

Teaching Programming with MIT App Inventor: A Literature Review

Mariuxi Vinueza-Morales, Máster¹, Jorge Rodas-Silva, Máster², Ana Chacón-Luna, Máster³
and Héctor Serrano Mantilla, Máster⁴

¹Universidad Estatal de Milagro, Ecuador, mvinuezam@unemi.edu.ec,
2jrodass@unemi.edu.ec, 3achaconl1@unemi.edu.ec, 4hserranom@unemi.edu.ec

Abstract– MIT App Inventor is a visual block programming environment created for mobile application development by novice programmers, and is widely used around the world for learning programming. This paper describes the literature review conducted by applying search and selection criteria, using specialized search engines, followed by the analysis of the 35 selected studies (experiments and experiences) on the teaching-learning process of programming conducted by teachers from various countries, using App Inventor. The findings indicate a high acceptance in the academic community of App Inventor as an effective tool for motivation and performance of students who are initiated in programming, without distinction in the educational level.

Keywords– App Inventor, Programming, Computational thinking, Programming environment.

Digital Object Identifier (DOI):
<http://dx.doi.org/10.18687/LACCEI2020.1.1.49>
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

Enseñanza de programación mediante MIT *App Inventor*: una revisión de literatura

Mariuxi Vinueza-Morales, Máster¹, Jorge Rodas-Silva, Máster², Ana Chacón-Luna, Máster³
and Héctor Serrano Mantilla, Máster⁴

¹Universidad Estatal de Milagro, Ecuador, mvinuezam@unemi.edu.ec,
²jrodass@unemi.edu.ec, ³sachaconl1@unemi.edu.ec, ⁴hserranom@unemi.edu.ec

Resumen— MIT *App Inventor* es un entorno de programación visual por bloques creado para el desarrollo de aplicaciones para móviles por programadores novatos, y es ampliamente utilizado en todo el mundo para el aprendizaje de programación. En el presente se describe la revisión de literatura realizada a partir de la aplicación de criterios de búsqueda y selección, utilizando motores de búsqueda especializados, seguido del análisis de los 35 estudios seleccionados (experimentos y experiencias) sobre el proceso de enseñanza-aprendizaje de programación realizados por docentes de diversos países, mediante el uso de *App Inventor*. Los hallazgos indican una gran aceptación en la comunidad académica de *App Inventor* como herramienta eficaz para la motivación y el rendimiento de los estudiantes que se inician en la programación, sin distinguir en el nivel educativo.

Palabras clave—*App Inventor*, programación, pensamiento computacional, entorno de programación.

I. INTRODUCCIÓN

El objetivo de la revisión sistemática realizada consistió en explorar y analizar los estudios desarrollados a nivel global sobre las experiencias de enseñanza-aprendizaje formal de programación en instituciones educativas mediante el uso del entorno MIT *App Inventor for Android* (en adelante MIT AI).

MIT AI es una iniciativa de *Google Labs* y el *Massachusetts Institute of Technology* (MIT), que consiste en un entorno de programación visual para el desarrollo de aplicaciones funcionales para teléfonos inteligentes y tabletas, con sistema operativo Android y iOS, por personas de todas las edades; y para incentivar a los niños(as) y jóvenes a crear tecnología más allá de sólo consumirla. De acuerdo a sus creadores, la plataforma de desarrollo en línea, constituida por un lenguaje basado en bloques, favorece la creación de aplicaciones completas y de alto impacto en mucho menos tiempo que los entornos de programación tradicionales [1].

MIT AI cuenta con una interfaz de usuario que incluye el editor de diseño y el editor de bloques; el primero, permite arrastrar y soltar los elementos de la interfaz de usuario de la aplicación para posicionarlos; y el segundo, es el entorno en el que el usuario o “inventor” puede trazar de forma visual la lógica de la aplicación, usando bloques codificados por colores que se unen como piezas de rompecabezas para describir el programa [2].

La base fundamental del diseño de MIT AI consiste en el uso de “componentes” para reducir la complejidad de lidiar con interfaces de programación de aplicaciones específicas de la plataforma (APIs), y los detalles relacionados con la

administración del estado del hardware del dispositivo; y en el uso de “bloques” para eliminar la complejidad del lenguaje de programación subyacente. Esto permite al usuario pensar en el problema en cuestión en lugar de los detalles que generalmente se requieren para el desarrollo de aplicaciones en lenguajes basados en texto [2].

La plataforma de MIT AI es utilizada años tras año por miles de personas de muchos países para iniciarse en el mundo de la programación y crear aplicaciones que puedan brindar soluciones a problemas reales; es también utilizada por instituciones y docentes con el propósito de enseñar principios de programación a estudiantes de distintos niveles educativos[3]–[7]; para enseñar a programar a docentes de diferentes áreas del conocimiento, para mejorar los procesos de enseñanza-aprendizaje en su ámbito [8]–[11]; y también es examinada para el diseño de aplicaciones para móviles que faciliten el aprendizaje de gran diversidad de materias [12], [13].

La revisión de la literatura realizada, permitió examinar los experimentos y/o experiencias documentados sobre el uso de MIT AI para la enseñanza de programación, los métodos de enseñanza empleados, programas y contenidos, los principales hallazgos de los estudios sobre el uso del entorno, y la efectividad del aprendizaje de programación mediante MIT AI en los distintos niveles de educación; aspectos que pueden ser considerados por instituciones educativas y docentes para la planificación de los programas de enseñanza de ciencias de la computación y de programación en distintos niveles de formación.

II. METODOLOGÍA

A. Estrategia de búsqueda y criterios de selección

La búsqueda de la literatura fue realizada en las bases de estudios científicos y académicos pertinentes al tema de investigación a fin de garantizar la calidad de la información obtenida, es decir, fuentes reconocidas internacionalmente en los ámbitos de ciencias de la computación e informática, ingeniería y educación. Se utilizaron los siguientes motores de búsqueda especializados que garantizan la búsqueda de referencias de todas las regiones del mundo: *Association for Computing Machinery (ACM Digital Library*, <https://dl.acm.org/>), *Institute of Electrical and Electronics Engineers (IEEE Xplore® Digital Library*, <https://ieeexplore.ieee.org/>), *Microsoft Academic* (<https://academic.microsoft.com/>), *Education Resources Information Center - ERIC* (<https://eric.ed.gov/>), *Science Direct* (<https://www.sciencedirect.com/>) de Elsevier®, *Springer Link* (<https://link.springer.com/>) de Springer, Google

Digital Object Identifier (DOI):
<http://dx.doi.org/10.18687/LACCEI2020.1.1.49>
ISBN: 978-958-52071-4-1 ISSN: 2414-6390

Académico (<https://scholar.google.es/>), *Directory of Open Access Journal* – DOAJ (<https://doaj.org/>), *Bielefeld Academic Search Engine* – BASE (<https://www.base-search.net/>), SciELO – *Scientific Electronic Library Online* (<http://www.scielo.org/>), Redalyc – Red de Revistas Científicas de América Latina y el Caribe, España y Portugal (<https://www.redalyc.org/>), y Dialnet (<https://dialnet.unirioja.es/>).

La búsqueda se realizó bajo los siguientes términos: se empleó el descriptor “App Inventor”; un intervalo de tiempo desde el año 2010 -fecha de lanzamiento de MIT AI- hasta el año 2020; y tipos de documentos que correspondan a artículos publicados en revistas científicas (*Articles*) o memorias de eventos científicos (*Conference paper, Proceedings*), tesis de maestría y doctorados. Los términos de búsqueda dieron por resultado 1744 publicaciones que hacen referencia al descriptor utilizado, a partir de los cuales se descartaron todas aquellas publicaciones que no correspondían específicamente a la enseñanza de programación, es decir, no se consideraron los estudios que involucran el uso de MIT AI para la enseñanza de otras áreas de conocimiento, como idiomas, historia, ciencias, entre otros; y las publicaciones repetidas en varios buscadores.

Los criterios utilizados para la selección de los estudios para su posterior análisis, fueron: (1) contar con identificación (ISSN/ISBN) y ubicación (URL/DOI), (2) estudios tanto experimentales como no experimentales, (3) los participantes bajo estudio pueden pertenecer a cualquier nivel educativo, desde K-12 a universitario.

III. RESULTADOS Y ANÁLISIS DE LA INFORMACIÓN

A. Generalidades

Los resultados obtenidos a partir de la búsqueda se describen a continuación. Un total de 35 artículos fueron seleccionados bajo el cumplimiento de los criterios de búsqueda y selección establecidos (ver Tabla I).

TABLA I
NÚMERO DE TRABAJOS SELECCIONADOS POR MOTORES DE BÚSQUEDA ESPECIALIZADOS

Base de datos	Nº
ACM Digital Library	9
IEEEExplore Digital Library	8
Microsoft Academic	6
Scholar Google	8
Springer	2
ERIC	1
Dialnet	1
	35

Los estudios analizados tienen origen en países diversos, siendo los países con mayor número de publicaciones: USA (8) y Grecia (8), seguidos por Corea del Sur (4), Brasil (3), Turquía (2), Arabia Saudí (2), España (2) y seis publicaciones restantes entre Bulgaria, China, Colombia, Eslovaquia, Indonesia y Malasia. Por otra parte, a pesar de que el entorno MIT AI fue lanzado en el año 2010, se observa que el 54% de las publicaciones datan de los últimos años (2017 a 2019).

TABLA II
NÚMERO DE TRABAJOS SELECCIONADOS POR AÑO DE PUBLICACIÓN

Año de publicación	Nº
2012	2
2013	4
2014	5
2015	2
2016	3
2017	4
2018	8
2019	7

Respecto al nivel educativo de los estudios clasificados, la mayor parte de los mismos están enfocados a experiencias de enseñanza-aprendizaje para estudiantes de Secundaria, seguido por estudiantes universitarios y estudiantes de Primaria, como se aprecia en Fig. 1.

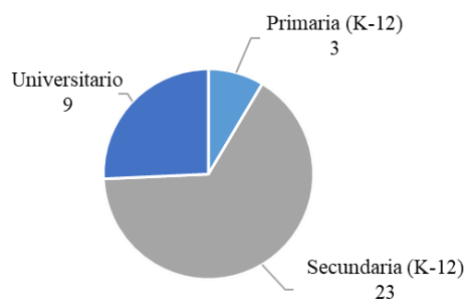


Fig. 1 Clasificación de los estudios de acuerdo al nivel educativo.

A fin de profundizar en el análisis de las publicaciones seleccionadas, se procedió a catalogarlas en tres grandes grupos, en función a su propósito principal. En este sentido, 32 de los 35 estudios tuvieron por objeto desarrollar experiencias o experimentos de enseñanza de programación y pensamiento computacional mediante MIT AI, en estudiantes de diferentes edades y antecedentes de uso de entornos para programar; mientras que 2 estudios presentan propuestas metodológicas para enseñar programación con MIT AI, y un único estudio cuyo propósito consistió en conocer la percepción de estudiantes y profesores respecto a experiencias con dicho entorno en encuentros formativos previos. También cabe destacar que 7 de los estudios aplicaron tanto MIT AI como otro entorno u lenguaje (Alice, Scratch, Android, MicroWorlds Pro, Java) para comparar resultados en el desempeño y la motivación de los participantes.

Por otra parte, en relación al conocimiento previo de los participantes, 24 de los 36 estudios contaron con participantes sin experiencia previa en programación en algún entorno o lenguaje, 9 de los estudios se realizó con grupos de participantes que sí contaban con alguna experiencia previa programando en algún entorno o lenguaje de programación, y 2 de los estudios se enfocaron en grupos mixtos de participantes con y sin experiencia previa.

B. Tipos de investigación aplicados

El tipo de investigación desarrollado en los estudios seleccionados es relevante. Las investigaciones de tipo experimental, implican un control riguroso de las variables bajo estudio para conocer sus efectos, aunque pueden dificultar la generalización de los hallazgos y la validez externa [14]. Por otra parte, los estudios no experimentales, brindan información importante desde el punto de vista cuantitativo y también cualitativo, a partir de la experiencia de los estudiantes y docentes que participan en el proceso de enseñanza-aprendizaje, las prácticas que puedan ser replicables y/o adaptadas en otras regiones en cuanto a la enseñanza de programación mediante el entorno MIT AI.

En este sentido, los autores consideran que incluir en el análisis todos los tipos de investigaciones halladas, da por resultado una revisión literaria más completa y que refleja la realidad de la actividad científica que se viene desarrollando en la comunidad académica. Así, de las 35 publicaciones seleccionadas bajo criterio, 7 estudios corresponden a un diseño de investigación experimental (experimento o pre-experimento), 27 estudios son de tipo no experimental, y 1 estudio que emplea ambos enfoques.

C. Estrategias de enseñanza

Un modelo de enseñanza implica la sistematización de metas, estrategias, materiales, contenidos, bajo un esquema diseñado para generar experiencias concretas de aprendizaje [15]. En la actualidad, los docentes tienen a su disposición incontables herramientas para aplicar en el proceso de enseñanza que dirigen, y con ello brindar una educación de calidad que logre los objetivos dispuestos. Los modelos de enseñanza aplicados en los estudios bajo análisis pueden identificarse como cuatro grandes grupos: *Problem-Based Learning*, *Project-Based Learning*, *Studio-Based Learning* y *Mission-Based Learning* (ver Tabla III), similares en su enfoque constructivista, pero que difieren en algunos aspectos metodológicos [16].

TABLA III
MODELOS DE ENSEÑANZA APLICADOS

Modelos de enseñanza principal	Referencia(s)	%
<i>Problem-Based Learning (PBL)</i>	[17], [18],	6%
<i>Studio-Based Learning (SBL)</i>	[19] ^a , [20] ^a , [21],	9%
<i>Project-Based Learning (PBL)</i>	[22], [6], [23], [24], [19] ^a , [20] ^a , [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46]	83%
<i>Mission-Based Learning (MBL)</i>	[47]	3%

^aEstudios que aplican más de un modelo

En más del ochenta por ciento de los estudios se optó por un modelo basado en proyectos, donde los docentes establecían previamente los objetivos de aprendizaje y suministraban contenido e instrucciones a los participantes, pero la implementación de los conceptos y el diseño de las aplicaciones

debía ser desarrollada por los estudiantes como proyectos individuales y/o grupales, para su posterior evaluación. De esta forma, los participantes tienen espacio para el análisis y el uso de los conceptos enseñados, así como para el desarrollo del pensamiento computacional, la iniciativa y la creatividad.

En cuanto a las estrategias didácticas más utilizadas para la transmisión de los contenidos a los estudiantes por parte de los docentes, destacan: las charlas breves, la creación de videos tutoriales, las instrucciones orales y escritas; con el propósito de dirigir la mayor cantidad de tiempo disponible en prácticas y menor proporción de tiempo a la explicación teórica, dejando atrás el modelo centrado en el docente. Entre las técnicas de aprendizaje empleadas, destacan: la programación grupal y en pares, e intercambio de roles, aula invertida, grupos de discusión.

D. Programas de contenidos

Un total de 18 estudios hacen referencia en la publicación a los conceptos de programación específicos que fueron enseñados, mientras que en 17 de las publicaciones no se especifican. No obstante, al tratarse, en gran mayoría, de estudios y experimentos dirigidos a alumnos sin conocimientos previos de programación y sin experiencia con otros entornos o lenguajes, los programas de clases estuvieron orientados, en principio, a familiarizar a los participantes con el entorno de MIT AI, para la subsiguiente enseñanza de conceptos básicos de programación, entre los que se contemplan pero no se limitan a: secuencia, programación dirigida por eventos, valores booleanos, número aleatorio, *if-then-else*, condiciones booleanas, variables, inicialización e incremento de variables, contadores, expresiones lógicas, bucles, listas, procedimientos, sensores del teléfono móvil y el objeto de reloj, uso y operación de subrutinas, diseño de interfaz de usuario.

Cabe señalar que el alcance y la profundización de los conceptos están relacionados, además, con la duración las experiencias, la frecuencia y tiempo dedicado a los encuentros, que difieren significativamente de unos estudios a otros.

En dicho sentido, la duración de los estudios realizados varía en un rango de que va desde 3 horas en la experiencia más corta, hasta un año escolar completo. En gran parte de las publicaciones no se declara con exactitud la duración de sus experiencias/experimentos en una unidad de tiempo específica, por ejemplo, en horas, que permitiese realizar un análisis comparativo con exactitud. Sin embargo, en función a la información recabada, se estimó el tiempo en semanas que durarían los estudios que declararon su duración en meses o en periodos académicos, como se aprecia en Fig. 2.

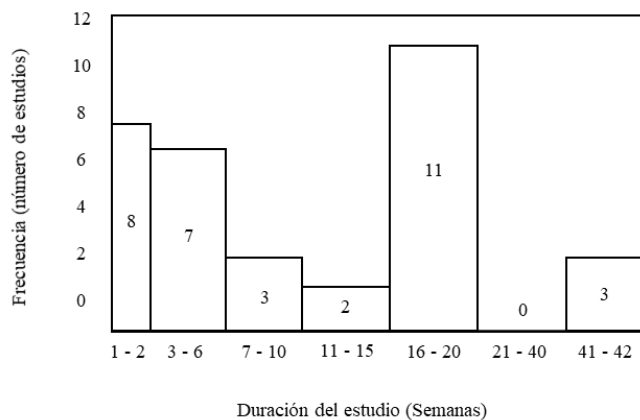


Fig. 2 Distribución de frecuencia de los estudios en función a la duración.

E. Métodos de evaluación

Las dimensiones de los resultados obtenidos comprenden aspectos motivacionales respecto a la programación, de desempeño sobre los conceptos y prácticas, de percepción sobre la experiencia en el entorno MIT AI y sobre los procedimientos de enseñanza-aprendizaje. Los estudios se enfocaron primordialmente en la evaluación por parte del docente, mientras que algunos exploraron la coevaluación y la autoevaluación.

Las estrategias para evaluar los resultados obtenidos en cuanto a la percepción de los participantes sobre el entorno MIT AI, incluyen: cuestionarios a estudiantes y a docentes, entrevistas individuales y grupales, discusión en grupos focales; y se consideraron variables como la facilidad/dificultad de uso, cualidades del medio, accesibilidad y practicidad.

Los métodos aplicados para evaluar el desempeño de los estudiantes en términos del aprendizaje efectivo de programación fueron diversos, e incluyeron: la evaluación de las aplicaciones y proyectos desarrollados por los estudiantes mediante rúbricas o mediante aplicaciones diseñadas para dicho propósito (*CodeMaster*); el uso indicadores como la cantidad de descargas de los proyectos en la plataforma web de MIT AI; aplicaciones del cuestionario de conocimiento de programación (QPK); mapas mentales elaborados por los estudiantes, y *tests* de conocimiento *pre* y *post* enseñanza. Los temas evaluados comprenden aquellos aprendidos en función a cada programa de estudio, como: conceptos y estructuras básicas de programación, habilidades de programación GUI, habilidades de desarrollo de aplicaciones, elementos de MIT AI utilizados, funcionalidad y apariencia, entre otros.

En el ámbito motivacional, se emplearon escalas reconocidas como el *Computer Attitude Questionnaire* (CAQ) o Escala de Actitud hacia la Computadora, en castellano, y el *Motivated Strategies for Learning Questionnaire* (MSLQ), conocido como Cuestionario de Motivación y Estrategias de Aprendizaje.

Sobre la periodicidad de las evaluaciones, varió de un estudio a otro, sin embargo, la mayor parte de los estudios aplicaron las evaluaciones antes y después de la experiencia

completa de aprendizaje (*pre* y *post*), para realizar comparaciones en el desempeño, la motivación, y poder detectar posibles cambios; otros estudios aplicaron *tests* al final de cada lección, a fin de examinar el nivel de conocimientos específicos adquiridos; y otros sólo al final de la experiencia, especialmente para evaluar la percepción sobre el entorno MIT AI y / o el proceso global de enseñanza-aprendizaje diseñado.

De los 35 estudios, sólo en el 40% se aplicaron métodos de evaluación de sus resultados mediante instrumentos que fueron comprobados estadísticamente como válidos y confiables, mientras que en el 60% de los estudios se aplicaron cuestionarios diseñados por los docentes para la experiencia, pero cuya validez y confiabilidad no fueron previamente estudiadas.

F. Principales hallazgos

Los hallazgos obtenidos a partir de las experiencias y experimentos realizados por la comunidad académica, muestran resultados satisfactorios respecto al proceso de enseñanza-aprendizaje de programación mediante MIT AI. Los estudios analizados, revelan efectos positivos a partir de la aplicación de dicho entorno, siempre en función a los objetivos que perseguía cada investigación en particular.

En términos generales, los estudios estuvieron orientados a examinar, entre otros: el impacto del lenguaje de programación por bloques -específicamente MIT AI- en la disposición de los estudiantes hacia la computación y la programación, mejoras en el rendimiento, la efectividad en el aprendizaje y en la aplicación de conceptos, el progreso percibido, el compromiso, la autoconfianza de los estudiantes, y la percepción general del proceso de aprendizaje por parte de los estudiantes y de los docentes.

1) *Hallazgos por niveles educativos*: MIT AI fue percibido en la mayoría de los estudios como una herramienta efectiva, útil y práctica, en todos los niveles educativos probados (primaria, secundaria y universitario). No se observan diferencias significativas en la percepción del entorno, más allá de los programas de estudio diseñados propiamente para cada nivel. Así, los estudios desarrollados en niveles de primaria, destacan el impacto de MIT AI en la capacidad de resolución de problemas de los niños y niñas, así como en su introducción en el mundo de la computación (ver Tabla IV). Igualmente, los estudios en niveles de secundaria, resaltaron la importancia de AI en mejorar la disposición e interés de los jóvenes hacia las ciencias de computación, presentar su aplicabilidad en el día a día, mejorar el autoaprendizaje, el trabajo en pares o en grupo, el diseño creativo, y en el desarrollo de habilidades informáticas y del pensamiento computacional (ver Tabla V).

Por otra parte, a nivel de estudios universitarios, los estudios hacen énfasis en factores pedagógicos, en la obtención de logros académicos significativos, así como de los aspectos motivacionales, tanto en estudiantes que inician su carrera en ciencias de la computación como en aquellos que no (ver Tabla VI). A pesar de que MIT AI es un entorno diseñado para el aprendizaje de programación de aplicaciones para móviles, a

cualquier edad, sólo tres estudios estuvieron enfocados a evaluar el impacto sobre el éxito académico en el aprendizaje de programación de aplicaciones para móviles [38], [29], [37], a diferencia del resto de estudios, que se enfocaron en el uso de

MIT AI para enseñar la lógica y conceptos básicos de programación.

TABLE IV
PRINCIPALES HALLAZGOS DE LOS ESTUDIOS DE ENSEÑANZA DE PROGRAMACIÓN REALIZADOS A NIVEL PRIMARIA

Autor	Objetivo(s)	Principales hallazgos
[39]	Enseñar conceptos informáticos mediante la programación de aplicaciones móviles de Android utilizando MIT AI.	*El uso de MIT AI y la planificación de las lecciones ayudaron a los estudiantes a evaluar el aprendizaje de contenidos como algo fácil, divertido. *El desarrollo en equipo estimula la discusión entre los estudiantes, siempre buscando la mejor manera de resolver los problemas o al tomar decisiones de diseño. Además, construir la aplicación sujeta a análisis por dos personas conduce a una mayor calidad y menos defectos.
[18]	Diseñar un plan de programación apropiado de MIT AI para estudiantes de primaria dotados y cambiar el modelo de aprendizaje de los tres pasos de Renzulli para estudiantes dotados.	*La educación de programación en MIT AI para estudiantes de primaria dotados afectó positivamente la capacidad de resolución de problemas, la autoeficacia y el logro del proceso de programación.
[43]	Enseñar cómo desarrollar aplicaciones para estudiantes de primaria con el programa MIT AI.	*Los estudiantes intentaron resolver el problema activamente con gran interés. Tenían una fuerte motivación y un sentido de logro de que la aplicación podría ser ampliamente conocida por muchos usuarios a través del dispositivo inteligente. *Mostraron un pensamiento activo para resolver el problema y las nuevas ideas.

TABLE V
PRINCIPALES HALLAZGOS DE LOS ESTUDIOS DE ENSEÑANZA DE PROGRAMACIÓN REALIZADOS A NIVEL SECUNDARIA

Autor	Objetivo(s)	Principales hallazgos
[26]	Proporcionar una plataforma para que los estudiantes diseñen aplicaciones móviles usando conceptos de informática y pensamiento computacional. Diseñar una aplicación para móviles por parte de los estudiantes usando MIT AI.	*Percepción de los estudiantes: el curso desarrolló habilidades y destrezas sobre este tema y el interés en la informática.
[22]	Recopilar las opiniones de los estudiantes y profesores sobre la implementación de MIT AI a través del uso de escenarios educativos.	*Según las opiniones de los estudiantes y los profesores, el contenido de los escenarios educativos se presentan con claridad y ayudan a aprender el contenido sin estar relacionado con ningún conocimiento previo de lenguajes de programación.
[33]	Investigar si las respuestas de los estudiantes a la escala CAS son diferenciados por la intervención MIT AI vs Scratch. Investigar si el conocimiento en programación de los estudiantes mejoró de acuerdo con su desempeño en la escala de evaluación después de aprender programación usando Scratch y MIT IA.	*Todos los estudiantes mejoraron su enfoque conductual y su conocimiento en programación, al final de la intervención. Además, los alumnos a quienes se les enseñó que el ambiente de MIT IA resultó en mejores resultados de aprendizaje y actitudes, percepciones y motivaciones más positivas para participar en la programación que aquellos a quienes se les enseñó usando Scratch.
[34]	Investigar cómo MIT AI puede integrarse en los cursos de ingeniería y ciencias de la computación de la escuela secundaria para fomentar el pensamiento computacional mediante el desarrollo de aplicaciones móviles.	*Los estudiantes que tenían experiencia en lenguajes de programación mejoraron más en conceptos de pensamiento computacional, como la recopilación de datos, la representación de datos y los algoritmos. *Mejora en las habilidades de resolución de problemas. *App Inventor ayuda a enseñar a personas que tienen algo de experiencia, así como a aquellos que tienen poca o ninguna experiencia en programación.
[24]	Conocer el desempeño de los estudiantes en pensamiento computacional. Aceptación de MIT AI en el curso introductorio de programación.	*Los proyectos fueron calificados por CodeMaster con valores entre 7 y 15, de un máximo de 45 puntos. *MIT AI como lenguaje de programación visual por bloques es atractivo para los estudiantes hasta la fase de diseño. Sin embargo, la implementación de los bloques (codificación) no es atractivo para la mayoría de los estudiantes.
[25]	Sugerir una estrategia para cerrar la brecha entre la programación visual en MIT AI 2 y Java para mediar una transferencia fluida de conocimiento de un contexto a otro.	*Para programación de aplicaciones móviles dentro de la educación secundaria superior, MIT AI complementado con código Java Generador es considerada la mejor opción. *En MIT AI, el progreso de los estudiantes es rápido. Después de comprender los principios fundamentales y obtener alguna experiencia de implementación al programar un conjunto de aplicaciones de muestra, los estudiantes son capaces de autoaprendizaje y prefieren trabajar en sus propios proyectos tomando su profesor como consultor. *Estudiantes con un interés más serio en la programación tienden a estar menos entusiasmados con el uso de MIT AI en un período a largo plazo.
[27]	Conocer cuál de los dos entornos de programación (MIT AI vs. Alice) contribuye más eficazmente al aprendizaje de los conceptos básicos de programación para programadores novatos.	*Desde un punto de vista tecnológico, MIT AI parece emerger como más poderoso para adopción en la educación secundaria en comparación con Alice. *A partir de un punto de vista pedagógico, ambos tienen sus puntos fuertes y débiles. MIT AI se considera una herramienta ideal para involucrar a los estudiantes jóvenes en comparación con Alice, ya que usa los dispositivos móviles de los estudiantes con fines de aprendizaje.

[31]	(Monitorear las percepciones de los estudiantes sobre la popularidad y la dificultad percibida de ciertas actividades / lecciones a través de la implementación de un curso de App Inventor en una escuela secundaria griega. Detectar cualquier diseño de curso o actividad / plan de lección e implementación factores que afectan las percepciones de los estudiantes. Evaluar su experiencia con MIT AI en contraste con su experiencia previa con MicroWorlds Pro y Scratch.	*Los estudiantes prefieren MIT AI sobre Scratch y MicroWorld Pro. *Los estudiantes que no estaban interesados en las lecciones de ciencias de la computación anteriores participaron con gran entusiasmo en las actividades de MIT AI. *Cuando se logra la autonomía, es el momento del aprendizaje basado en proyectos y el proceso de diseño creativo, donde los estudiantes desarrollan sus propias ideas y crean sus propias aplicaciones.
[37]	Diseñar y programar aplicaciones móviles con la herramienta MIT AI. Estimular la inteligencia lógica de los alumnos mediante la programación por bloques de desarrollo visual.	*Los alumnos valoran la importancia del trabajo en equipo y una gran mayoría ve de manera muy positiva la división de roles.
[42]	Dar conocimiento temprano de la tecnología de programación Android y redes de computadoras especialmente para estudiantes de la escuela vocacional. Diseñar un control de equipos electrónicos simples usando un teléfono inteligente programado usando MIT AI.	*Los alumnos obtienen nuevos conocimientos desde la aplicación de tecnología Android con dispositivos electrónica utilizando el MIT AI. *Los participantes pueden ensamblar equipos directamente electrónica a partir de un simple circuito Led hasta robots móviles y controlándolo usando el teléfono a través de bluetooth.
[44]	Utilizar la programación de juegos en las clases de programación para estudiantes de secundaria y analizar los efectos educativos.	*Aumento en las habilidades de pensamiento computacional, habilidades de pensamiento creativo y apoyo percibido para la programación después de las clases.
[35]	Enseñar lógica de programación de forma interactiva. Investigar las formas de mediación de MIT AI en el aprendizaje, considerando la interacción entre el alumno y la plataforma.	*Visualización positiva del uso de la plataforma como facilitador del proceso de enseñanza y aprendizaje de la lógica de programación como un primer contacto con el desarrollo de aplicaciones y la inserción de los programadores principiantes en el universo lógico de la programación.
[36]	Enseñanza de programación a estudiantes de escuelas públicas a través de la herramienta MIT AI.	*La propuesta fue bien recibida por los estudiantes. *Las actividades de extensión interinstitucional tienen un impacto positivo porque permite el intercambio de conocimientos y recursos para el beneficio de la comunidad.
[45]	Introducir algunos de los principales conceptos de programación en un forma interactiva utilizando MIT AI.	* *Aumento en la comprensión de los conceptos de programación presentados. *Los estudiantes han cambiado de opinión con respecto a su interés por la programación. *MIT AI apoya el aprendizaje activo a través de la programación diaria asignaciones y aplicando conceptos aprendidos en sus mini programas.
[46]	Usar dos entornos de programación basados en bloques (MIT AI y Scratch) como herramientas para facilitar el aprendizaje de programación para principiantes.	*Ambos entornos de programación para principiantes parecían ser plataformas atractivas para introducir conceptos fundamentales en la programación de computadoras y ambos parecen atractivos para estudiantes especializados y no especializados.
[48]	Implementar un programa de intervención educativa en la asignatura de Tecnología de ESO y analizar los efectos sobre la actitud, la motivación, la colaboración, la creatividad y el aprendizaje del lenguaje de programación.	*Los estudiantes presentan un mayor interés por los contenidos y por aprender cosas nuevas en comparación con el inicio. *Los participantes encontraron interesante trabajar con MIT AI y crear aplicaciones móviles. *Los alumnos consideran útil, importante e interesante el uso de MIT AI. *El uso de MIT AI eliminó las distracciones que ocurren al usar lenguajes de programación y ayudó a centrarse en la comprensión de los algoritmos.
[28]	Medir hasta qué punto es posible aprovechar el interés de los estudiantes en la tecnología móvil para fomentar su interés en la creación de aplicaciones móviles y alinear el pensamiento computacional e histórico.	*Los resultados sugieren que aunque los estudiantes inicialmente carecían experiencia en programación, estaban entusiasmados por aprender cómo para crear una aplicación móvil e indicaron interés en continuar la crear aplicaciones.
[20]	Explorar la posibilidad y la idoneidad de usar dispositivos móviles e MIT IA para motivar a los estudiantes a aprender sobre ciencias computacionales y conceptos fundamentales de programación.	*Un crecimiento estadísticamente significativo en el dominio del conocimiento de los estudiantes y en su compromiso. *Los estudiantes y los maestros percibieron positivamente la efectividad de los diseños pedagógicos sobre el uso de la programación de dispositivos móviles para fomentar el desarrollo de conceptos fundamentales de programación de los estudiantes a través de MIT AI.
[29]	Presentarles la idea básica de la programación. Cambiar la idea errónea sobre el campo de la computación. Cerrar la brecha educativa entre la academia y los estudiantes de secundaria en el campo de la informática. Atraer a estudiantes que se encuentran en una fase crítica de elegir su camino futuro.	*Las actividades de capacitación práctica con conferencias mínimas ayudaron a acelerar el proceso de aprendizaje. *Los estudiantes al final del campamento tenían dos proyectos ejecutándose en su teléfono móvil Android: <i>Intelligence Games Collection</i> y sus propios proyectos desarrollados. Estos proyectos alcanzaron el objetivo principal del campamento que era: aplicar y diseñar conceptos de computación utilizando desarrollo de aplicaciones móviles, estos conceptos incluyen: bucles, listas, declaración de decisiones, animación y generación de GUI (<i>Graphical User Interface</i>).
[17]	Investigar las diferencias en la motivación de los estudiantes en aprender programación haciendo uso de Scratch y MIT AI en la enseñanza.	* Se descubrió que la orientación hacia el objetivo intrínseco, el valor de la tarea, el control de las creencias de aprendizaje y la autoeficacia aumentaron utilizando estos dos entornos de programación de aprendizaje de nivel de entrada desde el principio hasta la mitad del curso. No se encontró ningún efecto sobre la motivación extrínseca.
[40]	Conocer las experiencias de los estudiantes del uso de MIT AI en el salón de clases.	*MIT AI podría hacer que la programación sea agradable y accesible para principiantes, lo que permite a los estudiantes desarrollar habilidades informáticas y pensamiento computacional.
[6]	Enseñar a programar aplicaciones para Android.	*Iniciar la instrucción con el lenguaje de bloque y luego mostrar la asignación directa a una versión equivalente de Java, ayudó a los estudiantes a comprender el desarrollo de aplicaciones en Java.

[23]	Desarrollar un modelo de aprendizaje basado en proyecto mediante MIT AI. Presentar el pensamiento computacional y programación, y estimular el interés en la informática.	*Hubo un ligero aumento en la disposición favorable hacia la informática. *Los estudiantes de secundaria pueden aprender muy rápidamente las ideas computacionales básicas a través de MIT IA y aplicarlos de manera significativa para crear aplicaciones. * MIT AI es una herramienta efectiva para presentar a los programadores novatos programación.
------	--	--

TABLA IV
PRINCIPALES HALLAZGOS DE LOS ESTUDIOS DE ENSEÑANZA DE PROGRAMACIÓN REALIZADOS A NIVEL UNIVERSITARIO

Autor	Objetivo(s)	Principales hallazgos
[19]	Conocer la respuesta a la implementación de dos factores pedagógicos: el uso de IA y el aprendizaje basado en estudio (<i>Studio-Based Learning</i>).	*Respuesta positiva a los factores pedagógicos, y actitudes positivas hacia la materia. *Logros académicos significativos. *El enfoque experimental tiene un efecto positivo en motivación, creatividad, logro y actitud de los estudiantes hacia la materia.
[47]	Promover el pensamiento computacional en los estudiantes con AI como primer lenguaje de computación.	*Los estudiantes perciben el proceso de aprendizaje como más fácil, interesante, con alto grado de aceptación y obtienen resultados satisfactorios. *El 90% de los estudiantes completan el experimento en el acto y corren sin problemas en su teléfono móvil.
[21]	Hacer uso de AI y la adopción de SBL como la principal metodología de enseñanza en programación. Mejorar la comprensión como educadores de la motivación y percepción de los estudiantes que no son estudiantes de informática en los cursos introductorios de CS _a .	*El enfoque experimental tuvo un efecto positivo en la motivación, el rendimiento y la actitud de los estudiantes hacia CS _a . *La mayoría de los participantes lo hicieron muy bien y mostraron creatividad; ninguno reprobó el curso.
[49]	Conocer el punto de vista de los estudiantes sobre el uso de MIT AI en distintos niveles educativos.	*Independientemente del número de los lenguajes de programación que conocen, los estudiantes creen que este ambiente debe usarse para enseñar: primero, en un grado de licenciatura, seguido a un nivel de Maestría y, por último, en la Secundaria. *Cuanto más lenguajes de programación saben los estudiantes, cuanto más altas son las calificaciones que dan a la anterior declaración.
[30]	Proponer un currículum a corto plazo para enseñar programación de computadoras usando MIT AI a los estudiantes de primer año que se especializan en informática como el primer lenguaje de programación estudio, que puede alentar su interés en la programación.	El plan de estudios propuesto puede contribuir a aumentar el interés en la programación de computadoras e incluso la confianza en sí mismo en el desarrollo de aplicaciones para <i>smartphone</i> .
[50]	Conocer la percepción de los estudiantes del curso de Desarrollo de Aplicaciones, con una introducción al curso de programación como requisito previo, respecto al uso de MIT AI para la enseñanza.	*Las respuestas positivas respaldan la práctica de usar App Inventor para enseñar no solo a programadores principiantes, sino también a programadores más experimentados. *App Inventor se puede utilizar para apoyar la enseñanza de conceptos informáticos más avanzados.
[32]	Investigar los efectos (en éxito académico, y opiniones y preferencias profesionales en el contexto de la educación en programación) a corto y largo plazo del uso de BBVPL _b como una herramienta de segundo nivel al enseñar programación a futuros maestros de TIC de K12.	*Los estudiantes novatos se desempeñaron significativamente mejor usando el lenguaje basado en bloques y los estudiantes experimentados se desempeñaron significativamente mejor usando el lenguaje basado en texto. *Los estudiantes con experiencia han mostrado dificultades para transferir sus conocimientos existentes de programación al entorno basado en bloques de App Inventor.
[38]	Evaluar el impacto sobre el éxito académico del uso de MIT AI como instrumento educativo para enseñar conceptos de desarrollo de aplicaciones GUI orientadas a objetos, a estudiantes de CS2 con cursos previos de conceptos de programación con C#. Investigar si el éxito previo en programación en ambientes basados en bloques durante la secundaria y CS1 tiene impacto en el éxito en CS2.	*Los estudiantes con experiencia en el desarrollo de aplicaciones GUI desde la secundaria obtienen mejores resultados en CS2 utilizando el SBL _c ; los que no tienen experiencia desde la secundaria obtienen mejores resultados en CS2 utilizando el BBVPL _b . *Los estudiantes que se acostumbraron a desarrollar aplicaciones utilizando un lenguaje / entorno de programación determinado tuvieron dificultades para adaptarse a un lenguaje / entorno nuevo y radical. *La educación de programación de la escuela secundaria afecta positivamente el éxito en educación de programación de pregrado.
[41]	Experimentar con la efectividad MIT AI como herramienta para desarrollar aplicaciones para móvil.	*Los estudiantes del grupo que usaron MIT AI como herramienta de desarrollo fueron capaces de completar más proyectos que el grupo que desarrolló sus aplicaciones usando <i>Android Studio</i> . * MIT AI es mucho más fácil, más rápido y capaz de ayudar el aprendizaje del alumno poco calificado en comparación con el grupo que aprendió usando AS.

^aCS: Computer Science. ^bBBVPL: Blocks-Based Visual Programming Language. ^cSBL: syntax-based object-oriented programming language.

2) *Hallazgos según la experiencia previa en programación*: Los estudios revelan que MIT AI es una herramienta efectiva para presentar a los estudiantes novatos las nociones sobre programación, así como las ideas computacionales básicas y su uso significativo en la creación de aplicaciones. También puede ser un medio efectivo para aumentar la disposición y motivación inicial de los estudiantes principiantes, es decir, estudiantes sin ninguna experiencia previa con la programación o con poca experiencia en otro

entorno basado en lenguaje visual por bloques, hacia la práctica. Así, las investigaciones apuntan a que los principiantes se desempeñan mejor utilizando lenguaje basado en bloques que en lenguajes basados en texto, y a que el aprendizaje desde niveles educativos previos a la universidad mejora el desempeño de los estudiantes en niveles más avanzados.

No obstante, la efectividad de AI es cuestionada por algunos investigadores cuando los estudiantes poseen conocimientos de lenguajes basados en texto, quienes al utilizar

MIT AI, en algunos casos, presentan disminución de su rendimiento, posiblemente atribuidos a problemas de motivación –al querer aprender lenguajes que les sean más útiles a futuro-, o a dificultades para transferir los conocimientos existentes al entorno basado en bloques de MIT AI, lo cual a su vez podría tener su origen en que el estudiante ha aprendido un lenguaje de memoria, más allá de la propia lógica de la programación [32], [38].

Lo descrito anteriormente contrasta con hallazgos en otros estudios, que respaldan la enseñanza de conceptos más avanzados y del uso de MIT AI también en programadores más experimentados y especializados, como en [49] y [46], o incluso, que MIT AI debe emplearse para enseñar, con preferencia, en niveles universitarios [50].

Respecto a experiencias de transición desde MIT AI hacia un lenguaje basado en texto, los estudios recomiendan iniciar la instrucción con el lenguaje de bloque y luego mostrar la asignación directa a una versión equivalente en el lenguaje que se desea enseñar, mediante aplicaciones como *App Inventor Java Bridge*, o generadores de código, para ayudar a los estudiantes a aprender sobre la lógica y principios fundamentales de la programación, mientras que comprenden el desarrollo de aplicaciones en otro lenguaje [6], [25].

3) *Comparación de MIT AI con otros entornos:* En los estudios donde se llevaron a cabo experiencias/experimentos de comparación entre MIT AI y otros lenguajes y entornos, los resultados cuantitativos y cualitativos apuntan a una preferencia por MIT AI en gran parte de los casos (ver Tabla VII).

4) *Ventajas identificadas en el uso de MIT App Inventor:* Durante las experiencias desarrolladas con MIT AI, se identificaron ventajas y desventajas en el uso en términos pedagógicos y operativos, sin embargo, considerando la alta frecuencia de actualizaciones que se realizan al entorno (la última registrada en diciembre de 2019, durante la realización de la presente revisión de literatura), sólo se consideraron las ventajas y/o desventajas descritas en la literatura del último año (ver Tabla VIII).

TABLA VII
PRINCIPALES HALLAZGOS DE LA COMPARACIÓN DE ENTORNOS DE PROGRAMACIÓN CON MIT AI

Autor	Entornos	Hallazgos
[41]	MIT AI – Android Studio	Los estudiantes del grupo que usaron IA fueron capaces de completar más proyectos que el grupo que usó AS. MIT AI es mucho más fácil, más rápido y capaz de ayudar el aprendizaje del alumno poco calificado en comparación con AS.
[33]	MIT AI - Scratch	Los alumnos a quienes se les enseñó en MIT AI resultó en mejores resultados de aprendizaje y actitudes, percepciones y motivaciones que aquellos a quienes se les enseñó usando Scratch.
[31]	MIT AI - Scratch - MicroWorld Pro.	Los estudiantes prefieren MIT AI sobre Scratch y MicroWorld Pro. Los estudiantes que no estaban interesados en las lecciones de ciencias de la computación anteriores participaron con gran entusiasmo en las actividades de MIT AI.
[27]	MIT AI - Alice	Desde un punto de vista tecnológico, MIT AI parece emerger como más poderoso para adopción

		en la educación secundaria en comparación con Alice. MIT AI se considera una herramienta ideal para involucrar a los estudiantes jóvenes en comparación con Alice, ya que usa los dispositivos móviles de los estudiantes con fines de aprendizaje.
[46]	MIT AI - Scratch	Ambos entornos de programación parecían ser plataformas atractivas para introducir conceptos fundamentales en la programación de computadoras y ambos parecen atractivos para estudiantes especializados y no especializados.
[17]	MIT AI - Scratch	La orientación hacia el objetivo intrínseco, el valor de la tarea, el control de las creencias de aprendizaje y la autoeficacia aumentaron utilizando ambos entornos de programación. No se encontró ningún efecto sobre la motivación extrínseca.

TABLA VIII
PRINCIPALES VENTAJAS DESCRITAS POR EL USO DE MIT AI

Ventajas descritas	Autor
MIT AI admite un emulador que proporciona un teléfono inteligente virtual con la capacidad de ejecutar la aplicación producida en tiempo real. Esta característica es muy útil cuando surgen problemas de conectividad o legislación sobre el uso de teléfonos inteligentes dentro de la configuración del aula.	[22]
En MIT AI la programación es fácil, la arquitectura basada en web es una muy buena solución. El entorno ofrece suficientes bloques para desarrollar la lógica de las aplicaciones móviles.	[49]
MIT AI es eficaz como herramienta de codificación para promover objetivos cognitivos, de colaboración, de trabajo en equipo y de autogestión de base amplia.	[33]

IV. REFLEXIONES FINALES

Los retos que existen en el proceso de enseñanza-aprendizaje de programación a estudiantes novatos, tales como el desarrollo del pensamiento computacional, del pensamiento crítico y la resolución de problemas, el aprendizaje del lenguaje a emplear, formular secuencias de instrucciones y algoritmos, entre otros, siguen presentes independientemente del nivel educativo al que se enseñe. No obstante, actualmente existen recursos disponibles que han mostrado ser eficaces para hacer frente a dicho retos y facilitar el proceso tanto a los estudiantes como a los docentes, como lo son los entornos para programación mediante lenguaje visual por bloques.

Sin embargo, el sólo hecho de cambiar de entorno o lenguaje no es suficiente para mejorar la enseñanza de programación en las instituciones educativas, la implementación de nuevos entornos debe ir acompañada de programas de enseñanza cuidadosamente planificados, en función a los objetivos que se persigan, a los contenidos que se deseen impartir, y al público objetivo; las experiencias y experimentos analizados en la revisión literaria, fueron exitosas cuando se consideraron las variables mencionadas.

En este sentido, cabe resaltar la efectividad de MIT *App Inventor* para enseñar los principios básicos de la programación a estudiantes sin experiencia previa en la materia, bajo clases estructuradas donde el docente guíe el proceso de enseñanza a partir del planteamiento de proyectos, y que sean los alumnos

quienes desplieguen su aprendizaje a través del desarrollo de los mismos.

MIT *App Inventor* presenta una ventaja sobre otros entornos de programación visual por bloques para principiantes, y es el hecho de que los usuarios pueden apreciar la aplicabilidad de sus creaciones, es decir, disfrutar de una aplicación en funcionamiento en el propio teléfono inteligente o tableta una vez finalizado el proyecto, lo cual incrementa la motivación de los estudiantes para continuar el aprendizaje.

La revisión de literatura realizada, da un claro panorama sobre los esfuerzos de la comunidad académica a nivel mundial para mejorar, cada vez más, el proceso de enseñanza-aprendizaje en ciencias de computación y en programación, materia tan importante en la actualidad, y que continuará siendo en el futuro. Así, resalta el interés de docentes e investigadores de países con diversos antecedentes y experiencias en la enseñanza de programación, en estudiar y diseñar experiencias educativas innovadoras y de calidad, para distintos niveles educativos.

Igualmente, destaca el gran interés de los investigadores en conocer los factores que afectan la motivación de los estudiantes y su percepción sobre la programación, para plantear mejoras académicas; no obstante, también destaca el hecho de que muchos de los estudios analizados no fueron metodológicamente diseñados para ser válidos y confiables desde un punto de vista estadístico, sin embargo, si presentan un importante aporte cuantitativo y cualitativo obtenido a partir de las experiencias.

Finalmente, ya que los profesionales de la docencia cuentan con la facilidad y la motivación brindada por las herramientas como MIT AI para la enseñanza de programación, es de considerar para futuras investigaciones la profundización en el diseño e implementación de programas de enseñanza que sean innovadores y a su vez integrados con la formación en otras áreas del conocimiento, para que pueda percibirse su practicidad. Así mismo, se puede profundizar en estudios orientados a mejorar la transición de los estudiantes de la programación en bloques a los lenguajes basados en texto, para aquellos estudiantes que necesiten progresar a nivel académico. No se debe dejar de lado la importancia de estudiar la eficacia de las innovaciones desarrolladas tanto a nivel de herramientas como a nivel pedagógico.

REFERENCIAS

[1] Massachusetts Institute of Technology (MIT), «MIT App Inventor - About Us». [En línea]. Disponible en: <https://appinventor.mit.edu/about-us>. [Accedido: 27-ene-2020].

[2] S. Kong y H. Abelson, *Computational Thinking Education*. SpringerOpen, 2019.

[3] P. Musmarra, «Teaching mobile development with App Inventor and pair programming», en *16th International Conference e-Society 2018*, 2018, pp. 178-184.

[4] P. Rosales y I. Vasconcelos, «O Pensamento Computacional associado ao MIT App Inventor na aprendizagem dos alunos». São Paulo, 2017.

[5] A. Soares, «Reflections on Teaching App Inventor for Non-Beginner Programmers: Issues, Challenges and Opportunities», *Inf.*

Syst. Educ. J., vol. 12, n.º 4, 2014.

[6] A. Wagner, J. Gray, J. Corley, y D. Wolber, «Using App Inventor in a K-12 Summer Camp», en *SIGCSE '13: Proceeding of the 44th ACM technical symposium on Computer science education*, 2013, n.º January 2015, pp. 621-626.

[7] S. Grover y R. Pea, «Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students», en *SIGCSE 2013 - Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 2013, pp. 723-728.

[8] A. Ortega García, A. Ruiz-Martínez, y R. Valencia-García, «Using App Inventor for creating apps to support m-learning experiences: A case study», *Comput. Appl. Eng. Educ.*, vol. 26, n.º 3, pp. 431-448, 2018.

[9] D. Ramos, P. B. De Moura Oliveira, y E. J. Solteiro Pires, «App inventor as a tool to reach students», en *ACM International Conference Proceeding Series*, 2015, pp. 311-316.

[10] Y. C. Hsu y Y. H. Ching, «Mobile app design for teaching and learning: Educators' experiences in an online graduate course», *Int. Rev. Res. Open Distance Learn.*, vol. 14, n.º 4, pp. 117-139, 2013.

[11] J. Liu, C. H. Lin, P. Potter, E. P. Hasson, Z. D. Barnett, y M. Singleton, «Going mobile with app inventor for android: A one-week computing workshop for K-12 teachers», *SIGCSE 2013 - Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 433-438, 2013.

[12] B. MacKellar, «App inventor for android in a healthcare IT course», en *SIGITE '12 - Proceedings of the ACM Special Interest Group for Information Technology Education Conference*, 2012, pp. 245-250.

[13] D. Tang, «Empowering Novices to Understand and Use Machine Learning With Personalized Image Classification Models, Intuitive Analysis Tools, and MIT App Inventor», Massachusetts Institute of Technology, 2019.

[14] R. Hernández Sampieri, C. Fernández Collado, y P. Baptista Lucio, *Metodología de la investigación*, 6a edición. McGRAW-HILL, 2014.

[15] L. Morán, «Criterios para análisis comparativo de modelos y diseños educativos», *Educ. y Educ.*, vol. 11, n.º 2, pp. 1-10, 2008.

[16] J. Ineiche, «Problem based learning, project based learning e studio based learning: Três etapas na explicitação da originalidade do dispositivo pedagógico do atelier de arquitetura», en *PROJETAR Originalidade, criatividade e inovacao no projecto contemporaneo: ensino, pesquisa e prática*, 2015, pp. 1-16.

[17] S. Nikou y A. Economides, «Transition in Student Motivation during a Scratch and an App Inventor course», en *2014 IEEE Global Engineering Education Conference (EDUCON)*, 2014, n.º April.

[18] H. Seo y Y. Lee, «The Effects of Learning App Inventor Programming Education Plan for Gifted Elementary Students», *J. Korean Assoc. Inf. Educ. Vol.*, vol. 21, n.º 1, pp. 13-22, 2017.

[19] K. Ahmad y P. Gestwicki, «Studio-based learning and app inventor for android in an introductory CS course for non-majors», *SIGCSE 2013 - Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, pp. 287-292, 2013.

[20] G. Chatzinikolakis y S. Papadakis, «Motivating K-12 students learning fundamental Computer Science concepts with App Inventor», *Proc. 2014 Int. Conf. Interact. Mob. Commun. Technol. Learn. IMCL 2014*, n.º Imcl, pp. 152-159, 2015.

[21] K. Nasser, «Measuring the impact of app inventor for android and studio-based learning in an introductory computer science course for non-majors», Ball State University, Muncie, 2012.

[22] E. Seralidou y C. Douligeris, «Learning with the AppInventor programming software through the use of structured educational scenarios in secondary education in Greece», *Educ. Inf. Technol.*, vol. 24, n.º 4, pp. 2243-2281, 2019.

[23] K. Roy, «App inventor for android: Report from a summer camp», en *SIGCSE '12 - Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 2012, pp. 283-288.

[24] J. A. Martínez-Valdés y N. A. Martínez-Ijajá, «An experience with the App Inventor in CS0 for the development of the STEM

- didactics», *ACM Int. Conf. Proceeding Ser.*, pp. 51-56, 2018.
- [25] T. Tóth y V. Michaličkova, «From App Inventor to Java: A Strategy for Mediating the Transition», *ICETA 2018 - 16th IEEE Int. Conf. Emerg. eLearning Technol. Appl. Proc.*, pp. 591-596, 2018.
- [26] B. Z. Barkana y W. McDonough, «AP Computer Science Principles: Designing the hourly App in MIT App Inventor», *2019 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2019*, pp. 1-6, 2019.
- [27] S. Papadakis y V. Orfanakis, «Comparing novice programming environments for use in secondary education : App Inventor for Android vs . Alice Comparing novice programming environments for use in secondary education : App Inventor for Android vs . Alice Stamatiou Papadakis * Vasileios Orf», *Int. J. Technol. Enhanc. Learn.*, vol. 10, n.º January, pp. 44-72, 2018.
- [28] Y. Jimenez y C. Gardner, «Using App Inventor & History as a Gateway to Engage African American Students in Computer Science», en *2015 Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 2015, n.º June 2018.
- [29] S. Alhumoud, H. Al-Khalifa, M. Al-Razgan, y A. Alfaries, «Using App Inventor and LEGO mindstorm NXT in a Summer Camp to attract High School Girls to Computing Fields», en *2014 IEEE Global Engineering Education Conference (EDUCON)*, 2014.
- [30] B. Kim, «Computer Programming Education using App Inventor for Android», *Rev. la Soc. Inf. y Comun. Corea*, vol. 10, n.º 21, pp. 467-472, 2012.
- [31] G. Panselinas, E. Fragkoulaki, N. Angelidakis, S. Papadakis, E. Tzagkarakis, y V. Manassakis, «Monitoring Students' Perceptions in an App Inventor School Course», *EJERS Eur. J. Eng. Res. Sci.*, vol. CIE 2017, pp. 5-10, 2018.
- [32] C. Mihci y O. Nesrin, «Teaching GUI-Programming Concepts to Prospective K12 ICT Teachers: MIT App Inventor as an Alternative to Text-Based Languages», *Int. J. Res. Educ. Sci.*, vol. 3, n.º 2, pp. 543-559, 2017.
- [33] S. J. Papadakis, «Evaluating the efficiency of two programming environments in shaping novices' attitudes, perceptions, beliefs and knowledge in programming: a comparison between Scratch and App Inventor», *Int. J. Teach. Case Stud.*, vol. 10, n.º 1, pp. 31-52, 2019.
- [34] R. Faqih, «Teaching young people computational thinking using MIT App Inventor», Kansas State University, United States, 2019.
- [35] N. Leôncio, C. Calaça, P. Rogério, y R. Ferreira, «Programação em blocos com o Mit App Inventor: Um relato de experiência com alunos do ensino médio», en *VI Congresso Brasileiro de Informática na Educação*, 2017, n.º Cbie, pp. 1159-1163.
- [36] A. Bordin, C. Gonçalves, N. Vargas, J. Gonçalves, y P. Silas, «Ensino de programação para o ensino médio com App Inventor: Um relato de experiência extensionista através da colaboração interinstitucional», en *VI Congresso Brasileiro de Informática na Educação*, 2017, pp. 1179-1183.
- [37] I. Sanz, «Propuesta de aprendizaje para estudiantes del siglo XXI: Proyecto colaborativo basado en App Inventor», Universidad Pública de Navarra, España, 2018.
- [38] C. Mihci y N. Ozdener, «Programming education with a blocks-based visual language for mobile application», *10th Int. Conf. Mob. Learn. 2014 Program.*, pp. 149-156, 2014.
- [39] R. Filho, C. Wangenheim, J. Hauck, y G. Madeiros, «Desenvolvimento de uma unidade instrucional para ensinar o desenvolvimento de apps no ensino fundamental com o app inventor», *Departamento de Informática e Estatística*. Universidade Federal de Santa Catarina, Florianópolis, p. 12, 2019.
- [40] K. Perdikuri, «Students' experiences from the use of MIT app inventor in classroom», *ACM Int. Conf. Proceeding Ser.*, vol. 02-04-Octo, 2014.
- [41] N. A. Mohamed Mokmin, «The evaluation of tools for android application programming», *ACM Int. Conf. Proceeding Ser.*, vol. Part F1483, pp. 5-9, 2019.
- [42] K. Exaudi, Rendyansyah, A. Perdana, A. Fali, S. Siswanti, y M. Ajran, «Peningkatan Skill Programming Bagi Siswa Siswi SMKN 1 Indralaya Selatan Melalui Pelatihan Android Menggunakan App Inventor untuk Mengendalikan Peralatan Elektronika», en *Prosiding Annual Research Seminar 2018 Computer Science and ICT ISBN*, 2018, vol. 4, n.º 1, pp. 978-979.
- [43] H. Rim, «Android App. Implementation Teaching using App. Inventor for Elementary school students», *J. Korea Multimed. Soc.*, vol. 16, n.º 12, pp. 1495-1507, 2013.
- [44] S. Kim y S. Kim, «Middle-School Programming Classes Utilizing App Inventor Game Creation and the Analysis of their Educational Outcomes», *J. Korea Game Soc. 2018*, vol. 18, n.º 3, pp. 49-60, 2018.
- [45] S. A. A-Ghamdi, N. A. Al-Rajhi, N. M. Al-Onaizi, y H. S. Al-Khalifa, «Using app inventor 2 in a summer programming workshop: Improvements over previous years», *IEEE Glob. Eng. Educ. Conf. EDUCON*, vol. 10-13-April, n.º April, pp. 383-388, 2016.
- [46] N. Zaranis, V. Orfanakis, S. J. Papadakis, y M. Kalogiannakis, «Using Scratch and App Inventor for teaching introductory programming in Secondary Education . A case study . Using Scratch and App Inventor for teaching introductory programming in secondary education . A case study Stamatiou Papadakis * Michail Kalogiann», *Int. J. Technol. Enhanc. Learn.*, vol. 8, n.º 3/4, pp. 217-233, 2016.
- [47] X. Tang, «Teaching Experience for Non-computer Major App Inventor Program Design Course Xianchuan», *ICST Inst. Comput. Sci. Soc. Informatics Telecommun. Eng.*, vol. 243, pp. 232-236, 2018.
- [48] S. García y R. Roig, «Studio-Based Learning , una metodología al servicio de la enseñanza de los lenguajes de programación», en *Congreso Internacional de Filosofía de la Educación CEMOrOc-Feusp / IJI-Univ do Porto*, 2018, pp. 67-78.
- [49] T. S. Georgiev, «Students' Viewpoint about Using MIT App Inventor in Education», en *Call for Papers: 21-st International Conference on Computer Systems and Technologies CompSysTech*, 2020.
- [50] A. Soares y N. Martin, «Teaching Non-Beginner Programmers with App Inventor: Survey Results and Implications», en *Information Systems Education Journal (ISEDJ)*, 2015, vol. 13, n.º 5.