- PsycInfo (<https://www.ebsco.com/products/research-databases/psycinfo>)
- MedLine (<https://www.nlm.nih.gov/bsd/pmresources.html>)

- CINAHL (<https://www.ebsco.com/e/latam/productos-y-servicios/base-de-datos-para-investigacion/cinahl-complete>)
- Scopus (<https://www.scopus.com/home.uri>)
- WOS (<https://www.fecyt.es/es/recurso/web-science>)

Dependiendo del contexto en el que se desarrolle el meta-análisis se puede consultar directamente en revistas relevantes al tema de investigación “Se recomienda complementar el uso de fuentes formales con fuentes informales de búsqueda, tales como contactar con expertos de reconocido prestigio en el campo para solicitarles estudios no publicados, acceder a los ‘colegios invisibles’ o consultar libros de actas de congresos, tesis doctorales y otras estrategias que permitan acceder, en la medida de lo posible, a la ‘literatura fugitiva’. Con objeto de analizar la fiabilidad del proceso de selección de los estudios, al menos dos codificadores deben realizar dicho proceso de forma independiente y comprobar el grado de acuerdo”. [7]

2.1.3.3. Codificación de los estudios

Después de la selección de los estudios que cumplen con los criterios establecidos se debe realizar un manual de codificación este permitirá codificar las características de los estudios. Esto se realiza debido a que algunas características de los estudios pueden afectar los resultados. Esta tarea es necesaria si se quiere explicar por qué estudios de un tema tienen resultados diferentes o contradictorios. [7]

Con este propósito se elaboran las variables moderadoras de los estudios y se las aplica a los mismos. De estos se distinguen las características metodológicas, que pueden ser la metodología y el diseño de los estudios un ejemplo de esto es el tipo de diseño del estudio, si es un grupo control y un grupo experimental, el tamaño de la muestra entre otras. [7]

2.1.3.4. Cálculo del tamaño del efecto

En esta fase se calcula el efecto obtenido de cada uno de los estudios, este cálculo debe ser de forma homogénea en todos los estudios, de forma de que sea capaz de que todos los estudios tengan la misma métrica. La homogeneización de los resultados es imprescindible porque los estudios analizan los resultados de formas distintas. [7]

Una de las mejores formas de unificar la unidad métrica es por medio del cálculo de un índice del tamaño del efecto. [7] “El tamaño del efecto es un índice estadístico que mide el grado en

que existe el fenómeno que se está investigando en cada estudio y no se deja afectar por el tamaño muestral, como ocurre con las pruebas de significación estadística”. [11]

Análisis estadístico e interpretación. Después de la codificación y el cálculo del tamaño de efecto se crea una base de datos donde las filas son los estudios y las columnas son las potenciales variables moderadoras. Así como el tamaño del efecto obtenido de cada uno de los estudios. Antes de realizar el análisis estadístico se debe realizar un paso previo este no es más que describir las características de los estudios codificados. Luego de eso el meta-análisis pasa por las siguientes fases: [7]

1. Cálculo del tamaño del efecto medio el intervalo de confianza y cuál es la valoración de su significación estadística.
2. Análisis de la heterogeneidad de los tamaños del efecto
3. Realizar un análisis de heterogeneidad de todos los tamaños del efecto
4. Si el resultado del análisis de heterogeneidad es correcto se buscan las variables moderadoras de tal variabilidad.

2.1.3.5. Publicación

Esta es la última fase del meta-análisis como en cualquier otra investigación se debe publicar. Dentro del informe escrito se debe incluir: introducción, método, resultados y discusión y conclusiones.[7]

2.2. Marco Referencial

Existen diferentes artículos y tesis donde se realizan análisis meta-analíticos sobre leguajes programación visual dentro de estas encontramos el artículo de Joana M. Costa y Guilhermina L. Miranda con el tema “Relation between Alice software and programming learning: a systematic review of the literature and meta-analysis” aquí se muestra una revisión sistematica de la literatura, incluido un meta-análisis sobre la efectividad del uso del software Alice en el aprendizaje de programación en comparación con un lenguaje de programación convencional en este estudio se concluyó que hay pocos resultados experimentales sobre la efectividad del lenguaje de programación Alice sin embargo los resultados experimental existentes que se enviaron al meta-análisis nos permitieron asumir, con cierto margen de seguridad, que una estrategia de enseñanza que use Alicia debe obtener resultados más efectivos que el uso de un lenguaje de programación convencional. [12]

Otro trabajo relacionado es el de Partha Pratim Ray con el tema ‘‘A Survey on Visual Programming Languages in Internet of Things’’ en este artículo se muestra una encuesta sobre lenguajes de programación visual en el internet de las cosas aquí se puede ver diferentes tipos de LVP tanto de código abierto como de código cerrado. Como conclusión de este artículo se tiene una evaluación de diferentes parámetros en los VPL tanto de porque razón se escogen más unos que otros como cuales son los más usados también se mencionan los pros y los contras de estos. Fuera de estas cuestiones seleccionadas, la interfaz de usuario pobre, la generación lenta de códigos, la falta de un modelo estandarizado y la ausencia de una capa de acción parecen estar mermando el lugar de los LVP en el momento actual. [13]

Además, encontramos el trabajo de Jesennia Cárdenas-Cobo, Amilkar Puris, Pavel Novoa-Hernández, José A. Galindo and David Benavides con el tema de ‘‘Recommender Systems and Scratch: An integrated approach for enhancing computer programming learning’’ en este estudio se habla sobre una extensión para Scratch llamada CARAMBA que incluye un sistema de recomendación de ejercicios. CARAMBA puede personalizar el aprendizaje de los estudiantes con Scratch sugiriendo ejercicios adecuados para los estudiantes en este artículo se muestran los resultados de una evaluación sobre los efectos de enseñar programación usando la extensión para Scratch CARAMBA tanto en el aprendizaje de conceptos básicos como el rendimiento general de los estudiantes esta investigación arroja resultados positivos en el uso de la extensión CARAMBA como herramienta de aprendizaje dando una tasa de aprobación del 52% que fue un 8% más alta que el uso de Scratch sin la extensión. [14]

CAPITULO III
METODOLOGÍA DE LA INVESTIGACIÓN

3.1. Metodología de la investigación

A continuación, se explica la metodología empleada en la investigación del presente proyecto. La metodología empleada sirvió para recopilar toda la información pertinente para lograr los objetivos propuestos en la investigación. Se establece la localización, tipos de investigación, métodos de investigación y tratamiento de los datos. En la metodología aplicada al Meta-Análisis se procedió a recopilar trabajos relacionados con el tema, a través de la consulta a dos bases de datos bibliográficas de alto impacto científico

3.1.1. Localización.

El proyecto de investigación se desarrolló en LA Universidad Técnica Estatal de Quevedo ubicada en la Av. Quito km 1 ½ vía Santo Domingo en la ciudad de Quevedo de la provincia de Los Ríos. La investigación se desarrolló aproximadamente en 90 días correspondientes al año 2019.



Ilustración 1: Localización UTEQ

3.1.2. Tipos de investigación.

1. **Investigación documental (Bibliográfica):** Se utilizó este tipo de investigación para buscar bibliografías asociadas al tema de investigación, en libros y artículos científicos. Mediante esta investigación se logró recopilar información de distintas fuentes bibliográficas sobre temas relacionados al impacto de Scratch en el aprendizaje de programación, luego se procedió a analizar cada trabajo publicados sobre el tema.

3.1.3. Métodos de Investigación.

2. **Método lógico inductivo.** Debido a la naturaleza propia de los meta-análisis, el principal método adoptado fue el lógico inductivo, pues se pretende elevar conocimientos generales a partir de casos particulares, esto es, a partir de las evidencias recolectadas de la literatura (casos particulares) se busca derivar una afirmación general (impacto de Scratch en el aprendizaje de la programación).

3.1.4. Fuentes de recopilación de información.

Fuentes Secundaria: Las fuentes utilizadas en esta investigación fueron: El catálogo de Scopus adquirido por la Universidad Técnica estatal de Quevedo y Web of Science. Mediante la búsqueda en estas bases de datos bibliográficas se logró obtener trabajos relacionados al tema.

3.1.5. Tratamiento de los datos.

El proceso estadístico de la investigación se desarrolló mediante la recolección de los datos obtenidos basados en una búsqueda de estudios en bases de datos bibliográficas digitales. Se utilizó el programa estadístico R-Studio para el procesamiento de datos en donde se obtuvo distintos diagramas acerca del análisis meta-analítico sobre el impacto de Scratch expuestos en el capítulo de resultados. En Excel se clasificó los estudios que comparaban a Scratch con el método tradicional y otros enfoques distintos al tradicional de acuerdo al método de aprendizaje. Dentro del documento Excel además se agregó un campo donde se especifica las variables moderador.

3.1.6. Recursos Humanos, Tecnológicos y Materiales.

3.1.6.1. Hardware

- ✓ Computador
- ✓ Impresora

3.1.6.2. Software

- ✓ Windows 10 home
- ✓ Microsoft office 2016 (power point, Excel, Word)
- ✓ R-Studio

3.1.6.3. Suministros

- ✓ Anillados

- ✓ Resmas de papel a4
- ✓ Lapiceros

3.1.6.4. Talento humano

El presente proyecto de investigación está elaborado por:

- ✓ **El autor:** Hablich Proaño Juan José Ricardo
- ✓ **Dirigido por:** PhD. Pavel Novoa Hernández director del Proyecto de Investigación.

3.2. Metodología aplicada al Meta-Análisis

El método seleccionado para el desarrollo de la investigación, sigue una estrategia de meta-análisis [15], [16], en el que se busca medir el tamaño del efecto global a partir de la recolección de trabajos relevantes de la literatura. En el caso de la presente investigación, el tamaño del efecto global será el impacto de Scratch en el aprendizaje de programación. Para asegurar la rigurosidad en la búsqueda de la bibliografía asociada, se han seguido los estándares de calidad de la declaración PRISMA [17], en especial, a los criterios de selección de los estudios. Como consecuencia, se procedió recopilar los trabajos relacionados con el tema, a través de la consulta a dos bases de datos bibliográficas de alto impacto científico: Web of Science y Scopus [18].

Los términos incluidos en la consulta fueron seleccionados a partir de las siguientes palabras claves:

3.1. Scopus:

TITLE (scratch AND (learn* OR comput* OR programm* OR "systematic review" OR meta-anal*))

3.2. Web of Science:

TITLE (scratch AND (learn* OR comput* OR programm* OR "systematic review" OR meta-anal*))

Un detalle relevante aquí, es que los principales términos fueron buscados a nivel de título, pero al mismo tiempo, se exigió que los trabajos incluyesen al término ‘meta, scratch’ también en al menos uno de los metadatos. De esta forma, se logra la inclusión de aquellos estudios en los que Scratch fue objeto de investigación.

Los criterios de inclusión y exclusión que se emplearon para llevar a cabo el análisis sistemático de la literatura, se muestran en la Tabla 2. Adicionalmente, se estableció como rango de fechas desde 2001-2019, con el objetivo de poder considerar aquellas las investigaciones desde los inicios de Scratch, hasta la actualidad.

Tabla 2 Criterios de inclusión y exclusión

Criterios de inclusión	Criterios de exclusión
Publicaciones a partir de 2001 hasta 2019.	Estudios duplicados
Accesible en la fuente que lo alberga.	No accesibles
La enseñanza con Scratch fue comparada contra otro método de enseñanza	No es evaluado el aprendizaje con Scratch ni comparado
	Estudios sin datos numéricos

En general, el análisis sistemático de los trabajos se desarrolló en tres fases:

1. Se emplearon las consultas descritas anteriormente en Scopus y WOS;
2. Los resultados obtenidos en la fase 1) fueron filtrados de acuerdo a los criterios de inclusión y exclusión.
3. Se aplicaron filtros relacionados con los criterios. El número de trabajos que quedó como resultado de cada fase se puede observar en la Figura 8.

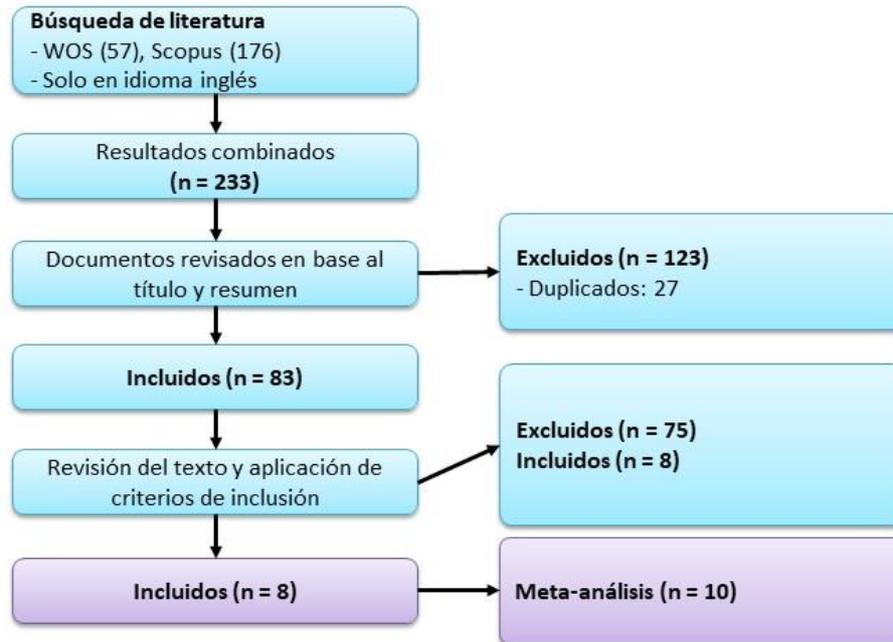


Figura 8 Estudios seleccionados

Dentro de los estudios seleccionados como meta análisis encontramos un total de 8 estudios al revisar cada uno de estos se encontró que en el caso de (Papadakis et al., 2016) y (Cardenas-Cobo et al, 2019) había más de 1 LVP con el que se comparaba a Scratch en este caso se codificaron a estos resultados con las letras en a) y b). Por esta razón el total de estudios de aprendizaje es de 10.

Hasta donde se conoce no hay trabajos que evalúen si la enseñanza de la programación con Scratch es mejor comparada con el método tradicional o contra otros LVP. Por tanto, con la intención de contribuir a esta línea de investigación, el presente trabajo se desarrolló un meta-análisis sobre el impacto de Scratch en el aprendizaje de programación, basado en los trabajos más relevantes publicados sobre el tema.

Para alcanzar dicho objetivo se formularon las siguientes preguntas:

1. ¿Cuál es el impacto global resumido de Scratch en el aprendizaje de programación?
2. ¿Es sensible este impacto a si se compara o no a Scratch con el método tradicional de enseñanza de la programación?
3. ¿Es significativamente diferente este impacto en las experiencias reportadas con estudiantes universitarios a las que emplean estudiantes de otros niveles educativos?

4. ¿Es relevante para el impacto si la duración de la intervención con Scratch es semestral o no?
5. ¿Varía este impacto de acuerdo a la región en donde fue llevado a cabo el estudio?
6. ¿Se publican más trabajos sobre experiencias positivas de Scratch en relación a las que son negativas?

La primera pregunta conduce a un cálculo estimado del tamaño del efecto global, típico de un meta-análisis que contempla datos continuos de un grupo experimental (en este caso Scratch) y uno de control (método tradicional o un enfoque alternativo). Las preguntas de la 2 a la 5 se pueden responder a través de un análisis de variables moderadoras, mientras que la pregunta 6 tiene que ver con el análisis de sesgo de las publicaciones. En el próximo capítulo se describen los resultados estadísticos obtenidos que permiten responder a estas interrogantes.

CAPÍTULO IV
RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

En este capítulo se presentan los resultados de la investigación. Primeramente, se expondrán las características generales de los trabajos de acuerdo a las variables moderadoras definidas. Más adelante, se resumen los resultados del meta-análisis desarrollado.

4.1.1. Características generales de los estudios recopilados

En la Tabla 3 y 4 se resumen las principales características de los 10 trabajos que fueron seleccionados a través del protocolo de búsqueda adoptado. Como se aprecia, los trabajos encontrados han sido publicados en los años comprendidos en el periodo 2008 a 2019, y poseen una distribución diversa en cuanto a los niveles de las restantes variables.

Tabla 3 Características principales de los estudios seleccionados

Autor.Year	Compara do.Con	Participa ntes.Uni versitari o	Duracio n.Semest ral	Región	País
(Yukselturk and Altiok, 2017)[19]	Tradicio nal	Sí	Sí	Eurasia	Turquía
(Cetin, 2016)[20]	Otros	Sí	No	Eurasia	Turquía
(Papadakis et al., 2016)a[21]	Otros	No	Sí	Eurasia	Grecia
(Papadakis et al., 2016)b[21]	Tradicio nal	No	Sí	Eurasia	Grecia
(Cardenas-Cobo et al, 2019)a[14]	Tradicio nal	Sí	Sí	América	Ecuador
(Cardenas-Cobo et al, 2019)b[14]	Otros	Sí	Sí	América	Ecuador
(Erol and Kurt, 2017)[22]	Tradicio nal	No	Sí	Eurasia	Turquía
(Wang et al, 2014)[23]	Tradicio nal	No	No	Eurasia	Taiwán
(De Kereki, 2008)[24]	Tradicio nal	Sí	No	América	Uruguay
(Nam et al, 2010)[25]	Otros	No	No	Eurasia	Corea

Tabla 4 Datos cuantitativos recopilados

Autor.Year	Year	Nc	Mc	Sc	Ne	Me	Se
(Yukselturk and Altiok, 2017)[19]	2017	151	4,55	1,64	151	4,83	1,59
(Cetin, 2016)[20]	2016	28	25,62	2,34	28	26,46	2,36
(Papadakis et al., 2016)a[21]	2016	35	4,25	4,16	34	8,13	5,28
(Papadakis et al., 2016)b[21]	2016	18	3,01	2,8	34	8,13	5,28
(Cardenas-Cobo et al, 2019)a[14]	2019	29	5,25	1,31	29	6,57	1,29
(Cardenas-Cobo et al, 2019)b[14]	2019	30	7,51	0,92	29	6,57	1,29
(Erol and Kurt, 2017)[22]	2017	26	20,32	8,71	26	29,51	9,36
(Wang et al, 2014)[23]	2014	48	3,98	0,59	43	3,69	0,72
(De Kereki, 2008)[24]	2008	15	7,01	1,73	15	7,02	0,69
(Nam et al, 2010)[25]	2010	30	9,55	3,50	30	10,01	3,16

Una vez revisados en profundidad los 10 resultados (Tabla 3) y teniendo en cuenta las variables definidas anteriormente, se observa que el interés por evaluar a Scratch desde un punto de vista de aprendizaje ha crecido a lo largo del tiempo. Por supuesto, la Tabla 3 solo muestra una cantidad muy reducida de los trabajos disponibles sobre este tema, dado que se han dejado fuera investigaciones que no cumplieron con nuestros criterios de inclusión y exclusión.

4.1.2. Descripción general de los trabajos

En la Tabla 4 se observan las variables moderadoras junto a una descripción a detalle da cada una.

Tabla 5 Descripción de las variables

Nombre de la variable	Significado y dominio
Comparado.Con	Con quien es comparado Scrach Nivel: {Método tradicional, Otros}
Participantes.Universitario	Se refiere al nivel de estudio de los participantes presentes en

	el estudio. Nivel {No, SI}
Duracion.Semestral	Tiempo en el cual fue realizado el estudio Nivel: {No, Si}
Región	Regiones en la que se realizaron los estudios Nivel {Eurasi, América}

En las Figuras 9, 10, 11 y 12 se puede observar las diferentes distribuciones que siguen los resultados recopilados de acuerdo a las variables moderadoras.

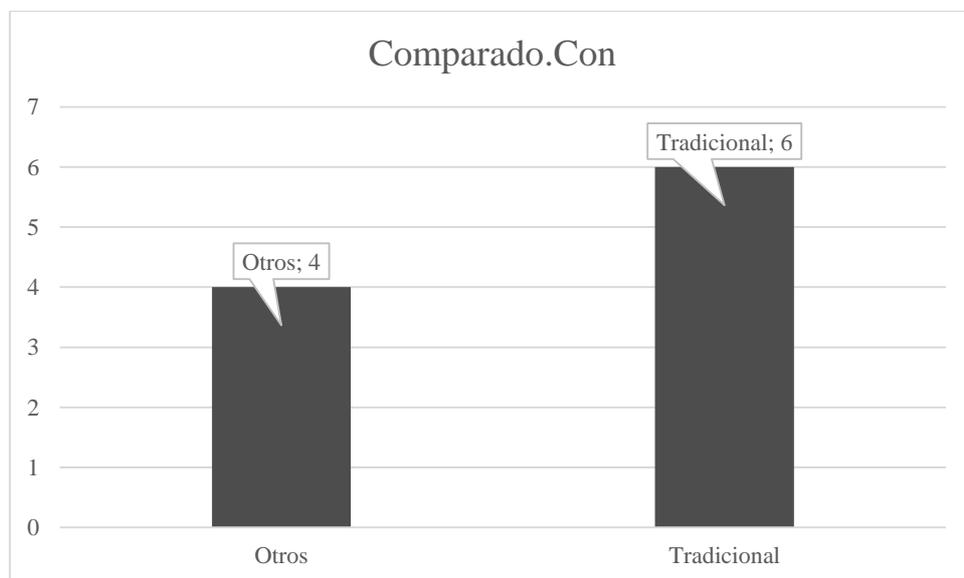


Figura 9 Variable moderadora Comparado.Con

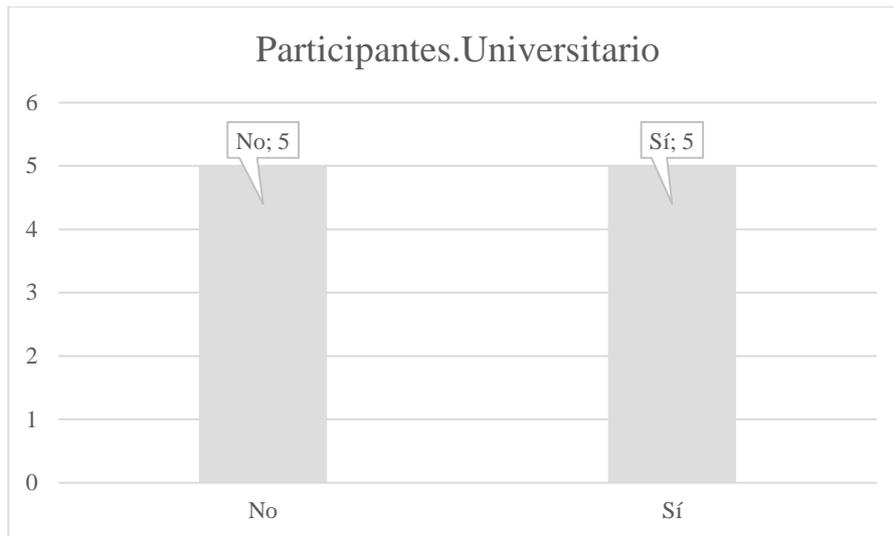


Figura 10 Variables moderadora Participantes.Universitario

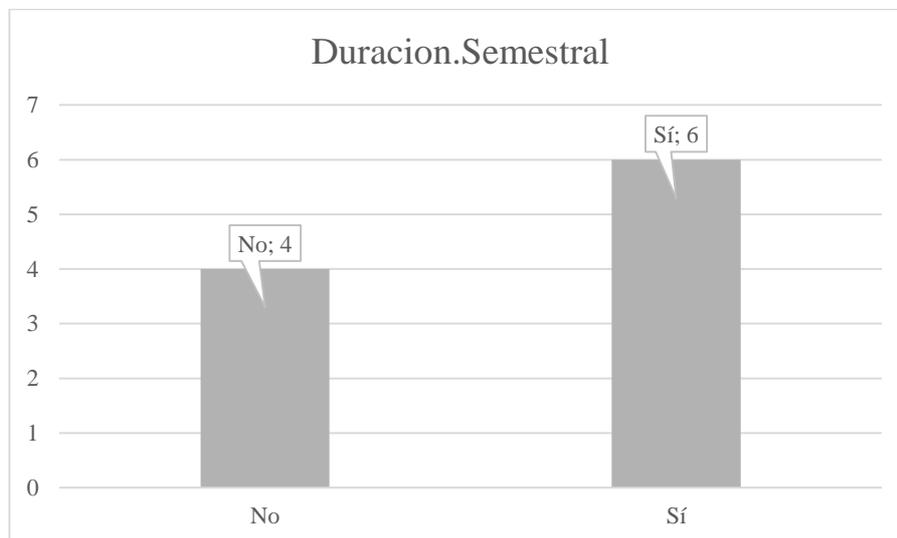


Figura 11 Variable moderadora Duracion.Semestral

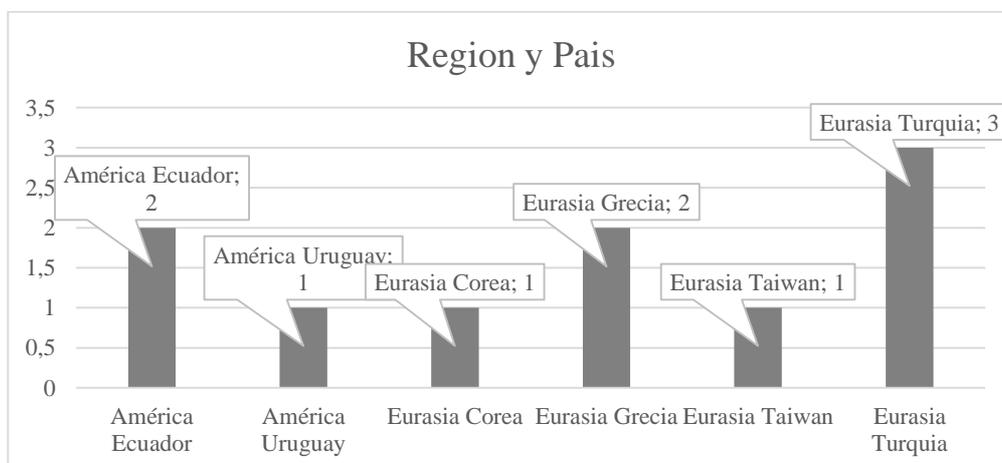


Figura 12 Variables moderadora Región y País

4.1.3. Meta-análisis

Una vez descritas las principales características de los estudios en función de las variables definidas, a continuación, se buscarán respuestas a las preguntas que guían a la investigación a través de un meta-análisis sobre diferencias de medias estandarizadas [26]. Esto supone una comparación entre los niveles de aprendizaje del grupo de control y el grupo experimental (Scratch).

Es importante tener en cuenta que en el meta-análisis basado en la diferencia de medias estandarizadas, el tamaño del efecto está definido entorno al 0. En este caso, este intervalo tendría la siguiente interpretación: un valor superior a 0 indicaría que el impacto de Scratch en el aprendizaje de programación es alto, mientras que un valor inferior a 0, lo contrario

A continuación, se muestran los resultados obtenidos en el meta- análisis:

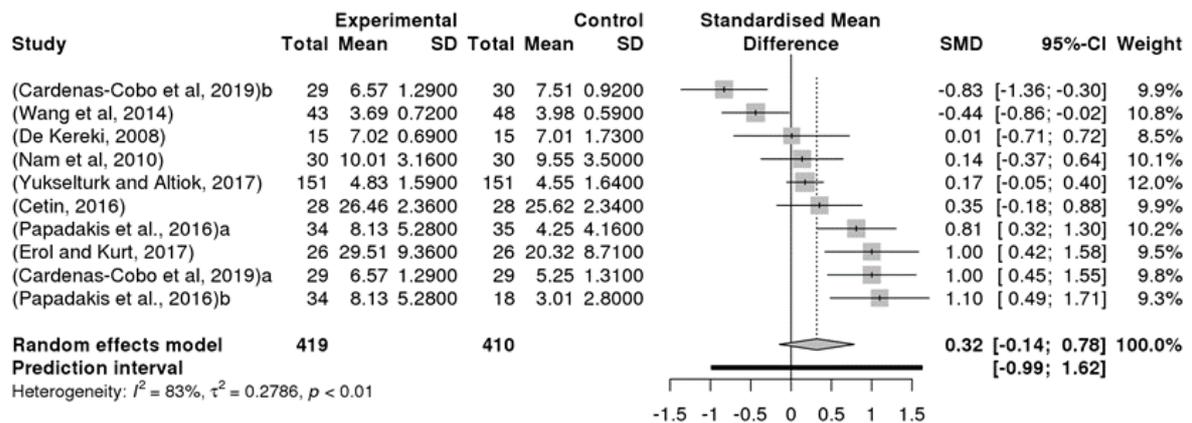


Figura 13 Gráfico de bosque impacto de Scratch en el aprendizaje de la programación

Al ajustar un modelo de efectos aleatorios teniendo en cuenta todos los estudios, la diferencia de medias estandarizada (impacto de Scratch en el aprendizaje de programación) fue de 0.32, con un intervalo de confianza de [-0.14; 0.78] al 95%. Estos resultados se pueden apreciar en la Figura 13. Esto indica que, de manera general, el impacto de Scratch sobre el aprendizaje de la programación es positivo. No obstante, el grado de heterogeneidad asociado a este resultado fue, de acuerdo al estadígrafo I^2 , del 83%. El p-valor asociado a la prueba de heterogeneidad Q fue inferior a 0.01, lo cual indica que la heterogeneidad no solo es alta [27] sino también significativa. Como consecuencia, el efecto global calculado anteriormente no puede considerarse homogéneo y, por tanto, se debe proceder con un análisis que explique el origen de tal heterogeneidad.

Un análisis leave one out mostró que los estudios con mayor valor absoluto del residuo estudentizado fueron (Wang et al, 2014) y (Cardenas-Cobo et al, 2019) b. Estos trabajos son los que más influyen en la heterogeneidad y en el resumen del tamaño del efecto. Este fenómeno se aprecia mejor en el gráfico de dispersión de la Figura 14.

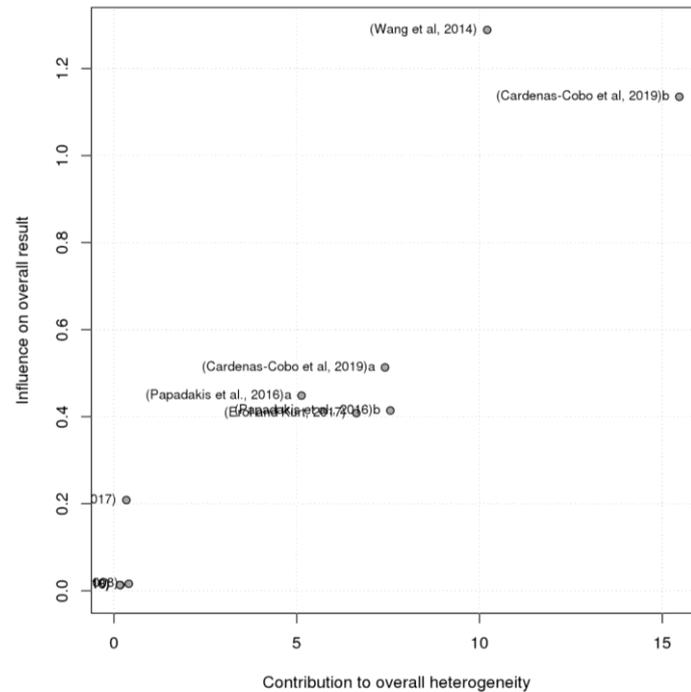


Figura 14 Gráfico de dispersión sobre la influencia de los estudios en la heterogeneidad y el tamaño del efecto.

Sin estos trabajos, un nuevo cálculo de la diferencia de medias estandarizada dio como resultado 0,56, con un intervalo de confianza del 95%, [0,19; 0,92] los cuales se pueden observar en la Figura 15.

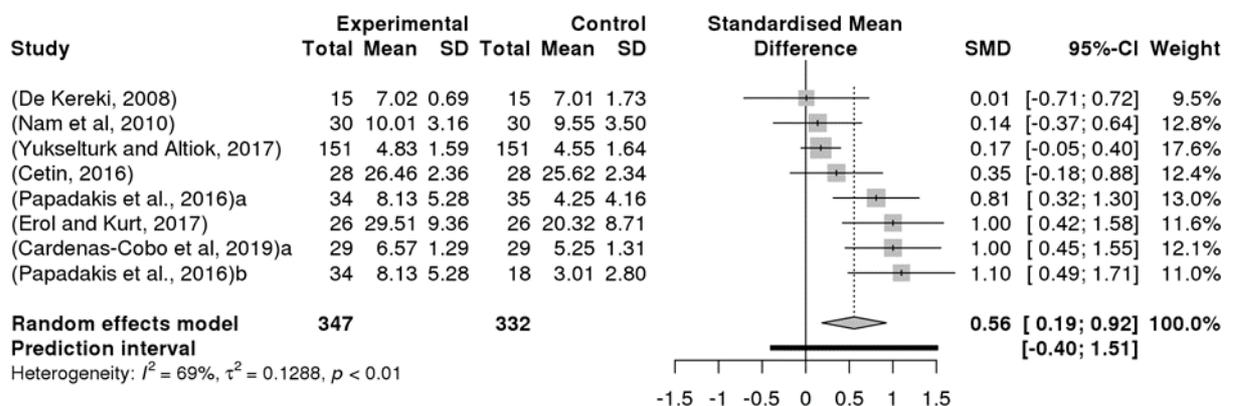


Figura 15 Gráfico de bosque omitiendo estudios de (Wang et al, 2014) y (Cardenas-Cobo et al, 2019)b

La heterogeneidad descendió apreciablemente ($I^2=69\%$), aunque sigue siendo significativa (p -valor < 0.01). La Figura 15 resume este último análisis, y muestra además por cada estudio, la diferencia de medias estandarizada con su respectivo intervalo de confianza del 95%.

4.1.3.1. Análisis de variables moderadoras

En la Figura 16 se observa que los estudios han sido separados por la variable moderadora *Comparado.Con*. En esta se puede apreciar que los (*Comparado.Con* = Tradicional) tienen como resultado una diferencia de medias estandarizadas de 0.64 con un intervalo de confianza del 95% [0.00; 1.28], una heterogeneidad del 79% y un (p -valor < 0.01) significativo.

Mientras que cuando Scratch es comparado con otros enfoques distintos al tradicional (*Comparado.Con* = Otros) se obtiene un efecto global de 0.44 con un intervalo de confianza del 95% [-0.42; 1.30], una heterogeneidad del 45% y un (p -valor = 0.16) no significativo.

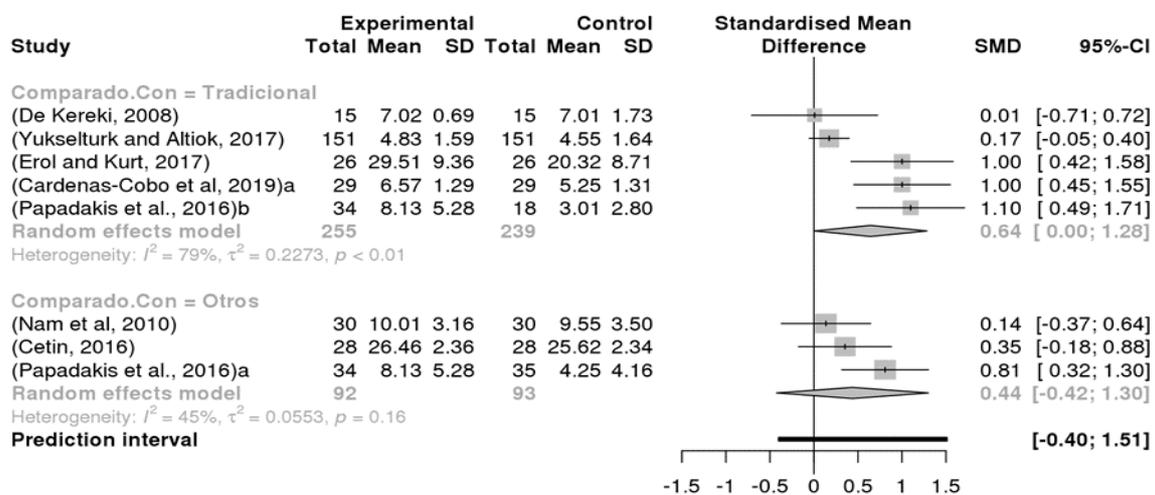


Figura 16 Gráfico de bosque correspondiente a la variable *Comparado.Con*

Aunque el resultado del p-valor no es significativo los resultados que se observan en la Figura 16 son interesantes debido a que al comparar Scratch con el método de enseñanza tradicional este muestra una ligera superioridad frente a este método, mientras que al compararlo con otros enfoques distintos al tradicional el resultado es más parejo.

En la Figura 17 se muestra la variable moderadora *Participantes.Universitarios* la cual se divide en dos grupos (*Participantes.Universitarios = Si*) y (*Participantes.Universitarios = No*)

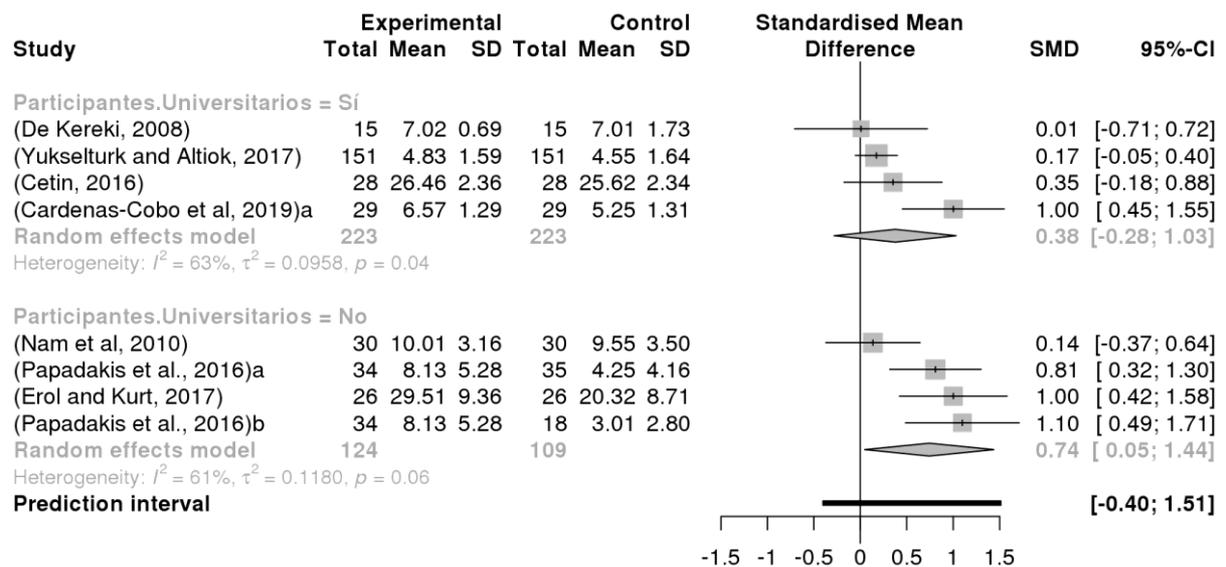


Figura 17 Gráfico de bosque correspondiente a la variable *Participantes.Universitarios*

Dentro de los estudios donde el nivel de estudio de los participantes era universitario el resultado de la diferencia de medias estandarizadas es de 0.38 con un intervalo de confianza del 95% [-0.28; 1.03], una heterogeneidad del 63% y un (p-valor = 0.04) significativo.

Mientras que en los estudios donde los participantes no son de nivel universitario tenemos un resultado de la diferencia de medias estandarizadas es de 0.74 con un intervalo de confianza del 95% [0.05; 1.44], una heterogeneidad del 61% y un (p-valor = 0.06) no significativo.

Según los resultados que se muestran en la Figura 17 se puede decir que el impacto de Scratch en el aprendizaje de la programación en un nivel inferior al universitario resulta más relevante que en un nivel universitario.

En la Figura 18 se representa el gráfico de bosque separado por la variable moderadora *Duración.Semestral*. La cual está separada en (*Duración.Semestral* = Si) y (*Duración.Semestral* = No)

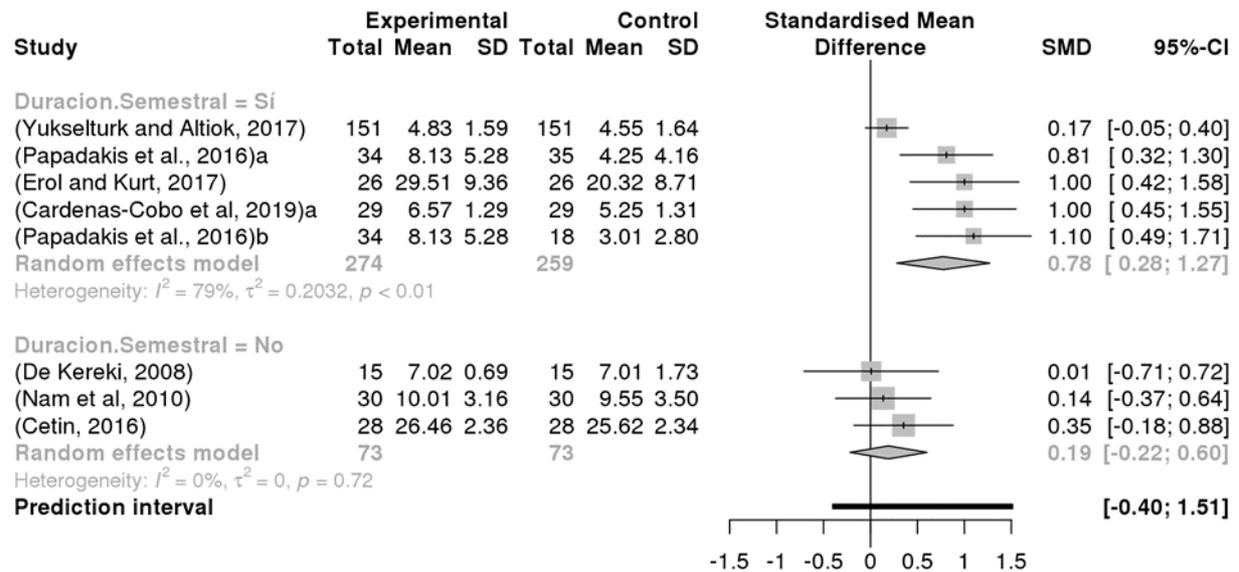


Figura 18 Gráfico de bosque correspondiente a la variable *Duración.Semestral*

En este gráfico podemos apreciar que los estudios que tuvieron una duración semestral nos dieron una diferencia de medias estandarizada de 0.78 con un intervalo de confianza del 95% [0.28; 1.27], una heterogeneidad del 79% y un (p-valor < 0.01).

Mientras que en los estudios que tuvieron una duración de menos de un semestre el resultado de la diferencia de medias estandarizada fue de 0.19 con un intervalo de confianza del 95% [-0.22; 0.60], una heterogeneidad del 0% y un (p-valor = 0.72) no significativo.

En la Figura 18 se aprecia que los estudios realizados en el periodo de un semestre tuvieron mejores resultados que los que no tuvieron esta duración. Con estos resultados y debido a que nuestro p-valor muestra un resultado significativo se puede decir que el impacto de Scratch en el aprendizaje de la programación con una duración de un semestre es superior a los que duran menos de un semestre.

En la Figura 19 se aprecian que los estudios seleccionados han sido separados por la variable moderadora *Region* la cual está dividida en (*Region = Eurasia*) y (*Region = America*)

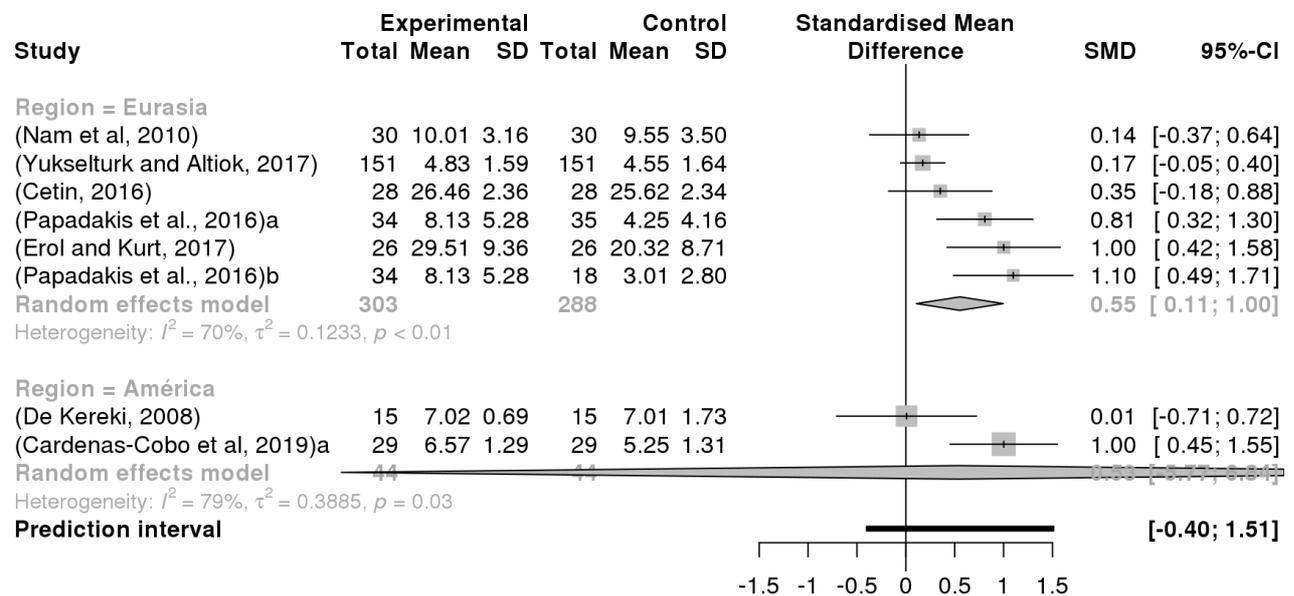


Figura 19 Gráfico de bosque correspondiente a la variable *Region*

En los resultados de los estudios realizados en la región de Eurasia la diferencia de medias estandarizada fue de 0.55 con un intervalo de confianza del 95% [0.11; 1.00], una heterogeneidad del 70% y un (p-valor < 0.01).

Mientras que en la región de América dio una diferencia de medias estandarizada de 0.53 con un intervalo de confianza del 95% [-5.77; 6.84], una heterogeneidad del 79% y un (p-valor = 0.03).

Estos resultados indican que la región no es una variable relevante ya que tanto en la Eurasia como en América se obtuvo una diferencia de medias parecida.

4.1.3.2. Análisis de sesgo de publicaciones

Con el objetivo de identificar posibles sesgos de publicación en la muestra de estudios recolectados, estos fueron representados en un gráfico de embudo (Figura 20), en el que se ilustra cómo se distribuyen de acuerdo a la precisión (errores estándar). La correspondiente prueba de asimetría de Egger dio como resultado un valor z igual a 1.559, con un p -valor asociado de 0.50365. Lo cual indica que no existe asimetría en el modelo y que, por tanto, no se aprecia sesgo de publicación en el conjunto de estudios seleccionados.

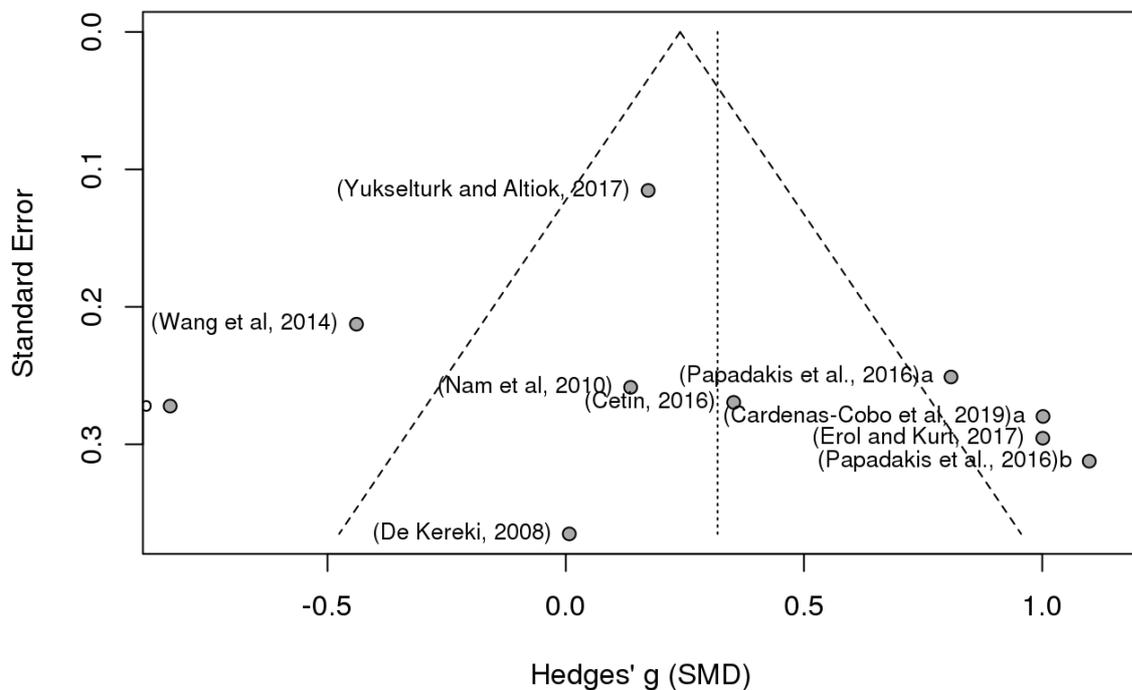


Figura 20 Test de Egger con todos los estudios seleccionados

Así también tenemos la Figura 21 donde se muestra otro test de Egger esta vez sin los estudios de (Wang et al, 2014) y (Cardenas-Cobo et al, 2019) b. que son los estudios con mayor valor absoluto.

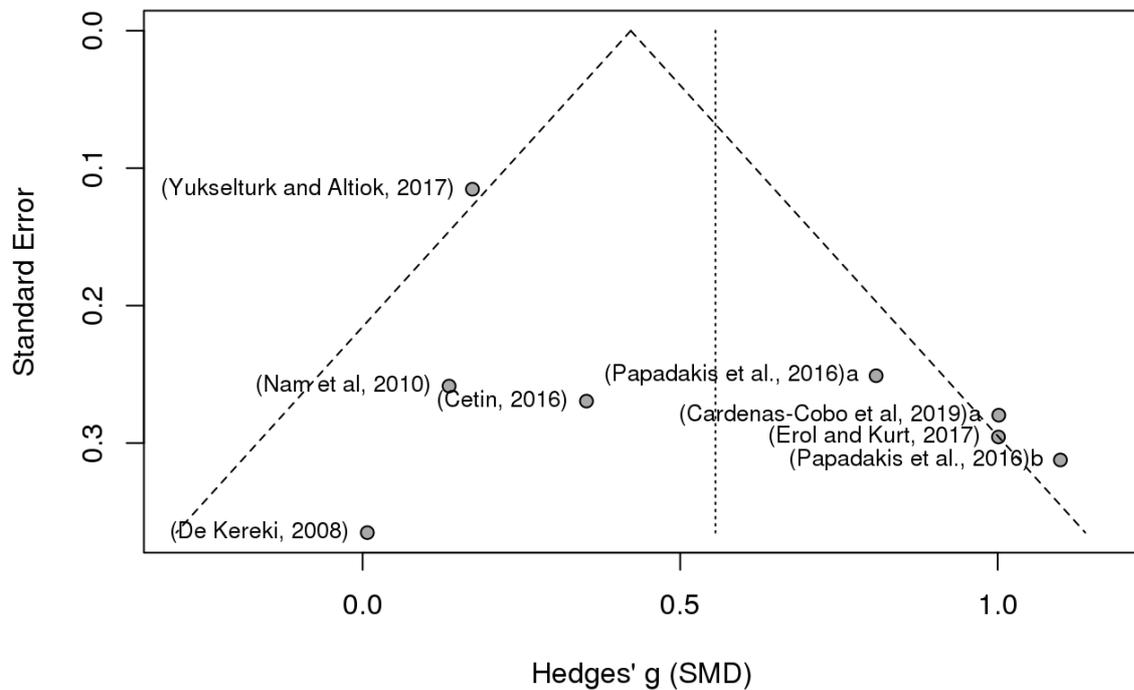


Figura 21 Test de Egger sin estudios con mayor valor absoluto

Aquí se identifican los siguientes resultados un valor z igual a 2.53, con un p-valor asociado de 0.12488 por lo tanto no existe asimetría en el modelo y que, por tanto, no se aprecia sesgo de publicación en el conjunto de estudios seleccionados.

4.2. Discusión de los resultados

Los resultados anteriormente descritos permiten responder, desde un punto de vista cuantitativo, las preguntas que originaron la presente investigación. En lo que sigue, se abordarán estas respuestas en el mismo orden que se formularon las preguntas.

1. ¿Cuál es el impacto global resumido de Scratch en el aprendizaje de programación?

El impacto global resumido de Scratch en el aprendizaje de programación manteniendo todos los estudios da como resultado un SMD de 0,32 con un intervalo de confianza de [-0,14; 0,78] al 95% este valor da a entender que el impacto que tiene Scratch en el aprendizaje de la programación no es muy alto y no se considera homogéneo. Por esta razón se realizó un análisis *leave one out* el cual permitió identificar dos estudios atípicos. En el cálculo sin estos estudios se obtuvo un SMD de 0,56, con un intervalo de confianza del 95%, [0,19; 0,92]. Como se puede apreciar, el valor del SMD aumentó con este resultado y se puede inferir que el impacto que tiene Scratch en el aprendizaje de la programación es positivo.

2. ¿Es sensible este impacto a si se compara o no a Scratch con el método tradicional de enseñanza de la programación?

No, no es sensible de acuerdo al análisis de variables moderadoras desarrollado.

3. ¿Es significativamente diferente este impacto en las experiencias reportadas con estudiantes universitarios a las que emplean estudiantes de otros niveles educativos?

No se puede afirmar que exista una diferencia significativa estadísticamente entre los niveles educativos. No obstante, es importante destacar que el resultado para el caso universitario fue $SMD = 0.38$. Mientras que los resultados que se obtuvieron para el resto de los niveles educativos fue superior ($SMD = 0.74$). Estos resultados permiten concluir que Scratch tiene mayor impacto en los niveles educativos inferiores al universitario.

4. ¿Es relevante para el impacto si la duración de la intervención con Scratch es semestral o no?

Sí, de hecho, esta variable (Duracion.Semestral) resultó significativa como moderadora el de tamaño del efecto calculado. En particular, se pudo observar que cuando la

intervención tiene lugar en todo un semestre, el impacto de Scratch es superior (0.78) a cuando se emplean intervenciones más cortas (0.19).

5. ¿Varía este impacto de acuerdo a la región en donde fue llevado a cabo el estudio?

No varía el impacto de acuerdo a la región. El análisis de la variable Region indicó que esta no modera significativamente al tamaño del efecto.

6. ¿Se publican más trabajos sobre experiencias positivas de Scratch en relación a las que son negativas?

Por medio del test de Egger se pudo apreciar que no existe una asimetría en la distribución de los trabajos. En otras palabras, existe una gran heterogeneidad de resultados tanto en valores positivos como negativos.

Finalmente, es importante indicar que, independientemente de las respuestas y conclusiones derivadas de la presente investigación, es importante advertir que las mismas deben interpretarse con precaución debido a la baja cantidad de estudios considerados en el meta-análisis.

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

La programación de computadoras es una competencia muy importante en la actualidad, en especial para los futuros profesionales de las tecnologías de la información y comunicación. Sin embargo, aún resulta difícil de aprender. Los lenguajes de programación visuales como Scratch han sido propuestos con el objetivo de facilitar este aprendizaje. No obstante, a día de hoy no se conoce cuál es el impacto de Scratch en el aprendizaje de la programación. En tal sentido, la presente investigación desarrolló un meta-análisis de las evidencias publicadas en la literatura para cuantificar dicho impacto. Como principal resultado se pudo determinar que, efectivamente, Scratch posee un impacto no solo positivo, sino también significativo desde el punto de vista estadístico, en el aprendizaje de la programación. Con este resultado se cumple con el objetivo general de la investigación, y se puede arribar a las siguientes conclusiones parciales:

1. Con las sendas consultas desarrolladas en dos de los catálogos más importantes del mundo: Scopus y WOS, se lograron aglutinar una gran cantidad de trabajos relevantes sobre el tema. Aunque la mayoría fue descartada por no cumplir con los requisitos definidos.
2. El meta-análisis desarrollado permitió obtener un valor para el tamaño del efecto (impacto de Scratch en el aprendizaje de la programación) de 0,56, con intervalo de confianza al 95% de [0,19; 0,92]. Esto es consistente con lo dicho anteriormente sobre el impacto positivo de Scratch.
3. Los resultados del meta-análisis fueron discutidos coherentemente durante la exposición de los resultados. Por lo que de esta forma se pudo dar respuestas a las interrogantes que guiaron el desarrollo del meta-análisis. En general, se pudo apreciar que el tamaño del efecto calculado no es sensible a la mayoría de las variables moderadoras, excepto la correspondiente a la duración de la intervención. En adición a esto, no se apreciaron sesgos en las publicaciones consideradas para el estudio.

5.2. Recomendaciones

Para futuras investigaciones se recomienda extender la búsqueda a la plataforma Google Scholar. De esta forma se cubrirían muchos más trabajos referentes al tema para lograr resultados más precisos.

Además, se recomienda aplicar Scratch en instituciones educativas donde se imparta la materia de Programación por ejemplo en la Universidad Técnica Estatal de Quevedo. De acuerdo a los resultados obtenidos en la presente investigación, Scratch posee un impacto positivo en todos los niveles educativos, aunque mucho más en las enseñanzas pre-universitarias.

Al concluir esta investigación también se puede recomendar el uso de los lenguajes de programación visual debido a que estos ofrecen un entorno más llamativo y entretenido a los estudiantes lo cual se transforma en una mayor motivación y atención por parte de los mismos al aprender programación.

CAPÍTULO VI
BIBLIOGRAFÍA

Bibliografía

- [1] B. Jost, M. Ketterl, R. Budde, and T. Leimbach, “Graphical programming environments for educational Robots: Open Roberta - Yet another one?,” *Proc. - 2014 IEEE Int. Symp. Multimedia, ISM 2014*, pp. 381–386, 2015.
- [2] “What is a Visual Programming Language (VPL)? - Definition from Techopedia.” [Online]. Available: <https://www.techopedia.com/definition/22855/visual-programming-language-vpl>. [Accessed: 13-Jun-2019].
- [3] “Visual Programming Guide | 2019 Overview of Available Languages and Software Tools.” [Online]. Available: <https://www.postscapes.com/iot-visual-programming-tools/>. [Accessed: 13-Jun-2019].
- [4] Pablo González Nalda, “Scratch,” 2014.
- [5] Liceo Carmelo, “¿Qué es Scratch?” [Online]. Available: [http://liceocarmelo.edu.uy/Liceo1/XO/Scratch/Manual de Scratch.pdf](http://liceocarmelo.edu.uy/Liceo1/XO/Scratch/Manual%20de%20Scratch.pdf). [Accessed: 13-Jun-2019].
- [6] Unidad De Gestión Educativa Local Andahuaylas, “Manual Básico de Scratch,” 2014.
- [7] J. Sánchez-Meca, “Cómo realizar una revisión sistemática y un meta-análisis*,” vol. 38, no. 2, pp. 53–64, 2010.
- [8] J. M. Last, “A Dictionary of Epidemiology, 4th edition.”
- [9] G. V. Glass, “Primary, Secondary, and Meta-Analysis of Research,” *Educ. Res.*, vol. 5, no. 10, p. 3, Nov. 1976.
- [10] I. Chalmers, L. V. Hedges, and H. Cooper, “A Brief History of Research Synthesis,” *Eval. Health Prof.*, vol. 25, no. 1, pp. 12–37, Mar. 2002.
- [11] J. Cohen, “Statistical Power Analysis for the Behavioral Sciences Second Edition.”
- [12] J. M. Costa and G. L. Miranda, “Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis,” *Br. J. Educ. Technol.*, vol. 48, no. 6, pp. 1464–1474, Nov. 2017.

- [13] P. P. Ray, "A Survey on Visual Programming Languages in Internet of Things," *Sci. Program.*, vol. 2017, pp. 1–6, 2017.
- [14] J. Cardenas-Cobo, A. Puris, P. Novoa-Hernandez, J. A. Galindo, and D. Benavides, "Recommender Systems and Scratch: An integrated approach for enhancing computer programming learning," *IEEE Trans. Learn. Technol.*, pp. 1–1, 2019.
- [15] M. Borenstein, L. V. Hedges, J. P. T. Higgins, and H. R. Rothstein, *Introduction to Meta-Analysis*. Chichester, UK: John Wiley & Sons, Ltd, 2009.
- [16] J. Gurevitch, J. Koricheva, S. Nakagawa, and G. Stewart, "Meta-analysis and the science of research synthesis," *Nature*, vol. 555, no. 7695, pp. 175–182, Mar. 2018.
- [17] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement," *PLoS Med.*, vol. 6, no. 7, p. e1000097, Jul. 2009.
- [18] P. Mongeon and A. Paul-Hus, "The journal coverage of Web of Science and Scopus: a comparative analysis," *Scientometrics*, vol. 106, no. 1, pp. 213–228, 2016.
- [19] E. Yukselturk and S. Altioek, "An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming," *Br. J. Educ. Technol.*, vol. 48, no. 3, pp. 789–801, May 2017.
- [20] I. Cetin, "Preservice Teachers' Introduction to Computing," *J. Educ. Comput. Res.*, vol. 54, no. 7, pp. 997–1021, Dec. 2016.
- [21] N. Zaranis, V. Orfanakis, S. Papadakis, and M. Kalogiannakis, "Using Scratch and App Inventor for teaching introductory programming in Secondary Education. A case study.," *Int. J. Technol. Enhanc. Learn.*, vol. 1, no. 1, p. 1, 2016.
- [22] O. Erol and A. A. Kurt, "The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement," *Comput. Human Behav.*, vol. 77, pp. 11–18, Dec. 2017.
- [23] H. Y. Wang, I. Huang, and G. J. Hwang, "Effects of an Integrated Scratch and Project-Based Learning Approach on the Learning Achievements of Gifted Students in

- Computer Courses,” in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, 2014, pp. 382–387.
- [24] I. F. de Kereki, “Scratch: Applications in Computer Science 1,” in *2008 38th Annual Frontiers in Education Conference*, 2008, pp. T3B-7-T3B-11.
- [25] D. Nam, Y. Kim, and T. Lee, “The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill,” 2010.
- [26] V. N. Nyaga, M. Arbyn, and M. Aerts, “Metaprop: a Stata command to perform meta-analysis of binomial data,” *Arch. Public Heal.*, vol. 72, no. 1, p. 39, Dec. 2014.
- [27] J. P. T. Higgins, “Measuring inconsistency in meta-analyses,” *BMJ*, vol. 327, no. 7414, pp. 557–560, Sep. 2003.

CAPÍTULO VII
ANEXOS

