

**FACTORES DETERMINANTES PARA EL  
APRENDIZAJE DE PROGRAMACIÓN DE COMPUTADORAS**

Propuesta de Disertación presentada al  
Departamento de Estudios Graduados  
Facultad de Educación  
Universidad de Puerto Rico Recinto de Río Piedras  
como requisito parcial para  
obtener el grado de Doctor en Educación

Por:

**Roberto Martínez Mejía**  
© Derechos reservados, 2021

Disertación presentada como requisito parcial para obtener  
el grado de Doctor en Educación

**FACTORES DETERMINANTES PARA EL APRENDIZAJE DE  
PROGRAMACIÓN DE COMPUTADORAS**

**ROBERTO D. MARTÍNEZ MEJÍA**

Magíster en Educación, Universidad Autónoma del Caribe, 2012  
Ciencias de la Educación, Lenguas Modernas Universidad del Atlántico, 2006  
Ingeniero de Sistemas, Universidad Simón Bolívar, 2005

Aprobada el 10 de mayo del 2021 por el Comité de Disertación:

---

Edwin J. Martínez Hernández, Ed. D.  
Director de Disertación

---

Mirla Méndez Solano, Ed. D.  
Miembro del Comité

---

Julio E. Rodríguez Torres, Ph. D.  
Miembro del Comité

## DEDICATORIA

Agradezco a Dios por la vida, la salud y la familia...

Dedico este trabajo a mi madre Cádiz Mejía Viloría quien me guio a lo largo de su vida, con sus sabios consejos y el buen ejemplo, me enseñó a perseverar y a confiar en Dios.

Gracias por todos los esfuerzos y sacrificios que hiciste para que tuviera una buena educación. Gracias por tus oraciones y bendiciones, por acompañarme en todos los momentos difíciles y ser mi apoyo e inspiración para lograr esta gran meta. Madre querida, que Dios te guarde y te tenga en su gloria.

Gracias a mi padre William Martínez Vivero porque me enseñaste grandes valores para la vida como la paciencia, responsabilidad y resiliencia. Gracias por tus palabras y todo el apoyo que siempre me brindas.

Gracias a mi esposa Brenda, por ser mi compañera de estudios, por su apoyo incondicional en todo momento. Gracias por ayudarme a canalizar y creer en mis ideas. Gracias por iniciar este sueño a mi lado y confiar en lo que un día fueron metas lejanas, hoy ambos lo hemos logrado.

A mis hijos Nathalie e Isaac, quienes me prestaron mucho de su tiempo para realizar estos estudios. Ustedes han sido mi estímulo para alcanzar grandes logros. Gracias por tener siempre una sonrisa para alegrar mis días.

A mis hermanos Hugo e Ivonne, con quienes crecí y aprendí desde pequeño a ser unidos, a tener confianza y a compartir en familia. Siempre aprendo mucho de ustedes, gracias por apoyarme desde el inicio hasta el fin de esta meta, les debo toda mi gratitud.

## RECONOCIMIENTOS

Al director de disertación, Doctor Edwin J. Martínez Hernández

Gracias por dirigir este trabajo en tiempos tan difíciles de pandemia. Su disposición, entrega, asesoramiento y calidad humana fueron vitales para terminar esta disertación con éxito.

A los miembros del comité Dr. Julio E. Rodríguez y Dra. Mirla Méndez gracias por los consejos y el apoyo que me brindaron. La vasta experiencia que poseen en sus distintas disciplinas fue de gran aporte para realizar esta disertación. Ustedes son excelentes seres humanos con la gran vocación de enseñar.

A los directores del Departamento de Ciencia de Cómputos Dr. José Ortiz y Dr. Humberto Ortiz por colaborar y abrir las puertas de este Departamento para realizar mi investigación.

A las consejeras del Departamento de Estudios Graduados de la Facultad de Educación Dra. Mercedes Salichs y Dra. Lis Cruz gracias por estar siempre presente en los momentos difíciles, valoro su compromiso y todo el apoyo que me brindaron.

A los estudiantes del Departamento de Ciencia de Cómputos, que de manera voluntaria participaron respondiendo el cuestionario de recopilación de información. Sus respuestas fueron de gran valía para los resultados de esta investigación.

## **RESUMEN DE LA DISERTACIÓN**

### **FACTORES DETERMINANTES PARA EL APRENDIZAJE DE PROGRAMACIÓN DE COMPUTADORAS**

Roberto D. Martínez Mejía

Director de disertación: Edwin J. Martínez Hernández, Ed. D.

El presente estudio investigó, sobre los factores determinantes en el aprendizaje de la programación de computadoras. Respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras. Además, factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos, desde la perspectiva de los estudiantes de Ciencia de Cómputos que han realizado el curso de introducción a la programación. De igual forma se indagó sobre la posible correlación entre la motivación y la autoeficacia de los estudiantes para el aprendizaje de la programación, con su rendimiento académico.

La metodología de investigación fue de naturaleza cuantitativa. Se utilizó un diseño de investigación por encuesta, de tipo transversal. Para fines de la presente investigación se utilizó un cuestionario autoadministrado en línea realizado para este propósito. Se recopilaron evidencias de validez relacionadas con el contenido y proceso de respuesta. El muestreo fue no probabilístico por conveniencia.

En cuanto a los hallazgos de esta investigación los participantes informaron que en el aprendizaje inicial de la programación de computadoras los temas de mayor nivel de dificultad son: representar soluciones mediante abstracciones, dividir problemas extensos de programación en subproblemas, y búsqueda de errores en un programa. Estos temas

tienen en común que requieren altos niveles de abstracción y la resolución de problemas. Por otra parte, los estudiantes destacaron el aprendizaje colaborativo mediante comunidades de aprendizaje, como los ambientes de aprendizaje más efectivos para el aprendizaje de la programación de computadoras. De forma similar, revisar tutoriales en línea sobre programación, fue el recurso educativo más efectivo para el aprendizaje de la programación, percibido por los participantes en el presente estudio.

Por último, se halló que la autoeficacia y la motivación de los aprendices de programación se constituyen en factores que pueden incidir favorablemente en el rendimiento académico de los estudiantes. En el presente estudio se pudo encontrar una correlación de Spearman positiva significativa entre los ítems de las subescalas de autoeficacia y motivación con el rendimiento académico informado por los participantes en el curso de fundamentos de la programación.

## TABLA DE CONTENIDO

HOJA DE APROBACIÓN.....	i
DEDICATORIA .....	ii
RECONOCIMIENTOS .....	iii
RESUMEN DE LA DISERTACIÓN .....	iv
LISTA DE TABLAS .....	viii
CAPÍTULO I. INTRODUCCIÓN .....	1
Trasfondo .....	1
Planteamiento del problema.....	3
Propósito.....	8
Justificación.....	9
Proceso .....	14
Preguntas de investigación.....	15
Definiciones de términos.....	17
CAPÍTULO II. REVISIÓN DE LITERATURA .....	19
Introducción .....	19
Programación de computadoras .....	19
El aprendizaje de Aprendizaje de Programación .....	21
Motivación para el aprendizaje de la programación .....	24
Autoeficacia para el aprendizaje de la programación .....	27
El aprendizaje de la programación de computadoras en comunidades de aprendizaje. ....	36
Estudios relacionados.....	40
Cuestionarios de autoeficacia y motivación para el aprendizaje inicial de programación de computadoras.....	43
CAPÍTULO III. MÉTODO.....	48
Introducción .....	48
Preguntas de investigación.....	48
Metodología .....	49
Diseño de investigación .....	50
Población y selección de participantes.....	51
Procedimiento.....	53

Diseño y desarrollo del instrumento.....	54
Evidencias de validez .....	56
Evidencia basada en el proceso de respuesta.....	57
Evidencia basada en la estructura interna.....	58
Análisis de datos.....	58
Aspectos éticos.....	59
<b>CAPÍTULO IV. RESULTADOS.....</b>	<b>61</b>
Introducción .....	61
Datos sociodemográficos .....	62
Resultados del Coeficiente de consistencia interna Alfa de Cronbach .....	66
Resultados recopilados por pregunta de investigación: .....	68
Contenidos del Curso.....	68
Ambientes en los que se aprende programación .....	71
Utilidad de los recursos .....	72
La motivación para el aprendizaje de la programación.....	74
Autoeficacia para el aprendizaje de la programación.....	77
Correlación entre la motivación y el rendimiento académico .....	79
Correlación entre la autoeficacia y el rendimiento académico.....	82
<b>CAPÍTULO V. ANÁLISIS Y CONCLUSIONES .....</b>	<b>85</b>
Discusión de los hallazgos .....	85
Conclusiones .....	91
Limitaciones del estudio y sus implicaciones .....	93
Sugerencias para futuros estudios .....	94
<b>REFERENCIAS.....</b>	<b>96</b>
<b>APÉNDICES.....</b>	<b>105</b>
Certificado de <i>CITI Program</i> .....	106
Autorización del CIPSHI del protocolo de investigación .....	109
Hoja de consentimiento informado .....	111
<b>RESUMEN BIOGRÁFICO DEL AUTOR .....</b>	<b>115</b>

## LISTA DE TABLAS

Tabla	Página
1. Matrícula total del programa de Bachillerato de Ciencia de Cómputos primer semestre del año académico 2020-2021. Se presenta clasificación, según su género y año académico .....	51
2. Distribución de los ítems de la sección IV, para las tres necesidades básicas (TDA).....	55
3. Frecuencia y distribución porcentual de participantes por sexo.....	63
4. Frecuencia y distribución porcentual de los participantes por tipo de escuela .....	63
5. Frecuencia y distribución porcentual de los participantes por edad.....	64
6. Frecuencia y distribución porcentual de los participantes por calificación .....	65
7. Frecuencia y distribución porcentual de los participantes por Conocimiento previo de programación.....	66
8. Coeficiente de consistencia interna Alfa de Cronbach calculado y su categorización (George & Mallery, 2003, como se cita en Gliem & Gliem, 2003).....	67
9. Frecuencia y distribución porcentual de los aspectos relacionados a los contenidos del curso.....	70
10. Aspectos relacionados a los ambientes en los que aprende programación.....	72
11. Aspectos relacionados a utilidad de los recursos.....	73
12. Aspectos relacionados a la motivación para el aprendizaje de la programación.....	76
13. Aspectos relacionados a la autoeficacia para el aprendizaje de la programación .....	78
14. Análisis de correlación de Spearman ítems subescala IV (Motivación) y el rendimiento académico en el curso introducción a la programación de computadoras.....	81
15. Análisis de correlación de Spearman ítems subescala V (Autoeficacia) y el rendimiento académico en el curso introducción a la programación de computadoras.....	83

# CAPÍTULO I

## INTRODUCCIÓN

### Trasfondo

*“Aprender a escribir programas exprime tu mente, y ayuda a pensar mejor, crea una manera de pensar sobre cosas que creo es útil en todos los dominios”*

*Bill Gates (CODE, 2015)*

En la época cuando fui estudiante subgraduado y cursé asignaturas como fundamentos de programación y estructura de datos, recuerdo que muchos de los compañeros de estudio sintieron frustración por no poder comprender los contenidos de estos cursos, al no asimilar las nuevas estructuras y los elementos básicos para solucionar problemas con la programación. Esa frustración se convirtió posteriormente en apatía hacia las clases de programación y por último en deserción. Tal como lo perciben Muñoz, Barría, Noël, Providel & Quiroz (2012) quienes afirman que el aplicar conceptos básicos o diseñar algoritmos relativamente simples parece ser algo difícil para el estudiante.

De igual manera, autores como Pérez (2008), Robins, Rountree & Rountree (2003) y Santimateo, Nuñez & González (2018) coinciden en que la programación de computadoras no es una asignatura fácil de estudiar (Lahtinen, Ala-Mutka & Järvinen, 2005). De esta forma, “la aplicación de conceptos básicos o el diseño de algoritmos que son relativamente simples para los docentes parece ser algo difícil para el estudiante” (Carbone, Hurst, Mitchell & Gunstone, 2009, como se cita en Muñoz, Barría, Noël, Providel & Quiroz, 2012, p.120).

De ahí que se reporten altas tasas de abandono (Robins, Rountree & Rountree, 2003, Santimateo, Nuñez & González, 2018, Lahtinen, Ala-Mutka & Järvinen, 2005). En este sentido, “La programación, al ser una actividad mental compleja y creativa, requiere de 4 características: inteligencia, conocimiento, habilidades y disciplina, las cuales se adquieren con el paso del tiempo” (Oviedo, s.f. como se cita en Pérez, 2008, p.233), de ahí que existan “estudiantes que no logran adquirir las habilidades necesarias para la programación, incluso después de la terminación de un curso de fundamentos de programación en la Ciencia de la Computación” (Insuasti, 2016, p. 236). Por lo tanto, “La enseñanza de la programación es una actividad compleja y difícil” (Pérez, 2008, p. 230), en la cual se busca como objetivo principal: “que los alumnos desarrollen sus capacidades, adquiriendo competencias para crear programas computacionales que resuelvan problemas reales” (Figueiredo & García-Peñalvo, 2018, p.2).

Por lo tanto, autores como Fuentes-Rosado & Moo-Medina, (2017) plantean que aprender programación no es como adquirir otro conocimiento. No se trata solo de memorizar un conjunto de pasos, se requiere saber cómo utilizar esas estructuras aprendidas para cada situación o problema a resolver. “Es allí donde radica la dificultad de aprender a programar; de tener un problema y crear una solución” (Fuentes-Rosado & Moo-Medina, 2017, p.76-77). Muchas veces se cree de manera errónea que la programación de computadoras es para genios (Enríquez, 2017), pero es un pensamiento muy equivocado. La programación es parte de nuestra vida diaria cada vez que tomamos decisiones, de manera inconsciente nuestro pensamiento crea una estructura jerárquica de las acciones a realizar para cumplir con la tarea, por ello podemos afirmar que todos somos potenciales programadores. En este sentido, Enríquez (2017) considera que la

programación se puede aprender y enseñar integrando estrategias dinámicas y motivadoras que desarrollen la creatividad y el pensamiento crítico, según la edad del estudiante.

De igual forma, autores como Villalobos, Casallas & Marcos (2005), analizan las formas de enseñanza en este campo y con base en los resultados encontrados, plantean una didáctica específica para el primer curso de programación de computadoras. Esto se debe a que los estudiantes enfrentan dificultades en el aprendizaje de la programación debido a factores como: “motivación, estilos diferentes de aprendizaje, experiencia previa, entre otros” (Casas, & Vanoli, 2007; Felder & Brent, 2005; Felder & Silverman, 1988, como se cita en Muñoz, Barría, Noël, Providel & Quiroz, 2012, p. 120). Por lo tanto, se hace necesario indagar con relación a esos factores determinantes en el aprendizaje de la programación. Dada la importancia de aprender los fundamentos de programación en carreras como Ciencia de Cómputos, y en general como una competencia que debe desarrollar cualquier ciudadano en el futuro.

### **Planteamiento del problema**

“La enseñanza de los Fundamentos de Programación es una parte esencial de un curriculum en Ciencia de la Computación” (Muñoz, Barría, Noël, Providel & Quiroz, 2012, p. 120). Hoy día, es común encontrar que un alto porcentaje de estudiantes toman cursos de programación, algunos motivados por el auge de la programación en el desarrollo de videojuegos, redes sociales y diversos tipos de plataformas. Lo anterior se puede corroborar con datos emitidos desde la Oficina de Estadísticas Laborales en Estados Unidos. Ellos exponen que la demanda de desarrolladores de software estará

creciendo entre un 28% y un 32% para 2020 (Mohanarajah, 2018). Según el informe CRA por sus siglas de *Computing Research Association* (Zweben & Bizot, 2016), las nuevas carreras de pregrado relacionadas a la informática y programación han aumentado constantemente durante los últimos siete años. También, ha influido el éxito de algunos programadores que poseen, actualmente fama y compañías multimillonarias. Estas grandes compañías se caracterizan por ofrecer muy buenos salarios, lo cual motiva a muchos estudiantes a emprender este tipo de carrera. Tal como lo plantea Jenkins (2001) quien considera que una de las razones principales de los estudiantes para obtener un título en computación es la perspectiva de un trabajo altamente remunerado como programador, lo que se constituye como una gran motivación para cualquier estudiante.

No obstante, muchos de ellos una vez empiezan a tomar sus primeros cursos de programación, comienzan a sentirse frustrados porque no logran comprender la lógica inmersa en el aprendizaje de un lenguaje de programación y cada día que pasa la exigencia se va haciendo mayor. Esto se debe a que el aprendizaje de cualquier lenguaje de alto nivel se realiza de forma progresiva. De ahí que, la programación de computadoras es considerada desafiante, ocasionando que los estudiantes principiantes se frustren, cansen y se rindan fácilmente (Koulouri, Lauria, & Macredie, 2014, citados en Mohanarajah, 2018).

Por lo tanto, existe una problemática con el aprendizaje de programación de computadoras en estudiantes de Ciencia de Cómputo o cualquier carrera donde deban tomar cursos de programación. El aprendizaje de estos cursos les resulta retador, debido a que deben desarrollar competencias para adquirir entendimientos conceptuales complejos, muchas veces con altos niveles de abstracción, mientras aprenden los

componentes de un nuevo lenguaje de código, que luego deben implementar para resolver un problema de programación y así desarrollar un programa que funcione (Gulatee & Combes 2006).

Así mismo, Insuasti (2016) considera que programar computadoras es una tarea difícil, debido a la complejidad del lenguaje y lo abstracto que son los conceptos de programación. De igual forma, Fuentes-Rosado & Moo-Medina (2017) afirman que aprender programación no es como adquirir otro conocimiento, no es solo un proceso de memorización. Para aprender a programar no basta con aprender las palabras reservadas de un lenguaje para poder aplicarlo. Por lo tanto, para estos autores aprender a programar consiste en: “plasmar, mediante un lenguaje de programación, la forma de solucionar un problema. Cada problema se soluciona de manera distinta y cada programador lo resuelve de una forma diferente” (p. 76).

Así, por ejemplo, cuando se expone a un grupo de estudiantes ante un problema de programación, se obtienen diversas maneras de solución. Sin embargo, a diferencia de otros campos académicos donde lo importante es solucionar el problema, en la programación siempre existirá una solución más eficiente entre un grupo de soluciones, esta será la que menos líneas de código tenga y la que consuma menos recursos (e.g. memoria y procesamiento) a la computadora. En este sentido, “La creación y el control de ambientes y soluciones computacionales a través de la programación, son cosas que para un individuo pueden ser difíciles de realizar” (Soloway & Spohrer, 1989, citados por Insuasti, 2016, p. 236).

De ahí que, autores como Mohanarajah (2018) consideren que el aprendizaje de la programación requiera al menos tres habilidades claves, la primera capacidad de abstraer

un problema y construir una solución paso a paso para un entorno particular, la segunda la capacidad de comprender las estructuras semánticas de un lenguaje de programación y elegir la estructura adecuada para diseñar la solución y la tercera la capacidad de usar la sintaxis correcta para implementar el diseño en el entorno dado.

Para desarrollar estas competencias existen iniciativas como “La Hora del Código” organizada por [Code.org](https://code.org), una organización pública sin fines de lucro dedicada a promover la participación en escuelas e institutos de las Ciencia de la Computación, con una especial atención en incrementar el número de mujeres y estudiantes de colectivos minoritarios que aprenden a programar. De la misma manera “*YouthSpark*” de la compañía Microsoft, se encuentra a disposición de más de un millón de jóvenes en Latinoamérica. Estos cursos tienen por objetivo acercar a los jóvenes entre 11 y 25 años a los principales conceptos de programación para que se interesen en el campo.

Las alternativas anteriores, buscan iniciar de forma gratuita el aprendizaje de la programación de computadoras, por medio de estas plataformas web, que promueven la incorporación de una hora de código de programación en las escuelas desde los niveles de primaria hasta superior. Esto es posible mediante recursos didácticos e interactivos, como sencillos videojuegos, que los estudiantes deben aprender a programar de manera progresiva. Con estas iniciativas se intenta demostrar que el aprendizaje de la programación no es difícil y que cualquiera que esté interesado puede aprender programación básica. En este sentido, algunas investigaciones como la de Kori, Pedaste, Leijen & Tõnisson (2016) resaltan la importancia de la experiencia previa del aprendizaje de la programación de computadoras antes de entrar a estudiar a nivel universitario algún curso relacionado a esta. Debido a que estos autores encontraron diferencias en los

desempeños de manera positiva en estudiantes que tenían experiencias previas en comparación con los que no.

Existen varios factores involucrados en el aprendizaje de la programación, algunos son inherentes a la materia en cuestión, mientras que otros están relacionados con el proceso de enseñanza, expectativas, aptitudes y experiencia previa tanto de los profesores como de sus estudiantes (Jenkins, 2002, como se cita en Ortega, 2016).

Algunas investigaciones como la de Lahtinen, Ala-Mutka & Järvinen (2005) encuentran que el mayor problema de los programadores novatos no parece ser la comprensión de conceptos básicos, sino su aplicación, por lo que los autores concluyen que aprender haciendo debe ser parte de los estudios de programación de computadoras todo el tiempo. Por otra parte, en el aprendizaje de la programación se deben tener en cuenta diversos factores como: motivación, estilos de aprendizaje diferentes, experiencia previa, entre otros (Muñoz, Barría, Noël, Providel & Quiroz, 2012, p.120), al igual que el factor emocional (Bosch, D'Mello & Mills, 2013). Por ello, se ha identificado que la habilidad para solucionar problemas de programación comprende aspectos cognitivos, metacognitivos, e incluso aspectos motivacionales (Mayer, 1998; Parham, Gugerty & Stevenson, 2010, citados en Ortega, 2016).

Por lo tanto, para entender la dificultad de aprender a programar uno de los aspectos que se debe abordar es la motivación (Kelleher & Pausch, 2003). Con relación a lo anterior, se afirma que la falta de motivación es una de las principales razones del fracaso de los estudiantes en los cursos de programación (Santos & Costa, 2006; Ramos, 2013, citados en Tavares, Henriques, & Gomes, 2017). De igual manera, Ortega (2016) expone que:

Según Park & Hyun (2014) la motivación y las habilidades son parte importante para el éxito en la programación de computadoras, además de la autoeficacia, la cual esta correlacionada positivamente con las habilidades computacionales y afecta el rendimiento académico en la programación de computadoras (Askar & Davenport, 2009) (p.7).

Por lo tanto, aunque son muchos los factores que se pueden asociar a las dificultades que presentan los estudiantes que se inician en el aprendizaje de la programación, resulta de gran importancia conocer la percepción de los estudiantes de Ciencia de Cómputos acerca de los factores determinantes para el aprendizaje inicial de la programación de computadoras, con el fin de indagar de qué manera influye su nivel de motivación y autoeficacia en este aprendizaje.

### **Propósito**

De acuerdo con lo mencionado anteriormente, se hace necesario investigar, los factores determinantes en el aprendizaje de la programación de computadoras. Respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras. Además, se indagará sobre factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos, desde la perspectiva de los estudiantes de Ciencia de Cómputos que han realizado el curso de Introducción a la Programación.

## Justificación

Los cambios tecnológicos y la llegada del internet a nuestras vidas han generado cambios en nuestra sociedad, que han transformado nuestro modo de pensar, de sentir, y de actuar (Ossa, 2002). Una revisión más detallada de las características de la sociedad permite concebirla como un producto de una evolución desde la época agraria o sociedad 1.0, pasando por una sociedad industrializada o sociedad 2.0, hasta llegar a la sociedad del conocimiento y la información o sociedad 3.0 (Moravec, 2013). Por lo tanto, existe la necesidad de ir al ritmo de estos cambios que requieren de nuevas habilidades para abordar los desafíos sociales creados por la Inteligencia Artificial (AI) y las tecnologías robóticas (Sanabria & Romero, 2018). De esta forma, se hace evidente desarrollar en las próximas generaciones competencias acordes con estos desafíos.

En este sentido, el marco de los Proyectos co-tecnocreativos llamado #CoCreaTIC, señala cinco principales competencias del siglo XXI: pensamiento crítico, colaboración, creatividad, resolución de problemas, e incluye el pensamiento computacional (Wing, 2006; Romero, 2016, citados en Sanabria & Romero, 2018). Esta última competencia representa una serie de destrezas y conocimientos del siglo XXI que incluyen la solución de problemas, el diseño de sistemas y el entendimiento del comportamiento humano a través de conceptos fundamentales de la Ciencia de Cómputos que se deben desarrollar en los estudiantes (Wing, 2006 citado en Álvarez, 2018). Estas competencias, que son requeridas en la actualidad son desarrolladas con el aprendizaje de la programación. De ahí la importancia de propiciar este aprendizaje desde los primeros grados escolares, porque potencian el desarrollo de habilidades de pensamiento que son necesarias para desenvolverse con éxito en la sociedad actual (Fracchia, Alonso de

Armiño & Martins, 2014). De esta forma, los estudiantes tendrán muy buenas bases para desempeñarse en cualquier campo profesional que seleccionen en el futuro.

Existe una creciente demanda laboral hacia el campo de las Ciencia de Cómputos, área donde se profesionalizan los aprendices de programación. De acuerdo con la Oficina de Estadísticas Laborales de Estados Unidos, se prevé que para 2020 habrá 1,4 millones de empleos en la industria de las ciencias computacionales. Sin embargo, para el año 2020 solo se espera que gradúen 400.000 estudiantes de ciencias computacionales, lo que representa una necesidad laboral en este campo de aproximadamente un millón de puestos. De ahí la importancia del tema abordado en esta investigación, porque pretende aportar a la búsqueda de soluciones fundamentadas desde la teoría de la autoeficacia para incentivar la motivación de los estudiantes a no desistir del aprendizaje de programación de computadoras. Es importante recalcar lo que se mencionó anteriormente: “la motivación en el aprendizaje de programación es baja, debido a la novedad de la asignatura, la dificultad de los alumnos en aprender conceptos abstractos y en su posterior uso en la resolución de problemas” (Beltrán, Sánchez & Rico, 2015, p. 195).

También es importante destacar el aprendizaje de la programación como medio para que los estudiantes desarrollen habilidades para la resolución de problemas, pensamiento lógico y creativo (Jara & Hepp, 2016), destrezas necesarias no solo para desempeñarse con éxito en otras áreas de estudio, sino también para la vida. Esto es así ya que, dentro de las funciones sociales de la educación está ayudar a que las personas desarrollen su potencial y se preparen para desenvolverse de acuerdo con las necesidades de su comunidad. De esta manera, abordar temas como el aprendizaje de la programación, permite atemperar a la educación con los nuevos requerimientos que han

surgido en la sociedad actual, producto del impacto de la tecnología digital en nuestras vidas (Fracchia, Alonso de Armiño & Martins, 2014).

Sin embargo, en la literatura se evidencia que los estudiantes tienen dificultades en el aprendizaje inicial de la programación de computadoras (Lahtinen, Ala-Mutka & Järvinen, 2005; Bruno, 2005; Santimateo, Nuñez & González, 2018; Ala-Mutka, 2003; Carbone, Hurst, Mitchell & Gunstone, 2009; Casas & Vanoli, 2007; Insuasti, 2016; Fuentes-Rosado & Moo-Medina, 2017). Por lo tanto, “la programación de computadoras es considerada a menudo una tarea difícil” (Insuasti, 2016, p. 236). Esto se debe en gran medida a que “requiere una correcta comprensión de conceptos abstractos” (Lahtinen, Ala-Mutka & Järvinen, 2005, p.1), mediante un “proceso de transformación de un plan mental de términos corrientes en términos compatibles con el ordenador” (Figueiredo & García-Peñalvo, 2018, p. 2). De ahí que, autores como Lahtinen, Ala-Mutka & Järvinen (2005) y Santimateo, Nuñez & González (2018), han incluido en sus estudios la percepción de los estudiantes con relación a las actividades y conceptos usualmente utilizados en un curso de programación.

Por lo tanto, “es de suma importancia detectar cuales son los contenidos y/o conceptos que de acuerdo con la percepción de los estudiantes son complicados de asimilar y aplicar” (McCarthy & McCarthy, 2005, según se cita en Muñoz, Barría, Noël, Providel & Quiroz, 2012, p.5). De esta forma, se puede hacer un “análisis de los contenidos del primer curso de programación con el fin de plantear alternativas para fortalecer las habilidades en los estudiantes” (Hernández, 2013, p. 89) y facilitar el aprendizaje de los fundamentos de la programación. De igual forma, indagar sobre la percepción de los estudiantes que han tomado el curso de introducción a la programación

con relación a los factores determinantes en el aprendizaje de la programación de computadoras, permitirá, mejorar en el futuro la práctica docente y de esta forma poder propiciar experiencias de aprendizaje más pertinentes.

En este sentido, algunos factores que dificultan el aprendizaje de los fundamentos de programación son: “complejidad de la sintaxis del lenguaje y los conceptos de programación; la carga cognitiva implicada en el aprendizaje de programación; el mal diseño de los objetos de saber, y la falta de habilidades cognitivas propias para la solución de problemas” (Insuasti, 2016, p.236). En este mismo orden de ideas, Bruno (2005) plantea que los alumnos no están habituados al uso de la lógica para la resolución de problemas, lo que plantea en la enseñanza inicial de la programación un gran esfuerzo. Por su parte, Jenkins (2001) señala un conjunto de elementos que afectan el aprendizaje de la programación de computadoras como: habilidades cognitivas, novedad educativa, diferencia entre el aprendizaje para programar y aprender un lenguaje de programación particular, la motivación, expectativas y diferentes ritmos de aprendizaje. Por lo tanto, son muchos los factores que se asocian a las dificultades que presentan los estudiantes en el aprendizaje de la programación, sin embargo, no se conoce la causa fundamental del elevado índice de reprobación en esta asignatura (Fuentes-Rosado & Moo-Medina, 2017).

Respecto a esto, un factor determinante señalado en la literatura para el aprendizaje inicial de la programación es la motivación (Jenkins, 2001; Park, Kim, Robertson & Kim, 2019). De ahí que existan muchos estudios enfocados en desarrollar la

motivación en los estudiantes para el aprendizaje de la programación (Tavares, Henriques & Gomes, 2017; Jenkins, 2001; Mohanarajah, 2018; Guo, 2017).

“La motivación es concerniente a la energía, la dirección, la persistencia y la equifinalidad, todos aspectos de la activación y de la intención” (Ryan & Deci, 2000, p. 3). La motivación se puede dividir en función de las diferentes razones que dan lugar a una acción. En este sentido, de acuerdo con Ryan & Deci (2000) existen tres tipos principales de motivación: desmotivación (falta de motivación), motivación extrínseca y motivación intrínseca. La motivación tiene diferentes grados en un rango continuo estando en los extremos la desmotivación y la motivación intrínseca, pasando por la motivación extrínseca (Deci & Ryan, 2000). En el campo de la enseñanza de la programación de computadoras, se encuentra que una de las razones por las que los estudiantes abandonan los cursos de programación es la falta de motivación intrínseca (Bergin & Reilly, 2005, citados por Mohanarajah, 2018).

En particular, el Recinto de Río Piedras de la Universidad de Puerto Rico, en el departamento de Ciencia de Cómputos de la Facultad de Ciencias Naturales, se oferta el programa subgraduado del mismo nombre. Éste dentro de su plan de clases en el segundo semestre, incluye el curso Introducción a la Programación de Computadoras. Este curso aborda la solución de problemas mediante el desarrollo de algoritmos, incluyendo técnicas de programación con determinados lenguajes. Por lo tanto, resulta importante indagar sobre la percepción de los estudiantes de este programa que han tomado este curso, en cuanto a los factores determinantes en el aprendizaje de la programación de computadoras respecto a factores internos como: el nivel de motivación y autoeficacia

percibida en el aprendizaje de la programación de computadoras. Y factores externos como: contenidos del curso, ambientes y recursos de aprendizaje más efectivos.

De ahí que, los resultados de esta investigación contribuirán al mejoramiento de la calidad de la educación de los estudiantes de Ciencia de Cómputos para el aprendizaje inicial de la Programación de Computadora. Esta asignatura es base fundamental para un buen desempeño de los estudiantes en los cursos siguientes de programación. De igual manera, los resultados de esta investigación contribuyen en un tema que a nivel mundial es de alta relevancia, siendo que se está considerando en algunos países desarrollados incluir la enseñanza de la programación desde los primeros niveles escolares, por lo tanto, no debería estar reservada sólo para aquellos que trabajan con algoritmos, desarrollo de software, apps o páginas web. La programación, tendrá cada vez un peso más importante y será tan necesario como leer o escribir en nuestro idioma (Ocre, 2016).

### **Proceso**

El método de investigación que se planea utilizar es uno de naturaleza cuantitativa. La investigación cuantitativa estudia problemas de investigación donde se busca establecer la tendencia general de las respuestas de los individuos y observar cómo esta tendencia varía entre las personas (Creswell, 2012). “Se fundamenta en la medición de las características de los fenómenos sociales, lo cual supone derivar de un marco conceptual pertinente al problema analizado, una serie de postulados que expresen relaciones entre las variables estudiadas de forma deductiva” (Bernal, 2011, p. 60).

Para esta investigación se utilizará el diseño de investigación por encuesta. Este diseño fue seleccionado porque permite al investigador recopilar información necesaria para examinar la percepción que tienen los estudiantes del programa de Ciencia de Cómputos que han tomado el curso de Introducción a la programación de computadoras, con relación a factores internos como: el nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras. Y factores externos como: contenidos del curso, ambientes y recursos de aprendizaje más efectivos.

Según Creswell (2008), los estudios por encuesta describen tendencias en los datos en vez de ofrecer explicaciones sobre causa y efecto ya que en este tipo de diseño no involucra un tratamiento como ocurre en los diseños experimentales.

### **Preguntas de investigación**

En los estudios cuantitativos, los investigadores usan preguntas de investigación para dar forma y enfocar específicamente el propósito del estudio. Las preguntas de investigación cuantitativa indagan sobre las relaciones entre las variables que el investigador busca conocer (Creswell, 2012). En particular, las preguntas que dirigen la presente investigación son las siguientes:

1. ¿Cuál es la percepción de los estudiantes subgraduados del programa de Ciencia de Cómputos respecto al nivel de dificultad de los contenidos del curso introducción a la programación de computadoras?

2. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los ambientes de aprendizaje más efectivos para la programación?
3. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los recursos educativos más efectivos para el aprendizaje de la programación?
4. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a su nivel de motivación para el aprendizaje de la programación?
5. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos de su nivel de autoeficacia para el aprendizaje de la programación de computadoras?
6. ¿Cuál es la correlación entre la motivación de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras?
7. ¿Cuál es la correlación entre la autoeficacia de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras?

## Definiciones de términos

A continuación, se incluyen las definiciones de términos relevantes en el contexto de la presente investigación de acuerdo con el planteamiento del problema, esto con el propósito de evitar ambigüedades.

1. Motivación: Desde la psicología, siguiendo la teoría de Atkinson, señala que la motivación de rendimiento se determina por el valor otorgado a la meta y las expectativas por lograrla, teniendo en cuenta las características de las personas con alta o baja necesidad de rendimiento, ansiedad y control interno (citado por Herrera et al., 2004).
2. Autoeficacia: Según Bandura (1977), la autoeficacia en sí se define como el conjunto de juicios de cada individuo sobre las capacidades propias para organizar y ejecutar las acciones requeridas en el manejo de posibles situaciones específicas. Tales juicios se entienden que tienen importantes efectos sobre la elección de conductas o actividades, sobre el esfuerzo empleado y la persistencia, y sobre los patrones de pensamiento y las reacciones emocionales ante las tareas (Blanco, 2010, p. 2).
3. Autoeficacia en la programación de computadoras: es el nivel de los esfuerzos mostrados para resolver problemas de programación y para abarcar el efecto del aprendizaje motivación y actitud hacia la programación.
4. Percepción del estudiante: es el proceso que permite interpretar la realidad y asignarle un significado (Carrancio, 2013).
5. Ambiente de aprendizaje: “es un sistema integrado por un conjunto de elementos físicos, sociales, culturales, psicológicos, pedagógicos, relacionados y

organizados entre sí que posibilitan generar circunstancias estimulantes favorecedoras de aprendizaje” (García-Chato, 2014, p. 71).

6. Rendimiento académico: corresponde a la calificación final obtenida por el estudiante en el curso de fundamentos de programación.

## **CAPÍTULO II.**

### **REVISIÓN DE LITERATURA**

#### **Introducción**

El presente estudio se propone investigar, los factores determinantes en el aprendizaje de la programación de computadoras. Respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras. Además, se investigará sobre factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos, desde la perspectiva de los estudiantes de Ciencia de Cómputos que han realizado el curso de introducción a la programación.

En este capítulo se incluyen los siguientes temas: definición de programación de computadoras y su aprendizaje. En particular, el aprendizaje de la programación de computadoras en comunidades de aprendizaje. Adicional a esto, se aborda la motivación y la autoeficacia para el aprendizaje de la programación, y al final del capítulo se presentan estudios cuantitativos relacionados con el aprendizaje inicial de la programación de computadoras.

#### **Programación de computadoras**

Programar “significa analizar, resolver, plantear alternativas a un determinado problema y definir cuáles son las órdenes o instrucciones que se debe dar a una computadora para que halle la solución al mismo” (Moroni, 2001, p.1). De manera sencilla, la programación también puede ser definida como un conjunto de instrucciones

finitas usando un lenguaje de alto nivel con un propósito o fin determinado. También, se puede asociar a la programación como un arte donde la creatividad, el ingenio, los conocimientos en lenguajes de programación y las habilidades cognitivas orientadas a la solución de problemas, son factores clave del éxito (Insuasti, 2016).

“Las computadoras y la forma de programarlas han evolucionado de una forma vertiginosa con el paso del tiempo” (Ferreira & Rojo, 2006, p. 1). Desde la aparición de “la primera máquina capaz de computar, diseñada por Charles Babbage en los años de 1830 y 1840” (Sahuquillo Borrás, Hassan Mohamed, Lemus Zúñiga, Molero, Ors Carot, & Rodríguez Ballester, 1997, citados en Hernández, 2013, p. 90), hasta la computación moderna, han surgido grandes cambios a través de cada una de las cinco generaciones en la evolución de las computadoras (Leyva, 1999).

Sin embargo, existe una clara distinción entre la Ciencia de Computación o Ciencia de Cómputos y la construcción de computadoras. La Ciencia de Cómputos puede definirse como “el estudio de la resolución de problemas con el computador. En cambio, el diseño y construcción de computadoras concierne a la electrónica y a la ingeniería” (Nieto, 2005, p. 42). De acuerdo con la *Association for Computing Machinery* y la *IEEE Computer Society* (2004) la Ciencia de Cómputos es considerada como “un área de conocimiento donde coinciden conceptos y habilidades esenciales para la práctica de la programación independiente del paradigma” (Hernández, 2013, p. 90).

Por lo tanto, “aprender a programar no implica únicamente alcanzar el conocimiento necesario para escribir programas en un lenguaje de computadora. De hecho, lo más importante es adquirir habilidades generales para crear planes y mecanismos que resuelvan problemas” (Sleman, 1986, como se cita en Casares, 1999, p.

15). En este mismo sentido, Villalobos & Casallas (2006, como se cita en Hernández, 2013) consideran la programación de computadoras, “como una actividad cuyo objetivo es ayudar a resolver un problema, generalmente de otras disciplinas, construyendo una solución que utiliza como herramienta el computador, para elaborar un programa de computadora” (p. 90).

Un programa de computadora se constituye en la “resolución formal de un problema y la obtención de su solución, siendo esta última operatoria, ejecutable, susceptible de ser puesta a prueba, verificable” (Moroni, 2001, p. 1). En este mismo sentido Sleeman (1986, citado en Casares, 1999) indica que el primer objetivo de un programador es especificar un plan detallado y realizable (un algoritmo) que solucione el problema. Lo siguiente es implementar este plan en un lenguaje de programación y, por último, depurar el programa resultante.

Por tanto, la programación “es el proceso de desarrollar e implementar un conjunto de instrucciones lógicas para que un dispositivo digital ejecute una tarea con el fin de resolver un problema y permitir la interacción con humanos, usando un lenguaje de programación” (Kazimoglu, Kiernan, Bacon & MacKinnon, 2011, citados por, Alvarez, 2018, p. 59-60).

### **El aprendizaje de Aprendizaje de Programación**

Aprender a escribir programas de computadora es conocido por ser difícil para muchos principiantes (Law, Lee & Yu, 2010). Aprender a programar equivale a aprender cómo construir mecanismos y cómo construir explicaciones. La necesidad de construir mecanismos y explicaciones trasciende el dominio de la programación, puesto que, las

personas ejecutan estas habilidades constantemente en la vida diaria, (Soloway, 1986).

De allí que se afirme que el aprendizaje de la programación de computadora requiere una carga cognitiva considerable en los estudiantes (Garner, 2009, citado por Korkmaz & Altun, 2014).

“La programación de algoritmos representa un caso de resolución de problemas que requiere representación mental del mundo real, adaptación para tener una solución computable y criterio para elegir una alternativa eficiente de implementación” (Palacios, 2000, citado en Bruno, 2005, p. 2). De ahí el nivel de dificultad que constituye el aprendizaje de esta.

De igual forma, Figueiredo & García-Peñalvo (2018) afirman que:

La programación es difícil. La programación es un proceso de transformación de un plan mental de términos corrientes en términos compatibles con el ordenador. La enseñanza de la programación tiene como principal objetivo lograr que los alumnos desarrollen sus capacidades, adquiriendo competencias para crear programas computacionales que resuelvan problemas reales (p.2)

Es bien sabido que las habilidades de programación son muy difíciles de mejorar y estas habilidades son una barrera importante para muchos estudiantes para completar sus carreras profesionales (Park, Kim, Robertson & Kim, 2019). Esto se evidencia con las altas tasas de deserción en los cursos de programación (Lahtinen, Ala-Mutka & Järvinen, 2005, Figueiredo & García-Peñalvo, 2018; Isuasti, 2016).

Un enfoque muy utilizado en la enseñanza de los fundamentos de programación es “abordar primero lo básico del lenguaje (o de los lenguajes) de programación, para luego

guiar a los estudiantes a través de estrategias donde se contemple la totalidad del proceso de programación de computadoras” (Isuasti, 2016, p. 236). Sin embargo, Soloway (1986, citado en Santimateo, et al., 2018, p. 14) indica que, “enseñar sintaxis y semántica de un lenguaje de programación no es suficiente para el aprendizaje de la programación, se considera de mayor importancia fortalecer habilidades para solucionar problemas”.

En este sentido, se ha comprobado, que “una estrategia trascendental es comenzar a enseñar programación utilizando los algoritmos como recursos esquemáticos para plasmar el modelo de la resolución de un problema” (Moroni, 2005, citada en Pérez, 2008, p. 233). Fuentes-Rosado, J. I., & Moo-Medina, M. (2017). “Existe actualmente un consenso general dentro de la comunidad educativa mundial sobre la necesidad de superar el tipo de enseñanza basada en la transmisión de contenidos para apuntarle en su lugar al desarrollo de capacidades” (López, 2009, p.4).

Sin embargo, ¿cómo aprende un novato en Ciencia de la computación a hacerse experto en programación?

Autores como Law, Lee & Yu (2010) exponen que ha sido una preocupación desde hace mucho tiempo, la búsqueda de alternativas para facilitar el aprendizaje de la programación. Resaltando, iniciativas como la de Allen, Cart-Wright y Stoler (2002, citados por Law, Lee & Yu, 2010) con DrJava, un entorno de desarrollo de programas personalizado. También se encuentra sistemas de animación de algoritmos como TANGO (Stasko, 1990, citados por Law, Lee & Yu, 2010) y ANIMAL (Röbling y Freisleben, 2002, citados por Law, Lee & Yu, 2010) que guían a los estudiantes a aprender dinámica de la ejecución del programa al demostrar visualmente cómo funcionan los algoritmos (Law, Lee & Yu, 2010).

## Motivación para el aprendizaje de la programación

La motivación es concerniente a la energía, la dirección, la persistencia y la equifinalidad que puede tener una persona para realizar una tarea o actividad (Ryan & Deci, 2000). En el ámbito educativo, la motivación del aprendizaje puede ser definida como el esfuerzo persistente que un estudiante realiza para aprender (Law, Lee & Yu, 2010). Se cree que la motivación es un facilitador para el aprendizaje y el éxito académico (Lin-nenbrink & Pintrich, 2002; Lynch, 2006, citados por Law, Lee & Yu, 2010). Sin embargo, “La motivación es un concepto abstracto y difícil de medir de manera objetiva” (Lumsden, 1994, citado en Arenas, 2014, p. 14).

De igual manera, la motivación del estudiante dirige su comportamiento hacia su meta de éxito en una clase de programación, y esta motivación influye en su persistencia, esfuerzo y energía para aprender algo nuevo (Ornlrod, 2008, citados por Park, Kim, Robertson & Kim, 2019). De acuerdo con Park, Kim, Robertson & Kim (2019) el nivel de motivación del estudiante está positivamente relacionado con su proceso de aprendizaje de programación de computadora. Según Arenas (2014, p. 14) en el aprendizaje de la programación de computadoras, tres factores principales afectan la motivación de los estudiantes:

-El estudiante espera poder programar cualquier tipo de software después de tomar una clase de programación, lo cual hace que se frustren sus expectativas al descubrir que no es así (Gries, 2002).

-El estudiante no encuentra una relación entre la profesión elegida y la programación Jenkins (2001).

-El estudiante cree que se necesita cierta habilidad innata para programar y si no la posee, no podrá aprender sin importar cuánto lo intente (Jenkins, 2002).

En este mismo sentido, Fang (2012), Garner (2009) y Nilsen y Larsen (2011) citados por Korkmaz & Altun (2014) señalan que muy frecuentemente los estudiantes experimentan dificultades con los cursos de programación ya que no entienden el contenido y la estructura algorítmica de los programas debido, principalmente a la baja autoeficacia y motivación.

De acuerdo con Ryan & Deci (2000) existen tres tipos de motivación: motivación extrínseca, motivación intrínseca y desmotivación. El término *motivación extrínseca* se refiere al desempeño de una actividad a fin de obtener algún resultado separable y, por lo tanto, contrasta con el de *motivación intrínseca* que se refiere al hacer una actividad por la satisfacción inherente que ocasiona la actividad por sí misma (Ryan & Deci, 2000). De allí que se enfatice en la motivación intrínseca como la tendencia inherente a buscar la novedad y el desafío, a extender y ejercitar las propias capacidades, a explorar, y a aprender (Ryan & Deci, 2000). Por otra parte, la desmotivación es un estado de ausencia de motivación, de falta de intención para actuar.

En su teoría de la autodeterminación (TDA), Ryan & Deci (2000) consideran que la motivación tiene diferentes niveles de autodeterminación. Siendo que, de mayor autodeterminación a menor, se encuentra primero la motivación intrínseca, seguida de la

motivación extrínseca y por último la desmotivación. La motivación intrínseca (regulación intrínseca) es enteramente interna, surgiendo del interés personal, curiosidad o disfrute de la tarea. Contraria esta, se encuentra la desmotivación (no regulación) que da como resultado inacción o acción sin intención real. En el medio está la motivación extrínseca, que se refiere al desempeño de una actividad a fin de obtener algún resultado separable y, por lo tanto, contrasta con el de *motivación intrínseca* que se refiere al hacer una actividad por la satisfacción inherente que ocasiona la actividad por sí misma. En la motivación extrínseca se distinguen cuatro estilos regulatorios que varían de regulación externa (acciones motivadas puramente por las consecuencias favorables o desfavorables anticipadas) para integrar regulación (en la que los valores y objetivos externos se han integrado completamente en la imagen de uno mismo) (Deci & Ryan, 2000).

Dentro de la teoría de la autodeterminación de Ryan & Deci (2000), se sugiere que la internalización y la integración son promovidas (o inhibidas) por el cumplimiento (o no cumplimiento) de tres necesidades psicosociales básicas: relación con los demás, competencia y autonomía. La atención de estas necesidades se constituye en un factor determinante para que un estudiante permanezca en sus estudios (Deci & Ryan, 2000). Lo anterior implica que para el aprendizaje de Programación de computadoras se debe atender estas necesidades.

La necesidad de autonomía “no se refiere a ser independiente, apartado ni egoísta sino más bien al sentimiento de voluntariedad que puede acompañar a cualquier acto, sea dependiente o independiente, colectivista o individualista” (Ryan y Deci, 2000, p. 9). La necesidad de relación con los demás, esboza el deseo de “sentirse perteneciendo y conectado con otros” (Ryan y Deci, 2000, p. 8), y la de competencia o autoeficacia, que

se refiere a la percepción de una habilidad adecuada para la realización de la actividad que se está desempeñando (Deci & Ryan, 2000).

Algunos estudios como el de Mohanarajah (2018), presentan un enfoque único para mantener la motivación intrínseca de los estudiantes hacia el aprendizaje de la programación, conservando un nivel de autoeficacia alto. En este sentido, en dicha investigación, mediante el uso de juegos digitales fáciles de aprender y jugar, los estudiantes mientras juegan son desafiados a solucionar los errores que encuentren. De esta forma, la corrección de errores se realiza como parte del juego y no como una interrupción molesta. Para corregir un error, los estudiantes primero deben inspeccionar el código y comprender la falla. Al hacer esto, los estudiantes obtienen un sentido de propiedad sobre el juego y, a su vez, aumentan su nivel de autoeficacia (Mohanarajah, 2018).

### **Autoeficacia para el aprendizaje de la programación**

La autoeficacia es otro aspecto importante que considerar para el aprendizaje inicial de la programación de computadoras. De acuerdo con Askar & Davenport (2009) la autoeficacia, está correlacionada positivamente con las habilidades computacionales y afecta el rendimiento académico en la programación de computadoras. De igual manera, Seturaman y Medley (2009, citados en Govender et al., 2014) indicaron que las creencias de los estudiantes sobre su autoeficacia en la programación podrían usarse para determinar qué tan bien los estudiantes están desempeñándose en sus cursos de programación. De allí que, se indique que entre los factores que afectan el éxito de un estudiante en el proceso de aprendizaje inicial en la programación de computadoras, se

destacan la actitud y la percepción de autoeficacia como los factores más determinantes (Anastasiadou & Karakos, 2011; Erdogan et al., 2008; Sacks et al., 1993; Austin, 1987 citados por Korkmaz & Altun, 2014).

La teoría de la autoeficacia propuesta por Bandura (1997), señala que la creencia del sujeto en su propia capacidad personal puede afectar las metas trazadas, su cumplimiento y compromiso con ellas. Las creencias de autoeficacia también son maleables y afectan el rendimiento intelectual de una persona. Por estas razones, los estudios sobre autoeficacia son muy relevantes. Esta teoría se ha aplicado al comportamiento en muchos dominios, incluyendo escuela, salud, deportes, terapia e, incluso, fobia a las serpientes (Bandura 1997). De igual manera, se ha utilizado ampliamente en las ciencias relacionadas con la conducta y los servicios humanos (Strousser, 1995). Algunas investigaciones en autoeficacia defienden que los maestros obtendrían más información al considerar no solo la competencia real de los estudiantes, sino también sus percepciones de su competencia (Govender et al., 2014). Por lo tanto, los escenarios donde se imparte la enseñanza y el aprendizaje de la programación de computadoras están estrechamente relacionados con el desarrollo de la autoeficacia.

Según Bandura (1997) cada individuo tiene unas creencias propias sobre la competencia, habilidades o destrezas que posee para realizar una tarea o actividad. En consecuencia, la forma en que las personas se comportan puede predecirse mejor por sus creencias sobre sus capacidades que por lo que realmente son capaces de lograr (Pajares, 2002). Esta creencia que tienen las personas en sí mismas es precisamente lo que se llama autoeficacia. Bonetto, Paoloni & Donolo (2017) mencionan que, “la autoeficacia alta predispone al sujeto a una mayor curiosidad y pasión por instruirse, una mayor

persistencia en disciplinas universitarias difíciles, un mayor valor atribuido a las tareas y, en consecuencia, mayores posibilidades de obtener el éxito académico” (p. 15). En contraste aquellas personas que dudan de su capacidad para enfrentar una tarea por temor al fracaso poseen una autoeficacia baja (Bandura, 1997).

Los estudiantes de programación muchas veces enfrentan este tipo de sentimientos, cuando creen no poder hacer un programa de computadoras. Esto normalmente ocurre debido a que solo memorizan las palabras reservadas que poseen los lenguajes de programación. Aunque sientan que entienden, y comprenden el código con el que se escribe un programa, no son capaces de resolver problemas planteados en estos lenguajes o crear su propia solución. “Estas dificultades se manifiestan independientemente al paradigma y/o lenguaje utilizado. Esto se puede deber a diversos factores, tales como motivación, sintaxis, estilos de aprendizajes, experiencia previa, entre otros” (Muñoz, Barcelos, Villarroel, Barría, Becerra, Noel & Silveira, 2015, p. 249).

Bandura (1997) identificó cuatro fuentes principales que constituyen las creencias de autoeficacia: a) Experiencias directas o de dominio. Esta fuente está relacionada con las experiencias previas que ha tenido el sujeto sobre lo que va a enfrentar. b) El aprendizaje vicario. Este tipo de aprendizaje se da por observación directa e imitación. Bandura (1997) asegura que el aprendizaje vicario es más efectivo que el expuesto por las teorías conductistas en donde el sujeto aprende por ensayo y error. c) La persuasión verbal. Está relacionada con los estímulos que recibe el sujeto exponiéndolo a palabras que le ayudan a subir su autoestima y por consiguiente a mejorar su percepción y creencia de poder enfrentar o realizar una acción. d) El manejo de los estados fisiológicos

y emocionales. Es la fuente que está relacionada con la forma como se activa nuestro organismo de manera psicológica y fisiológica para enfrentar la ejecución y organización de una actividad o tarea. Por ejemplo, la angustia, la preocupación, la ansiedad, la incertidumbre, el estrés y el estado de humor. A continuación, ampliaré cómo, desde este modelo de autoeficacia, se puede mejorar la problemática del aprendizaje de la programación en los estudiantes.

Algunos estudios como los de Kori, Pedaste, Leijen & Tõnisson (2016) resaltan la importancia de que los estudiantes tengan experiencia en programación antes de iniciar los estudios universitarios y en particular si se trata de una carrera de Ciencia de Cómputos. De esta forma, los autores exponen como en los primeros cursos de fundamentos e introducción a los lenguajes de programación se observa la gran diferencia en el desempeño entre un estudiante que posee experiencias previas positivas, con respecto al conocimiento específico de programación de computadoras y quien no las posee o tuvo una mala experiencia. En este sentido desde la teoría de la autoeficacia de Bandura (1997), poseer estas experiencias previas, sean positivas o negativas puede incrementar o disminuir la creencia que tiene un estudiante de sí mismo. Este concepto de experiencias directas, según el autor, es una de las fuentes primarias más poderosas para identificar la autoeficacia que posee una persona cuando se enfrenta a una situación.

Desde las experiencias directas, se puede aplicar la autoeficacia resiliente la cual, según Ruíz (2005), no se forma con éxitos fáciles. Esto se debe a que, si un estudiante solo experimenta experiencias positivas, se acostumbra a rápidos resultados y se desanima fácilmente ante el fracaso. En cambio, si se le enseña el sentido de autoeficacia resiliente se formará con la experiencia de vencer obstáculos mediante esfuerzos

perseverantes. Actualmente los estudiantes no toleran equivocarse, pues su nivel de tolerancia a los errores cometidos es muy bajo. Por ello, es importante que un estudiante entienda que los errores son parte importante del aprendizaje. Así, cuando los estudiantes estén convencidos de que tienen lo necesario para tener éxito, perseverarán ante las adversidades y se recuperarán rápidamente ante cualquier experiencia negativa que les ocurra.

En el caso del aprendizaje de la programación, es posible que un estudiante que presenta niveles de frustración muy altos, al no lograr resolver un problema de programación por medio de un algoritmo, generalmente empieza a mostrarse apático hacia la asignatura y a expresar pensamientos negativos antes de enfrentar nuevamente el problema (Kori, Pedaste, Leijen & Tõnisson, 2016). Situaciones como esta pueden ocurrir incluso, con estudiantes que en otras áreas académicas son muy buenos, al ser siempre exitosos no toleran el fracaso y, en vez de persistir, desisten de volver intentarlo. Por lo tanto, el desarrollo de la autoeficacia en los estudiantes es un aspecto muy importante para trabajar en ellos y que, incluso, puede ser decisivo para que logre concretar sus propósitos académicos (Mohanarajah, 2018).

Por otra parte, desde el aprendizaje vicario, se puede incentivar la autoeficacia del estudiante. En este sentido, mediante la realización de proyectos de programación de manera colaborativa, trabajando en grupos, en ambientes atractivos de aprendizaje que además se constituyan en espacios lúdicos (Jara & Hepp, 2016). Cuando los estudiantes observan cómo sus pares resuelven problemas de programación con aciertos y equivocaciones, construyen y moldean su propia experiencia vicaria, debido a que existe una relación de iguales. Precisamente, recursos como *Youtube*, foros, chats y las redes

sociales han permitido que el aprendizaje vicario esté generando una gran explosión de creatividad, e influyendo en la forma en que aprendemos. En este sentido, es común ver que actualmente existen comunidades de programadores formulando preguntas y resolviéndolas de forma colaborativa.

Este tipo de aprendizaje por observación ha suscitado un gran auge y revuelo en el mundo, no solo a nivel de programación, sino que ha permeado todos los campos del conocimiento humano, gracias a las herramientas digitales con las que contamos actualmente. Se debe incentivar a los estudiantes a participar en este tipo de comunidades de aprendizaje, donde junto a sus pares encuentren posibles caminos de solución a sus interrogantes. En estos espacios, ellos no solo se benefician del conocimiento de otras personas, sino que también cuando aportan sus ideas para ayudar a otros, afianzan su propio conocimiento y, sobre todo, fortalecen el aprendizaje respecto a cómo resolver problemas de programación.

El siguiente aspecto por considerar es la persuasión verbal. La autoeficacia puede ser estimulada por medio de la expresión oral, dirigiendo palabras alentadoras a modo de refuerzo positivo. Por ejemplo, cuando un estudiante hace aportes a la clase, los profesores pueden animarlo con frases motivadoras que resalten su participación, mencionando quizás esas cualidades positivas que lo destacan. Además, resulta muy significativo que estas palabras de ánimo provengan no solo de los profesores sino también de su círculo cercano de personas tales como compañeros de clases, amistades, hermanos, padres y otras que tengan un grado de importancia en su vida.

Desde un curso de programación de computadoras los estímulos verbales que recibe el estudiante le permitirán reforzar y aumentar su creencia en sí mismo,

aumentando su autoeficacia de forma positiva. El resultado de ese refuerzo se puede ver reflejado en la forma como interiorizan los problemas de programación. La persuasión verbal refuerza sus capacidades incentivando su perseverancia, para intentar una y otra vez solucionar problemas mediante algoritmos y mejorando así su actitud en el aprendizaje de la programación.

Como se indica previamente, otro aspecto importante para el desarrollo de la autoeficacia es el manejo de los estados fisiológicos y emocionales. En este punto se debe trabajar con las emociones de los estudiantes, sobre todo aquellas que no les permitan mostrar, o les bloquean sus cualidades positivas. Por ejemplo, en una clase de programación un estudiante resuelve un problema de manera creativa, es resaltado por el profesor por ese logro y se le invita a que comparta mediante una presentación a sus compañeros la manera como encontró la solución al problema. Puede ocurrir que esta situación le provoque estrés, ansiedad, y alteración de los nervios y no sepa cómo manejar ese estado de ánimo. Definitivamente, sino tiene control sobre las emociones su autoeficacia se verá disminuida. “Las personas se guían por sus estados corporales y psicológicos para evaluar sus capacidades. De este modo, estas infieren qué reacciones de tensión y de estrés son signos de pobre desempeño y/o vulnerabilidad” (Ruiz, 2005, p.10).

Por lo anterior, desde el modelo de la autoeficacia de Bandura (1997), se debe promover la exposición a experiencias directas de aprendizaje de la programación desde los primeros años de escolaridad (Soykan & Kanbul, 2018). Así el estudiante, de forma natural puede ir desarrollando habilidades que le permitan ir desde un menor a un mayor nivel de dificultad y le ayuden a sentirse más competente para la realización de una tarea.

Algunos autores proponen el uso de videojuegos (Gamificación), para hacer más ameno el aprendizaje de la programación (Mohanarajah, 2018). De esta forma, mediante el uso de algunos lenguajes de programación visual como *Scratch*, el estudiante puede reducir errores de sintaxis e incorporar nuevos conceptos de forma divertida. Además, desde la robótica, con programados como *Robots Leg*, los estudiantes pueden establecer a un determinado objeto funciones mediante un algoritmo y luego ver su ejecución. El uso de este tipo de material didáctico (*Scratch* y *Robots Lego*), tienen un alto potencial para desarrollar una aplicación más tangible y concreta de las habilidades requeridas para la programación (Muñoz et al., 2012). Desde mi experiencia como programador y docente de esta área, puedo asegurar que cuando se le plantean al estudiante ejercicios de programación basados en situaciones de la vida real, les resulta más significativo y se animan más fácilmente para la búsqueda de soluciones.

También, desde las experiencias vicarias, se puede generar en la escuela espacios para la creación de comunidades de aprendizaje, mediante las cuales, los estudiantes con mayor habilidad en la programación interactúen con otros de menor habilidad y puedan observarlos mientras resuelven un problema haciendo uso de la programación. De esta forma, pueden intercambiar ideas y experiencias con sus pares. Desde la teoría de la autoeficacia de Bandura (1997), mediante el modelado de la ejecutoria exitosa, los estudiantes pueden llegar a sentir que son capaces de realizar la misma actividad con éxito. Aún más, si observan a estudiantes como ellos realizar esa actividad, pueden asociar más fácilmente que tienen capacidades suficientes para realizar lo mismo.

Sin embargo, es importante resaltar que el rol del maestro es fundamental en todo este proceso, porque además de proveer las experiencias de aprendizaje pertinentes, debe

reforzar las creencias de los estudiantes de su capacidad para aprender a programar mediante el refuerzo positivo, de la persuasión verbal (Bandura, 1997). En este sentido, es necesario que el profesorado “tenga a su disposición diversas oportunidades de desarrollo profesional dirigidas a desarrollar nuevas destrezas tecnológicas para poder aplicar esos conocimientos y destrezas, tanto en la sala de clases, como en la vida profesional y personal” (Álvarez, 2018, p.17). La *International Society for Technology in Education* (ISTE) es una de las principales asociaciones de educadores que promueve el uso eficaz de la tecnología y el adiestramiento de los docentes. Dentro de sus estándares incluyen las habilidades y conocimientos que los educadores necesitan aprender, trabajar y enseñar en una sociedad global y digital, cada vez más conectada (*International Society for Technology in Education, 2008*)

Actualmente las compañías pioneras en tecnología están rompiendo con la tradición de contratar a alguien por los títulos que ostentan. Muchas de ellas han elegido contratar a sus empleados por lo que son capaces de aprender. De igual forma, algunas compañías como *Microsoft*, la cual conoce el potencial que existe en el aprendizaje de la programación para los estudiantes, ofrece de forma gratuita por medio de su portal *#yopuedoprogramar*, enseñanza a estudiantes y profesores sobre distintas herramientas de programación (Jara & Hepp, 2016). Por medio de esta campaña, la compañía tiene como propósito promover el desarrollo de programas sencillos que permitan crear sitios web, aplicaciones para celulares y otros productos o servicios digitales. Además, con esto se propicia el desarrollo de los estudiantes y mejora su pensamiento crítico y la capacidad de resolución de problemas.

Hoy en día se han “multiplicado iniciativas privadas impulsadas por empresas del rubro, fundaciones y organizaciones sin fines de lucro que invitan a los jóvenes a aprender a programar como una forma de complementar la escuela y ampliar sus posibilidades de aprender, crear y conseguir trabajo” (Jara & Hepp, 2016, p. 2). La organización más destacada para enseñar a los estudiantes estas habilidades del siglo XXI es *Code.org*, que se estableció como una organización sin fines de lucro con el objetivo de promover la educación en ciencias de la computación (Code.Org, 2017, citado por Soykan & Kanbul, 2018).

### **El aprendizaje de la programación de computadoras en comunidades de aprendizaje**

“Las comunidades de aprendizaje se basan en el conocimiento de la realidad social” (Elboj, Puigdellivol, Soler Gallart & Valls Carol, 2003, p. 37). “Tomando como base las aportaciones de diferentes autores como Freire, desde el ámbito de la educación, Bruner y Vygotsky, desde la psicología y el enfoque socio histórico del desarrollo, y Habermas, desde la sociología, entre otros” (Morales, Chaclán, Maldonado, Sontay, Montenegro & Magzul, 2013, p. 5). Uno de los conceptos fundamentales de las comunidades de aprendizaje es el de aprendizaje dialógico. Este hace referencia a que se “aprende a través de las interacciones entre iguales, profesorado, familiares, amigos y amigas... que produce el diálogo igualitario” (Elboj, et al., 2003, p. 53). En otras palabras, Rodríguez (2012, citado en Morales et al., 2013) indica que “todos aprendemos de las interacciones, con los iguales más capaces, con los adultos en general y con los adultos

profesionales de la educación, agrupamientos e interacciones en ningún caso excluyentes” (p. 5).

Actualmente, en el aprendizaje de la programación de computadoras, el desarrollo de comunidades de aprendizaje ha tomado gran relevancia. En este sentido, Teague (2008, citado en Lovos, 2015) resalta que el aprendizaje colaborativo en los estudiantes de programación “los alienta a pensar en voz alta y a verbalizar el proceso de resolución de problemas, satisfaciendo de esta forma, la necesidad de interacción de los alumnos actuales” (p. 14). Adicional a lo anterior, una de las habilidades requeridas por las empresas dedicadas a la industria del software es que los programadores de computadoras (desarrolladores) tengan la capacidad de trabajo en equipo y la habilidad de “aprender a aprender” (Lovos, 2015). Esto como consecuencia de la globalización, que ha llevado a “muchas empresas a realizar el desarrollo de sus productos de una manera distribuida, llevándose a cabo por diferentes equipos, e incluso desde diferentes países. Este nuevo paradigma de desarrollo se conoce como “Desarrollo Global del Software” (Herbsleb, 2001, citado en Vizcaíno, Valencia, Soto, García-Mundo & Piattini, 2016, p. 1).

Al presente se pueden destacar algunas iniciativas como las de la “Fundación Telefónica, como la “Comunidad de Programadores Robóticos” y “Comunidad Jóvenes Emprendedores” que reúne jóvenes con miras a propiciar el intercambio de experiencias provechosas para ellos y su formación y desarrollo” (Lacoa, Lacoa & Blai, 2016, p. 8). Esto permite un proceso de colaboración centrada en el aprendizaje. Adicional a lo anterior existen investigaciones que promueven la utilización de “juegos serios” que sirve de apoyo para la adquisición de algunos de los conocimientos y habilidades que son

necesarios en el Desarrollo Global del Software (Vizcaíno, Valencia, Soto, García-Mundo & Piattini, 2016).

“Desde el constructivismo social (Vygotsky, 1978) el aprendizaje no es un proceso puramente interno, sino un constructo social mediado por el lenguaje utilizado en el discurso social” (Cenich, 2014, p. 4). Desde esta concepción de aprendizaje se da prioridad a la interacción como un mecanismo para el desarrollo de las potencialidades del estudiante, en este sentido, “la idea de Vygotsky (1979) de que la interacción con alguien más experto permite alcanzar más o mejores aprendizajes es denominado por el autor, la Zona de Desarrollo Próximo (ZDP)” (Astudillo, Bast & Willging, 2016), la cual se define como “la distancia entre el nivel real de desarrollo, determinado por la capacidad real de resolver independientemente un problema, y el nivel de desarrollo potencial determinado a través de la resolución de un problema bajo la guía de un adulto o en colaboración de un compañero más capaz” (Vygotsky, p. 131). En este sentido, estudios como el realizado por Anderson & Gegg-Harrison (2013) se fundamentan en la ZDP planteada por Vygotsky (1979) y desarrollan una metodología de enseñanza para el aprendizaje inicial de la programación de computadoras. Para esto, Anderson & Gegg-Harrison (2013) incorporan material educativo en los cursos introductorios que se encuentran dentro de la intersección de la "Zona de desarrollo próximo" (que contiene conceptos que el alumno es capaz de entender) y "la zona de Confort " (que contiene conceptos que motivan al alumno y se presentan de una manera que el alumno conoce y con la cual se siente cómodo). A esta zona de intersección los autores la denominan "Zona de confort de desarrollo próximo", esta se basa en un entorno de aprendizaje

colaborativo donde los conceptos de programación se introducen con aplicaciones de juegos.

De allí que, “los espacios construidos en la interacción con profesores y compañeros de estudio serían en los que tendrían lugar los procesos de construcción de significados y de atribución de sentido a los contenidos” (Cenich, 2014, p. 5). De esta forma, el aprendizaje a través de estos ambientes permite “el desarrollo de habilidades a nivel individual y grupal que tienen lugar a partir de la discusión que se genera entre los alumnos de un grupo de trabajo al momento de explorar nuevos conceptos” (Lucero, 2003, citado en Lovos, 2015, p. 15). Sin embargo, de acuerdo con la clasificación realizada por Coll de comunidades de aprendizaje (s.f. citado en Morales, Chaclán, Maldonado, Sontay, Montenegro & Magzul, 2013, p. 8) estas pueden ser referidas al aula, a la escuela o centro educativo, ciudad, comarca, región o zona territorial e incluso comunidades de aprendizaje que operan en un entorno virtual. Esta última categoría es muy utilizada en el aprendizaje de la programación de computadoras, mediante la participación de foros de discusión en páginas web en la cuales acuden programadores de distintas partes del mundo para hacer preguntas y compartir su conocimiento.

En este sentido, la participación en equipos virtuales de programadores permite a los expertos en codificación estar más preparados e incluso más conscientes de los problemas que pueden aparecer al mezclar un determinado código con otro (Elbelman, 2018). De esta forma, el enfoque sociocultural de Vygotsky (1979) demuestra ser importante para el desarrollo del aprendizaje inicial de la programación de computadoras (Kasto, 2016). Por lo tanto, no se trata solo del conocimiento en sí, sino principalmente

de la capacidad de adquirir habilidades de pensamiento crítico para resolver problemas (Elbelman, 2018).

### **Estudios relacionados**

Algunas investigaciones que puedo destacar respecto al tema son las siguientes:

En orden cronológico, Lahtinen, Ala-Mutka & Järvinen en el año 2005, estudian las dificultades que enfrentan los estudiantes en el aprendizaje inicial de los lenguajes de programación, mediante la realización de una encuesta internacional de opiniones por más de 500 estudiantes y profesores, en el marco del proyecto Codewitz ([www.codewitz.net](http://www.codewitz.net)), enfocado en la búsqueda de soluciones que beneficien la enseñanza y el aprendizaje de la programación de computadoras. Particularmente en los temas de: conceptos de programación introductoria y lenguaje de estructuras (e.j. variables, bucles y declaraciones condicionales). El cuestionario utilizado en su investigación fue desarrollado por los autores, el cual consistió en tres secciones: *background*, contenidos del curso y aspectos de aprendizaje. Este cuestionario fue aplicado a estudiantes y profesores, con el objetivo de comparar las diferencias en sus concepciones.

La primera sección *background*, incluía aspectos como: año de estudios, experiencia en programación antes de la universidad, habilidades informáticas, lenguajes de programación utilizado, esto con el fin de comparar si el trasfondo tiene impacto en las dificultades de aprendizaje y de igual manera el lenguaje de programación estudiado.

La sección dos, se enfocaba en descubrir las dificultades en el aprendizaje de los contenidos del curso. Esta sección contenía dos partes: los problemas en la construcción de un programa y los conceptos de programación. Esta sección se presentó como una

escala tipo Likert, con opciones de respuesta: muy fácil de aprender (1) a muy difícil (5). Incluyendo la opción no lo sé.

La tercera sección contenía preguntas sobre el proceso de aprendizaje. El propósito de esta sección era conocer qué tipo de situaciones y materiales de aprendizaje los estudiantes encuentran más efectivos. En este caso las opciones de respuesta fueron al igual que la anterior sección del 1 al 5, pero variando para las situaciones de aprendizaje: 1 (nunca aprendo en este tipo de situaciones) y 5 (aprendo siempre); para los ítems asociados a los materiales educativos utilizados, se ofreció las opciones de respuestas desde 1 (prácticamente inútil) y 5 (muy útil).

El cuestionario fue completado por estudiantes y profesores de 6 universidades: Fachhochschule Furtwan-gen (FHF, Alemania), Universidad de Reykjavik (RU, Islandia), TamperePolitécnico (TPU, Finlandia), Universidad Tecnológica de Tampere (TUT, Finlandia), Universidad Tecnológica de Bucarest (UTCB, Ro-manía), y la Universidad Tecnológica de Ventspils (VENTA, Letonia). El cuestionario estuvo disponible durante 10 días. En total, 559 estudiantes y 34 profesores respondieron la encuesta.

Dentro de los hallazgos de este estudio se encuentra que el mayor problema de los programadores novatos no parece ser la comprensión de conceptos básicos, sino su aplicación. En los resultados de la encuesta tanto estudiantes como profesores, acordaron que el aprendizaje práctico, es el más adecuado para asimilar los conceptos por lo que los autores concluyen que aprender haciendo debe ser parte de los estudios de programación de computadoras todo el tiempo. Es importante mencionar que en el informe de esta investigación no se menciona la recopilación de evidencias de la validez de las

interpretaciones y usos de las puntuaciones para el instrumento utilizado, lo cual es un aspecto importante porque “la validez en su acepción más simple significa la certeza con que un instrumento mide lo que se propone medir” (Medina- Díaz, 2013, p. 1).

Basados en el estudio anterior, Santimateo, Nuñez & González, en el 2018, publicaron una investigación, que tuvo como objetivo analizar la percepción de los estudiantes respecto a la dificultad para aprender a programar, la utilidad de los recursos y ambientes de aprendizaje. La población objeto de estudio fueron estudiantes de las carreras de licenciaturas en informática de la Universidad de Panamá, Universidad Tecnológica de Panamá, Universidad Latina de Panamá. Para el desarrollo de esta investigación se diseñó un instrumento aplicado en el periodo 2015-2016 a estudiantes que han cursado por lo menos un semestre de programación de computadoras.

El cuestionario contó con cinco secciones. La primera parte información de los encuestados, las dos siguientes corresponden a las actividades y conceptos usualmente utilizados en un curso de programación con la escala (muy difícil=1, difícil=2, normal=3, fácil=4, muy fácil=5) y las dos últimas referidas a la utilidad de los ambientes de aprendizajes, usando la escala (nunca=1, algunas veces=2, normal=3, casi siempre=4, siempre=5) y recursos para aprender con la escala (nada útil=1, algo útil=2, normal=3, útil=4, muy útil=5) (Santimateo, Nuñez & González, 2018).

Con relación a las evidencias de la validez de las interpretaciones y usos de las puntuaciones para el instrumento utilizado, se reporta el cálculo del coeficiente de consistencia interna alfa de Cronbach dando como resultado 0.92. Valor que de acuerdo con George and Mallery (2003, según se cita en Gliem & Gliem, 2003) es considerado

excelente. Sin embargo, esta es solo una de las cinco posibles fuentes de evidencia de validez.

Como resultados de esta investigación se expresa que en la “percepción general de los estudiantes, los recursos y los ambientes de aprendizajes representan un nivel aceptable de utilidad, se distinguen los ejemplos de programas, los materiales visuales y los tutoriales web como los recursos de mayor utilidad para los estudiantes” (p. 17).

“Respecto a las actividades de programación, los estudiantes perciben que lo más difícil de aprender se centra en la división de un programa en módulos, muy asociado con la creación y uso de funciones” (p. 17). “En cuanto a los conceptos propios de la programación de computadoras, los más difíciles de aprender son: los punteros, la recursividad y el manejo de objetos” (p. 17).

Por otra parte, Bruno (2005) plantea que la enseñanza inicial de la programación requiere de un gran esfuerzo. Esto debido a que los alumnos no están habituados al uso de la lógica para la resolución de problemas.

### **Cuestionarios de autoeficacia y motivación para el aprendizaje inicial de programación de computadoras**

Korkmaz & Altun (2014) desarrollaron una investigación cuantitativa que tuvo como objetivo detectar la autoeficacia de los estudiantes, hacia el aprendizaje del lenguaje de programación C ++. La población objeto de estudio fueron estudiantes de los departamentos de ingeniería eléctrica y electrónica e ingeniería informática de cuatro universidades diferentes. La muestra quedó constituida por 378 estudiantes. Para recopilar la información, los autores realizaron una traducción del idioma turco al inglés,

y adaptación de la escala *Computer Programming Self-Efficacy Scale* (CPSES), diseñada por Ramalingam y Wiedenbeck (1998, como se cita en Korkmaz & Altun, 2014). El instrumento quedó conformado por 28 ítems. Las respuestas se ofrecieron en una escala de tipo Likert de 7 puntos que varía de 1 (no estoy seguro en absoluto) a 7 (absolutamente seguro). Como evidencias de validez de las puntuaciones y uso de las interpretaciones, los autores recopilan juicios de expertos para el proceso de traducción del cuestionario. Posteriormente realizan el cálculo del coeficiente de confiabilidad Alpha de Cronbach obteniendo un valor de 0.966.

Al final los autores reportan en sus hallazgos que, los niveles de percepción de autoeficacia de los estudiantes no difirieron con respecto al género y que los niveles de percepción de autoeficacia de los estudiantes del Departamento de Ingeniería Eléctrico-Electrónica fueron significativamente más bajos que los del Departamento de Ingeniería Informática.

Por su parte, Davidson, Larzon & Ljunggren (2010) realizaron una investigación con el objetivo de estudiar si existe diferencia estadísticamente significativa en autoeficacia entre los estudiantes que inician en el aprendizaje de la programación de computadoras y los que tienen un año de haber iniciado, de igual manera respecto a qué tan bien se consideran en el cumplimiento de los objetivos específicos del curso de programación. La población objeto de estudio estuvo conformada por dos grupos de estudiantes, uno que actualmente tomaba el curso de introducción a la programación y otro que había tomado el curso hace un año.

Los datos se recopilaron mediante una encuesta en línea, que incluía las mismas preguntas utilizadas por Askar & Davenport (2009) y Ramalingam & Wiedenbeck

(2004). Por tanto, la escala de autoeficacia quedó conformada por 32 preguntas, las cuales se agruparon en 4 categorías para indicar diferentes tipos de habilidades: factor 1 (independencia y persistencia), factor 2 (tareas de programación complejas), factor 3 (autorregulación) y factor 4 (tareas de programación simples).

Al final los autores indican en los hallazgos que, al considerar la autoeficacia como un todo, es decir, mirando el puntaje promedio sobre las 32 preguntas para ambos grupos de estudiantes. Esta comparación no muestra diferencia estadísticamente significativa, entre los dos grupos de estudiantes. Una mirada más cercana a los diferentes grupos de preguntas muestra un aumento estadísticamente significativo en la autoeficacia en la categoría 3 (Autorregulación), mientras que las otras categorías (Independencia, tareas complejas, tareas simples) no muestran cambios significativos.

Respecto al cumplimiento de los objetivos del curso de programación, hay tres habilidades en particular donde los estudiantes informan un aumento significativo durante el año posterior al curso introductorio: 1) analizar y diseñar soluciones para problemas simples, 2) depuración sistemática y 3) explicar los principios generales de cómo se estructura y funciona una computadora.

Law, Lee & Yu (2010) desarrollaron un estudio que tuvo como propósito explorar los posibles vínculos entre los factores motivadores clave y la eficacia de los estudiantes en el aprendizaje de programas informáticos, en cursos de programación de computadoras respaldados por un sistema de aprendizaje electrónico: sistema de evaluación de asignación de programación (PASS). Partiendo de que un estudiante con un alto nivel de eficacia puede desempeñarse bien en el aprendizaje de la programación de computadoras.

La población de estudio fueron estudiantes universitarios que habían tomado dos cursos de Programación de computadoras. Para la recopilación de la información Law, Lee & Yu (2010) elaboraron un cuestionario partiendo del diseñado por Law et al. (2009). Los autores como evidencias de validez de las puntuaciones y usos de las interpretaciones realizaron un estudio piloto y análisis factorial exploratorio de rotación oblicua. De igual manera, realizaron el cálculo del coeficiente de confiabilidad Alpha de Cronbach de 0.95.

Después de esto, la versión final del cuestionario quedó conformado por dos partes. La primera parte pide información demográfica como edad y género. La segunda parte, consta de 20 preguntas (ítems) que permiten la identificación de factores que tienen un efecto motivador positivo en el aprendizaje y cinco elementos para determinar el efecto electrónico percibido y la eficacia de los estudiantes.

Como partes de los hallazgos con pertinencia al presente estudio, los autores respetaron que, en primer lugar, entre los seis factores identificados, 'recompensa y reconocimiento' tienen el mayor efecto motivador en el aprendizaje. En segundo lugar, tres factores motivadores, a saber, 'actitud individual y expectativa', 'metas desafiantes' y 'presión social y competencia', presentaron una relación significativa y positiva con la eficacia. Los resultados sugieren la necesidad de un entorno de apoyo para el aprendizaje de la programación de computadoras debe involucrar mecanismos de recompensa, expectativa y objetivos claros donde se espera presión social. El factor intrínseco "actitud y expectativa individual" parece destacarse tanto por su fuerte motivación como por su alta correlación con eficacia.

Además, tanto los factores intrínsecos en estudio, a saber, 'actitud individual y expectativa' y 'metas desafiantes', son visiblemente correlacionado con la eficacia, mientras que, para los factores extrínsecos, solo se considera que la "presión social y la competencia" son significativas. Estos resultados plantean un desafío para el educador acerca de cómo organizar las actividades de enseñanza y aprendizaje para reforzar efectivamente estos factores motivadores en beneficio de mejorar la eficacia del aprendizaje de los estudiantes.

## **CAPÍTULO III**

### **MÉTODO**

#### **Introducción**

En este capítulo se incluye la metodología utilizada para la recopilación de los datos, el tipo de diseño seleccionado y las razones para su elección. De igual forma, las herramientas usadas en la etapa de la recopilación de la información, el tipo de muestreo y los requisitos para participar del estudio. Al final, se presenta el método estadístico utilizado en este estudio para el análisis de la información.

En el presente estudio se investigó, los factores determinantes en el aprendizaje de la programación de computadoras. Respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras. Además, se indagó sobre factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos, desde la perspectiva de los estudiantes de Ciencia de Cómputos que han realizado el curso de introducción a la programación.

#### **Preguntas de investigación**

Las preguntas que guiaron esta investigación son las siguientes:

1. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos respecto al nivel de dificultad de los contenidos del curso Introducción a la Programación de Computadoras?

2. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los ambientes más efectivos para el aprendizaje de la programación?
3. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los recursos educativos más efectivos para el aprendizaje de la programación?
4. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a su nivel de motivación para el aprendizaje de la programación?
5. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos de su nivel de autoeficacia para el aprendizaje de la programación de computadoras?
6. ¿Cuál es la correlación entre la motivación de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras?
7. ¿Cuál es la correlación entre la autoeficacia de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras?

### **Metodología**

El método de investigación es uno de naturaleza cuantitativa. Según Creswell (2012) en la investigación cuantitativa, el investigador identifica un problema de investigación basado en tendencias en el campo o sobre la necesidad de explicar por qué ocurre algo. Por lo tanto, este método de investigación es utilizado para describir una

tendencia general de las respuestas de los individuos y observar cómo esta tendencia varía entre las personas. De igual manera, permite establecer relaciones entre variables y explicar las causas de los cambios en los hechos sociales medidos (McMillan & Schumacher, 2005). En este tipo de investigación se recogen datos cuantificables, se analizan los datos estadísticamente y se trata de conducir la investigación de manera objetiva (Creswell, 2008). Esta metodología fue seleccionada porque permite conocer desde la perspectiva de los estudiantes subgraduados de Ciencia de Cómputos, sobre algunos factores internos tales como: el nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras, y además algunos factores externos tales como: contenidos del curso, ambientes y recursos de aprendizaje más efectivos. De igual forma, el enfoque cuantitativo, permite crear declaraciones de propósito, preguntas de investigación e hipótesis que son específicas, medibles y observables. También permite la recopilación de datos numéricos de un gran número de personas al usar instrumentos con preguntas y respuestas preestablecidas (Creswell, 2012).

### **Diseño de investigación**

De acuerdo con el propósito de la presente investigación se propuso la utilización de un diseño de investigación por encuesta. De acuerdo con Groves et al. (2009), los diseños de encuesta proveen un método sistemático para recopilar información de (una muestra de) entidades con el propósito de desarrollar descriptores cuantitativos (estadísticas) de las características de la población a la cual pertenecen las entidades. Dado que esta investigación indagó sobre la percepción de los estudiantes subgraduados del programa de Ciencia de Cómputos, la encuesta realizada fue de tipo transversal, con los datos recogidos en un punto en el tiempo (Creswell, 2008). En este sentido, el diseño

de una encuesta transversal debe considerar aspectos relacionados con la población que se estudiará, los sujetos de quienes se obtendrá información y la información que se busca captar (Hernández & Velasco-Mondragón, 2000).

### **Población y selección de participantes**

La población objeto de estudio estuvo conformada por los estudiantes del programa de Ciencia de Cómputos, de la Facultad de Ciencias Naturales de la Universidad de Puerto Rico, Recinto de Río Piedras (UPRRP). Este programa incluye en el segundo semestre, la asignatura CCOM 3033: Introducción a la Programación de Computadoras (3 créditos), esta asignatura tiene como prerrequisitos: CCOM 3030: Introducción a la Ciencia de Cómputos y MATE 3023: Precálculo I.

A continuación, se presenta la matrícula total del programa de Bachillerato de Ciencia de Cómputos de acuerdo con el primer semestre del año académico 2020-2021 (Ver Tabla 1). Según su género y año académico. Esta información es reportada por la División de Investigación Institucional y Avalúo de la UPRRP. La siguiente información, corresponde al semestre en que fue aplicado el cuestionario en esta investigación.

Tabla 1.

*Matrícula total del programa de Bachillerato de Ciencia de Cómputos primer semestre del año académico 2020-2021. Se presenta clasificación, según su género y año académico.*

Género	M	F	T
Primer año	28	8	36

*Notas:* M(Masculino), F(Femenino) y T(Total).

Tabla 1 (Continuación).

*Matrícula total del programa de Bachillerato de Ciencia de Cómputos primer semestre del año académico 2020-2021. Se presenta clasificación, según su género y año académico.*

Género	M	F	T
Segundo año	34	16	50
Tercer año	24	2	26
Cuarto año	23	5	28
Otro	3	1	4
Transitorio	2	1	3
Total	114	33	147

*Notas:* M(Masculino), F(Femenino) y T(Total).

Los requisitos para participar de esta investigación fueron: ser estudiante del programa subgraduado de Ciencia de Cómputos de la UPRRP, y que hayan tomado el curso: CCOM 3033: Introducción a la Programación de Computadoras. Todos los estudiantes que cumplían con estos requisitos fueron invitados a participar. Por tanto, para este estudio se realizó un muestreo no probabilístico por conveniencia. Esto implica que los participantes fueron seleccionados por su disponibilidad y voluntad para formar parte del estudio (Creswell, 2012). De ahí que, la muestra quedó conformada por un grupo de estudiantes “seleccionados sobre la base de ser accesibles o adecuados” (McMillan & Schumacher, 2005, p. 140), que cumplieron con los requisitos antes descritos y que de manera voluntaria respondieron el cuestionario desarrollado para la presente investigación. Como parte del proceso de selección de participantes, se solicitó al Director del Departamento de Ciencia de Cómputos, enviar mediante correo electrónico la invitación a participar del presente estudio, a todos los estudiantes activos

del departamento de Ciencias de Cómputo que cumplieran con los requisitos de esta investigación.

### **Procedimiento**

Antes de iniciar la investigación se completaron los siguientes requisitos institucionales: Inicialmente obtuve el Certificado del curso en línea sobre la protección de los derechos humanos ofrecidos por *CITI Program (Ver Apéndice A)*. Seguidamente recibí la aprobación de la propuesta de investigación por parte del comité de disertación, y posteriormente conseguí el consentimiento para iniciar la investigación de parte del Comité Institucional para la Protección de los Seres Humanos en la Investigación (CIPSHI) (*Ver apéndice B*).

Después de cumplir con los requisitos antes descritos, solicité la autorización para el desarrollo de la investigación al director del Departamento de Ciencia de Cómputos de la UPRRP. Inicialmente en la presente investigación se consideró la aplicación del cuestionario de forma presencial. Sin embargo, a causa de la pandemia producto del virus COVID-19 la UPRRP desde marzo de 2020 cambio sus clases de la modalidad presencial a virtual. Por tal motivo, se solicitó y se recibió aprobación de CIPSHI para modificar el protocolo de investigación. Con el fin de cambiar el modo de administración del cuestionario de impreso a en línea mediante *Google forms*.

Mediante correo electrónico los potenciales participantes recibieron un enlace que los dirigió a una hoja de consentimiento informado (*Ver Apéndice C*). En esta hoja, se les mencionó a los posibles participantes el propósito de la investigación, los riesgos y beneficios relacionados al estudio. Se hizo hincapié en que su participación en la

investigación era voluntaria. De igual manera, se informó que la información provista sería manejada de manera confidencial. También que las respuestas del cuestionario serán conservadas permanentemente, por el investigador, en un banco de datos sin información que pueda identificar a una persona. Al final, cada participante, después de leer la hoja de consentimiento, debía seleccionar si aceptaba o no colaborar con la presente investigación.

### **Diseño y desarrollo del instrumento**

Para fines de la presente investigación se utilizó un cuestionario, el cual fue auto-administrado en línea mediante *Google forms*. Por lo tanto, el mismo participante contestó el cuestionario (Medina-Díaz, 2010). Este, estaba conformado por preguntas cerradas. El participante seleccionó la respuesta que mejor se ajustaba a su experiencia adquirida en el curso: Introducción a la programación de computadoras.

El cuestionario utilizado fue desarrollado para la presente investigación. Las secciones que se incluyeron fueron: Parte I (aspectos sobre los contenidos del curso), Parte II (los ambientes en los que aprende programación), Parte III (utilidad de los Recursos de aprendizaje), Parte IV (la motivación para el aprendizaje de la programación), Parte V (la autoeficacia para el aprendizaje de la programación), y Parte VI (aspectos sociodemográficos) (*Sexo, Tipo de escuela de procedencia, edad, que calificación obtuvo en el curso de fundamentos de programación y si tiene experiencia previa en el aprendizaje de la programación de computadoras*). Esto en respuesta a las preguntas formuladas en la presente investigación. La parte IV Motivación para el aprendizaje de la programación. Se fundamentó en la teoría de la autodeterminación (TDA) de Ryan y Deci (2000). En particular los ítems se asocian a las tres necesidades

básicas que expone la TDA (Autonomía, Relación con los demás y Competencia) como factores determinantes para el desarrollo de la motivación intrínseca. En este sentido, los ítems que conforman esta parte están asociados a cada una de las tres necesidades básicas, como se especifican, en la Tabla 2.

Tabla 2.

*Distribución de los ítems de la sección IV, para las tres necesidades básicas (TDA)*

Necesidad básica según TDA (Ryan y Deci, 2000)	Nº de ítem
Autonomía:	1,4,7,10,13,16
Relación con los demás:	2,5,8,11,14,17
Competencia:	3,6,9,12,15,18

La parte V. Autoeficacia para el aprendizaje de la programación. Esta sección del cuestionario se basó en la teoría de la autoeficacia de Bandura (1977), siendo la autoeficacia, la percepción que tienen los estudiantes de Ciencia de Cómputos de su propia capacidad para aprender programación de computadoras. Algunos ítems que conformaron esta sección fueron incluidos después de realizar la revisión de literatura. En esta revisión, se pudo encontrar algunas investigaciones fundamentadas en la teoría de la autoeficacia de Albert Bandura y que desarrollaron cuestionarios para la recolección de la información en poblaciones similares a la de la presente investigación. En el caso de la utilización de ítems textuales o adaptación de algunos de ellos. Se sometió a autorización de sus correspondientes autores.

### **Evidencias de validez**

“La validez es el grado al cual la evidencia y la teoría apoyan las interpretaciones de las puntuaciones de una prueba que se utilizará con un propósito dado” (*American Educational Research Association, American Psychological Association & National Council on Measurement in Education*, 2014, p. 11). De allí, la importancia de incluir en el presente estudio, la recopilación de evidencias de la validez de los datos, siendo que, aun teniendo:

Todos los cuidados en la construcción del cuestionario, cabe la posibilidad de que la validez se afecte por el contenido y el vocabulario de las preguntas, el formato y la amplitud las respuestas, así como la disposición de las personas a contestar de manera honesta y otros factores (Medina-Díaz, 2010, p. 152)

Los *Standards for Educational and Psychological Testing* (AERA et al., 1999) identifican cinco tipos de fuentes de evidencia para la validez: evidencia relacionada con el contenido del instrumento, evidencia basada en el proceso de respuesta, evidencia basada en la estructura interna del instrumento, evidencia basada en la relación a otras variables y evidencia basada en las consecuencias del uso del instrumento. En la presente investigación se recopilaron las siguientes evidencias de validez:

**Evidencia basada en el contenido.** Esta se ocupa de determinar la representatividad y la relevancia de las preguntas para recoger datos acerca del constructo de interés (Medina-Díaz, 2010). Para este fin se sometió el instrumento a juicio de expertos. Este grupo de expertos estaba constituido por un conocedor del área de medición y construcción de instrumentos. Un experto del área de Ciencia de Cómputos.

En este caso, se solicitó a un profesor del Programa del Bachillerato de Ciencia de Cómputos de la UPRRP. Esta elección se basó en que, además de poseer el conocimiento específico, debían poseer la experiencia enseñando a la población objeto de estudio. Un tercer miembro del grupo de experto lo integró un profesional del área de Psicología o Consejería, conocedor de las teorías de Motivación y Autoeficacia para el aprendizaje. Por último, un cuarto experto fue del área de Tecnología del Aprendizaje.

Los expertos fueron contactados en persona o por correo electrónico. Una vez confirmaron su intención de participar en el proceso de validar el contenido del instrumento en general, se les envió mediante correo electrónico los documentos necesarios para la evaluación del instrumento. Estos documentos consistían en el cuestionario y una Planilla para la evaluación del contenido del instrumento. En esta planilla, los expertos incluyeron sus comentarios y recomendaciones con relación a: (a) claridad en la redacción, (b) comprensión de las preguntas, (c) relevancia con los componentes.

**Evidencia basada en el proceso de respuesta.** “Esta evidencia requiere examinar los procesos mentales o de razonamiento que las personas aplican para contestar los ítems del cuestionario” (Medina-Díaz, 2010, p. 154). En la presente investigación se realizaron entrevistas cognitivas. El objetivo principal es corroborar que las preguntas del instrumento sean comprensibles, que las respuestas se ajusten a lo que el sujeto quiere decir y que las respuestas poseen medidas válidas para lo que se desea medir (Fowler, 2009). Para este fin se contó con la participación de cinco estudiantes, del programa de Ciencia de Cómputos de la UPRRP. De esta forma, se pudo auscultar las demandas

cognitivas que conllevan las preguntas e instrucciones del cuestionario. De igual manera, los diferentes procesos que los participantes realizan para contestar (Medina-Díaz, 2010).

**Evidencia basada en la estructura interna.** Esta evidencia permite determinar si las personas que están en una situación similar respecto al constructo o el asunto bajo estudio contestan el conjunto de ítems provisto de la misma manera (Medina-Díaz, 2010). Para determinar la consistencia interna se calculó el coeficiente de confiabilidad Alfa de Cronbach. Este coeficiente es un método que se utiliza en instrumentos en los que no existe respuesta correcta o incorrecta en cada reactivo (McMillan, 2004). Como referente de los valores obtenidos del coeficiente de confiabilidad se utilizaron las categorías propuestas por Gliem y Gliem (2003). Estos autores indican que un valor de .90 o más es considerado excelente, .80 a .89 es considerado bueno, .70 a .79 es aceptable, .60 a .69 es considerado cuestionable, .50 a .59 es considerado pobre y .50 o menor es un valor no aceptable.

### **Análisis de datos**

Los resultados de esta investigación fueron analizados con el programado *Statistical Package for the Social Sciences* (SPSS) versión 23. Se utilizaron estadísticas descriptivas tales como medidas de tendencia central, frecuencias y porcentajes, con el propósito de examinar las respuestas y cómo estas fueron distribuidas según los ítems del cuestionario. Además, se realizó un análisis de correlación de Spearman para establecer la relación entre la motivación de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso Introducción a la Programación de Computadoras. Este análisis correlacional permitió contestar la sexta pregunta de investigación. De forma similar para atender la séptima pregunta de investigación se realizó un análisis de

correlación de Spearman entre la subescala de autoeficacia y el rendimiento académico en el curso Introducción a la Programación de Computadoras, informado por los participantes.

### **Aspectos éticos**

Para atender todos los requisitos del Comité para la protección de los seres Humanos en la Investigación (CIPSHI), en la presente investigación se incluyeron los siguientes aspectos. Primero la protección de los participantes de cualquier tipo de daño. Para esto se elaboró una hoja de consentimiento informado para los participantes. Dichas hojas cumplieron con lo establecido por CIPSHI. De esta forma, el participante antes de aceptar colaborar con la investigación era informado de que podía retirarse de la investigación en cualquier momento. De igual forma, las preguntas del cuestionario no eran de respuesta obligatoria y el participante podía no contestar según su voluntad.

Otro aspecto ético abordado fue la confidencialidad de los datos. Al respecto, toda la información recopilada de los participantes mediante el cuestionario se mantuvo en total confidencialidad. En este sentido, solamente el investigador y su director de disertación tuvieron acceso a los datos crudos, incluyendo la hoja de consentimiento informado. Como seguridad para acceder a las respuestas del cuestionario se asignó una contraseña conocida solo por el investigador. Las respuestas del cuestionario serán conservadas permanentemente, por el investigador, en un banco de datos sin identificadores.

De igual manera, se les indicó a los participantes, en la hoja de consentimiento informado, los riesgos y beneficios de formar parte de la presente investigación. En este

caso los riesgos potenciales asociados a la participación en esta investigación son mínimos, dado a que sólo se explorarán datos relacionados con la experiencia de los participantes en su aprendizaje en el curso de Introducción a la Programación de Computadoras. Por lo tanto, los posibles riesgos que podría experimentar un participante son cansancio y/o incomodidad ante una pregunta. Con relación a los beneficios, con este trabajo se espera contribuir al mejoramiento del proceso de enseñanza aprendizaje de la introducción a la programación. De esta forma, impactar positivamente a los estudiantes del programa de Ciencia de Cómputos.

## **CAPÍTULO IV**

### **RESULTADOS**

#### **Introducción**

El presente estudio se realizó con el propósito de investigar, los factores determinantes que inciden en el aprendizaje de la programación de computadoras. Respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras. Además, se investigó sobre factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos. En este capítulo se presentan los resultados obtenidos. En la primera parte se incluye la descripción de la muestra y los análisis descriptivos realizados para la sección de datos sociodemográficos. Posteriormente, se informan los resultados del Coeficiente de consistencia interna Alfa de Cronbach calculado y su categorización. De igual forma, se presentan los datos recopilados para dar respuesta a las preguntas de investigación que guiaron el estudio:

1. ¿Cuál es la percepción de los estudiantes subgraduados del programa de Ciencia de Cómputos respecto al nivel de dificultad de los contenidos del curso introducción a la programación de computadoras?
2. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los ambientes de aprendizaje más efectivos para la programación?

3. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los recursos educativos más efectivos para el aprendizaje de la programación?
4. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a su nivel de motivación para el aprendizaje de la programación?
5. ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos de su nivel de autoeficacia para el aprendizaje de la programación de computadoras?
6. ¿Cuál es la correlación entre la motivación de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras?
7. ¿Cuál es la correlación entre la autoeficacia de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras?

### **Datos sociodemográficos**

Los participantes en este estudio corresponden a una muestra por conveniencia de 35 estudiantes, del departamento de Ciencias de Cómputos, que han realizado el curso de introducción a la programación, pertenecientes a la Facultad de Ciencias Naturales del Recinto de Río Piedras de la Universidad de Puerto Rico.

Para la recolección de datos sociodemográficos se formularon en el cuestionario cinco preguntas. La primera pregunta indagaba sobre cuál es el sexo del participante. En la Tabla 3 se presenta la frecuencia y la distribución porcentual de los participantes de

acuerdo con su sexo. El análisis de esta variable muestra que 27 (77.1%) de las respuestas obtenidas en esta investigación fueron realizadas por hombres, mientras que 8 (22.9%) de las respuestas fueron brindadas por mujeres.

Tabla 3.

*Frecuencia y distribución porcentual de participantes por sexo*

Sexo	Frecuencia	Porcentaje
Hombre	27	77.1%
Mujer	8	22.9%
Total	35	100%

La segunda pregunta de esta sección apuntó a conocer en qué tipo de escuela superior terminó el participante, para responder a la pregunta se le presentaron las siguientes alternativas: Escuela pública, Escuela privada, Escuela especializada y *Home School*. En la Tabla 4 se muestra la frecuencia y distribución porcentual de los participantes por tipo de escuela. En estos resultados se destaca que la mayoría de los participantes 21(60%) provienen de Escuelas privadas. En un menor grado de Escuela pública 11(31.4%) y Escuela especializada 3(8.6%). Ninguno de los participantes señaló *Home school*.

Tabla 4.

*Frecuencia y distribución porcentual de los participantes por tipo de escuela*

Tipo de Escuela	Frecuencia	Porcentaje
Escuela pública	11	31.4%
Escuela privada	21	60.0%

Tabla 4 (continuación).

*Frecuencia y distribución porcentual de los participantes por tipo de escuela*

Tipo de Escuela	Frecuencia	Porcentaje
Escuela especializada	3	8.6%
<i>Home school</i>	0	0%
Total	35	100%

Otra variable abordada en esta sección del cuestionario correspondió a la edad de los participantes. En estos hallazgos (Tabla 5) se observa que la edad mínima informada por los participantes fue de 18 años mientras la edad máxima registrada fue de 23 años. El promedio de las edades registradas fue de 19.5 años y la mediana de 19 años.

Tabla 5.

*Frecuencia y distribución porcentual de los participantes por edad*

Edad	Frecuencia	Porcentaje
18	5	14.3%
19	15	42.9%
20	10	28.6%
21	2	5.7%
22	2	5.7%
23	1	2.9%
Total	35	100%

En la pregunta 4 de la sección de datos sociodemográficos, los participantes respondieron respecto a: ¿Qué calificación obtuvo en el curso de CCOM 3033 - Introducción a la Programación de Computadoras? La Tabla 6 muestra la frecuencia y distribución porcentual obtenida de los participantes. En estos resultados se destaca que

la mayoría de los participantes 19(54.3%) obtuvieron la calificación B. Seguidamente 8 participantes (22.9%) señalaron haber obtenido C y casi en igual medida (22.9%) ocho participantes seleccionaron la calificación A.

Tabla 6.

*Frecuencia y distribución porcentual de los participantes por calificación*

Calificación	Frecuencia	Porcentaje
C	8	22.9%
B	19	54.3%
A	8	22.9%
Total	35	100%

La última pregunta de esta sección del cuestionario pretendió conocer si el participante tenía conocimientos de programación antes de entrar al departamento de Ciencia de Cómputos. Para responder a esta pregunta el participante tenía dos alternativas de respuesta Si/No. La Tabla 7 muestra la frecuencia y distribución porcentual obtenida para estas alternativas. Se destaca que la mayoría de los participantes 34(97.1%) indicaron no poseer previamente conocimientos de programación. Solo un participante (2.9%) marcó la opción “Sí”.

Tabla 7.

*Frecuencia y distribución porcentual de los participantes por Conocimiento previo de programación*

Tenía conocimientos previos de programación	Frecuencia	Porcentaje
Si	1	2.9%
No	34	97.1%
Total	35	100%

### **Resultados del Coeficiente de consistencia interna Alfa de Cronbach**

“La validez es una cuestión del grado en que la teoría y la evidencia acumulada apoya las interpretaciones y las inferencias, derivadas de los datos y a tenor con el propósito del instrumento” (AERA, APA y NCME, 1999, citado en Medina-Díaz, 2010, p. 152). De allí la importancia de recopilar evidencias de validez de las interpretaciones y uso de los datos recogidos en el cuestionario utilizado en la presente investigación. Adicional a las evidencias de contenido y proceso de respuesta adquiridas en el proceso de construcción del cuestionario. También se recopiló evidencia de la estructura interna. Para este fin se obtuvo el coeficiente de confiabilidad de consistencia interna Alfa de Cronbach. Este coeficiente, “es el procedimiento más popular para estimar la consistencia interna de las respuestas de un conjunto de ítems” (Medina-Díaz, 2010, p. 155).

En la Tabla 8, se presentan los valores del coeficiente de consistencia interna Alfa de Cronbach calculado para cada una de las subescalas del cuestionario y de la escala en general. George & Mallery (2003) (como se cita en Gliem & Gliem, 2003) indican que un valor de Alfa de Cronbach de .90 o más es considerado excelente, .80 a .89 es

considerado bueno, .70 a .79 es aceptable, .60 a .69 es considerado cuestionable, .50 a .59 es considerado pobre y .50 o menor es un valor no aceptable. Según estos criterios, la subescala “Contenidos del curso” arrojó un Alfa de Cronbach de .937 esta se encuentra dentro del rango “Excelente”. La subescala “Ambientes en los que aprende programación” arrojó un Alfa de Cronbach de .839 se encuentra dentro del rango “Bueno”. La subescala “Utilidad de los recursos” arrojó un Alfa de Cronbach de .814 se encuentra dentro del rango “Bueno”. La subescala “Motivación para el aprendizaje de la programación” arrojó un Alfa de Cronbach de .901 se encuentra dentro del rango “Excelente”. La subescala Autoeficacia para el aprendizaje de la programación” arrojó un Alfa de Cronbach de .958 se encuentra dentro del rango “Excelente”.

Tabla 8.

*Coefficiente de consistencia interna Alfa de Cronbach calculado y su categorización (George & Mallery, 2003, como se cita en Gliem & Gliem, 2003).*

	Escala	Nº de ítems	Alfa de Cronbach	George & Mallery (2003, como se cita en Gliem y Gliem, 2003)
Aspectos generales relacionados con	Contenidos del curso	12	.937	Excelente
	Ambientes en los que aprende programación	9	.839	Bueno

Tabla 8 (continuación).

*Coefficiente de consistencia interna Alfa de Cronbach calculado y su categorización*

*(George & Mallery, 2003, como se cita en Gliem & Gliem, 2003).*

	Escala	Nº de ítems	Alfa de Cronbach	George & Mallery (2003, como se cita en Gliem y Gliem, 2003)
Aspectos generales relacionados con	Utilidad de los recursos	11	.814	Bueno
	Motivación para el aprendizaje de la programación	18	.901	Excelente
	Autoeficacia para el aprendizaje de la programación	15	.958	Excelente

### **Resultados recopilados por pregunta de investigación:**

#### **Contenidos del Curso**

La sección I del cuestionario se desarrolló para atender la primera pregunta de investigación: ¿Cuál es la percepción de los estudiantes subgraduados del programa de Ciencia de Cómputos respecto al nivel de dificultad de los contenidos del curso introducción a la programación de computadoras? Por lo tanto, las aseveraciones en esta sección se relacionan con los contenidos del curso de la asignatura Introducción a la Programación CCOM 3033. En este curso los estudiantes comienzan a utilizar estructuras

lógicas y demás elementos de los fundamentos de programación. En el cuestionario se pidió a los participantes que marcaran sus respuestas de acuerdo con la alternativa que mejor representó su nivel de dificultad en el aprendizaje de esos contenidos. La subescala “contenidos del curso” está compuesta por 12 ítems con una escala ordinal y las siguientes alternativas de respuesta: Muy fácil, Fácil, Normal, Difícil, Muy Difícil y No deseo contestar.

En la Tabla 9 se presentan los resultados obtenidos en esta sección del cuestionario. A continuación, se mencionarán los ítems con resultados relevantes como lo fueron los ítems 1, 2 y 3 que se refieren a los contenidos: reconocer la estructura de un programa, definición de variables y uso de condicional *if*. Estos ítems obtuvieron la mayoría de las respuestas en las alternativas Muy fácil o Fácil con los siguientes porcentajes 65.7%, 88.6% y 57.1% respectivamente.

En contraste, los contenidos en los cuales los participantes informaron tener mayor dificultad (al sumar las dos alternativas Muy difícil y Difícil). Se destacan los ítems 7, 8, 9 y 10. El ítem 7 que corresponde a: “representar soluciones mediante abstracciones”, obtuvo un 71.4%. El ítem 8 el cual incluye el tema: “dividir problemas extensos de programación en subproblemas”, arrojó 62.8%. El ítem 9 (funciones algorítmicas) fue seleccionado por el 62.8% de los participantes. Por último, el ítem 10 que indaga sobre la búsqueda de errores en un programa tuvo un 65.7%.

Tabla 9.

*Frecuencia y distribución porcentual de los aspectos relacionados a los contenidos del curso.*

Ítem	Muy Fácil $f(P)$	Fácil $f(P)$	Normal $f(P)$	Difícil $f(P)$	Muy Difícil $f(P)$
Item 1	3(8.6)	20(57.1)	7(20.0)	5(14.3)	
Item 2	10(28.6)	21(60.0)	2(5.7)	2(5.7)	
Item 3	10(28.6)	10(28.6)	9(25.7)	6(17.1)	
Item 4	9(25.7)	6(17.1)	6(17.1)	14(40.0)	
Item 5	10(28.6)	3(8.6)	7(20.0)	12(34.3)	3(8.6)
Item 6	6(17.1)	6(17.1)	8(22.9)	13(37.1)	2(5.7)
Item 7	6(17.1)	3(8.6)	1(2.9)	17(48.6)	8(22.9)
Item 8		6(17.1)	7(20.0)	12(34.3)	10(28.6)
Item 9	2(5.7)	2(5.7)	9(25.7)	17(48.6)	5(14.3)
Item 10			12(34.3)	10(28.6)	13(37.1)
Item 11	2(5.7)	8(22.9)	12(34.3)	10(28.6)	3(8.6)
Item 12	4(11.4)	3(8.6)	12(34.3)	7(20.0)	9(25.7)

*Notas.*  $f$  = frecuencia.  $P$  = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo.

### **Ambientes en los que se aprende programación**

Otra de las secciones del cuestionario es “Ambientes en los que se aprende programación” compuesta por 9 ítems. Las alternativas de respuesta son: Siempre, Casi siempre, A veces, Casi nunca y Nunca. Esta sección se desarrolló para atender la segunda pregunta de investigación: ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los ambientes de aprendizaje más efectivos para la programación?

Los resultados proporcionados por los participantes (Tabla 10) muestran una tendencia alta de aceptación a los distintos ambientes para el aprendizaje de programación planteados por esta subescala. De esta forma, del 88.5% al 62.8% fue el rango de respuestas que se ubicaron entre las alternativas Siempre o Casi siempre. El mayor porcentaje lo obtuvieron los ítems 4 y 5 los cuales se refieren al: (Aprendizaje de programación en clases prácticas como laboratorios y las clases teóricas) ambos ítems con 88.5%. En este mismo sentido podemos destacar el ítem 1 que trata sobre (el aprendizaje de programación en cursos virtuales) el cual obtuvo el 85.7%.

Los ítems 2, 3 y 8 mostraron resultados iguales o mayores al 80%. Los resultados más bajos fueron los ítems 6 y 7 con porcentajes por encima de 54%, sin embargo, es un resultado significativo para esta investigación.

Tabla 10.

*Aspectos relacionados a los ambientes en los que aprende programación*

Ítem	Siempre $f(P)$	Casi siempre $f(P)$	A veces $f(P)$	Casi nunca $f(P)$	Nunca $f(P)$
Item 1	17(48.6)	13(37.1)	3(8.6)	1(2.9)	1(2.9)
Item 2	16(45.7)	13(37.1)	3(8.6)	3(8.6)	
Item 3	11(31.4)	17(48.6)	3(8.6)	4(11.4)	
Item 4	17(48.6)	14(40)	2(5.7)	1(2.9)	1(2.9)
Item 5	18(51.4)	13(37.1)	3(8.6)		1(2.9)
Item 6	8(22.9)	11(31.4)	3(8.6)	10(28.6)	2(5.7)
Item 7	9(25.7)	13(37.1)	3(8.6)	7(20)	3(8.6)
Item 8	13(37.1)	15(42.9)	2(5.7)	3(8.6)	2(5.7)
Item 9	17(48.6)	10(28.6)	3(8.6)	1(2.9)	3(8.6)

*Notas.*  $f$  = frecuencia.  $P$  = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo.

### **Utilidad de los recursos**

La sección III del cuestionario se enfocó en indagar respecto a la utilidad de los recursos empleados para el aprendizaje de la programación de computadoras. Esto en respuesta a la tercera pregunta de investigación: ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los recursos educativos más efectivos para el aprendizaje de la programación?

En la Tabla 11 se presentan las respuestas de los participantes a esta sección del cuestionario. Estos datos se exponen, al igual que en las secciones anteriores, como datos agrupados en frecuencia y porcentaje para cada ítem. La subescala III: utilidad de los recursos, contiene 11 ítems con las siguientes cuatro alternativas de respuesta: Muy útil, Útil, Poco útil y Sin utilidad. Los resultados obtenidos señalan que la mayoría de los participantes seleccionaron las respuestas Muy útil o útil. Se destaca el ítem 6 (“Revisar tutoriales en línea sobre programación”) con un 97.1% de respuestas en las categorías Muy útil o útil. En estas mismas categorías de respuesta, los ítems 1, 2, 3, 4, 8 y 10 mostraron un rango de respuestas entre el 80% y 88.6%. Mientras que los ítems 9 y 11 tuvieron el porcentaje más bajo con 68.6% y 62.9% respectivamente.

Tabla 11

*Aspectos relacionados a utilidad de los recursos*

Ítem	Muy útil $f(P)$	Útil $f(P)$	Poco útil $f(P)$	Sin utilidad $f(P)$
Item 1	15(42.9)	14(40.0)	4(11.4)	2(5.7)
Item 2	16(45.7)	15(42.9)	2(5.7)	1(2.9)
Item 3	18(51.4)	12(34.3)	3(8.6)	2(5.7)
Item 4	16(45.7)	13(37.1)	4(11.4)	1(2.9)
Item 5	16(45.7)	11(31.4)	3(8.6)	4(11.4)
Item 6	21(60.0)	13(37.1)		1(2.9)
Item 7	14(40.0)	13(37.1)	6(17.1)	2(5.7)
Item 8	13(37.1)	16(45.7)	4(11.4)	2(5.7)

*Notas.*  $f$  = frecuencia.  $P$  = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo.

Tabla 11 (continuación).

*Aspectos relacionados a utilidad de los recursos*

Ítem	Muy útil $f(P)$	Útil $f(P)$	Poco útil $f(P)$	Sin utilidad $f(P)$
Item 9	15(42.9)	9(25.7)	6(17.1)	3(8.6)
Item 10	15(42.9)	13(37.1)	5(14.3)	2(5.7)
Item 11	12(34.3)	10(28.6)	7(20)	5(14.3)

*Notas.*  $f$  = frecuencia.  $P$  = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo.

### **La motivación para el aprendizaje de la programación**

Los ítems de la sección IV del cuestionario auscultan sobre la motivación hacia el aprendizaje en el curso Introducción a la Programación. Esto en respuesta a la cuarta pregunta de investigación: ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a su nivel de motivación para el aprendizaje de la programación?

En esta sección del cuestionario los participantes debían seleccionar la respuesta que mejor reflejara su opinión utilizando una escala tipo Likert con las categorías de respuesta: Completamente de acuerdo (CDA), De acuerdo (DA), Neutral (N), En desacuerdo (ED) y Completamente en desacuerdo (CED). La cantidad de ítems presentados en esta sección fueron 18, de los cuales seis fueron negativos. Esto con el propósito de atender las sugerencias dadas por Medina-Díaz (2010) para la construcción de cuestionarios quien advierte que se debe: “Redactar las premisas tanto en dirección

positiva como negativa, para evitar el error de respuesta fija o de repetición en la selección de las respuestas” (p. 53).

A continuación, se describen los ítems de la Tabla 12 que presentaron mayor relevancia. Para esto se agruparán las alternativas de respuestas CDA con DA y ED con CED. En ese sentido, se destaca que 31 participantes (88.5%) señalan estar CDA o DA con los ítems 1 y 10. Estos ítems indagan sobre la motivación propia por obtener buenas calificaciones y esforzarse para poder conseguir un buen trabajo. También, se obtuvo un alto porcentaje de respuestas (80%) en las alternativas CDA o DA en los siguientes ítems: ítem 2 (sentirse aceptado por parte de los compañeros de clase), ítem 3 (capacidad para aprender todo lo requerido en programación) e ítem 6 (cumplir con éxito tareas difíciles).

Con un porcentaje cercano a los anteriores se encuentra el ítem 16 (Mis decisiones reflejan lo que realmente quiero) obtuvo un (77.1%) de aceptación por parte de los encuestados. En esta misma tendencia, se encuentra el ítem 18 (Estoy seguro de mis habilidades para resolver problemas de programación) con (74.2%) y los ítems 5 (Estoy a gusto al trabajar en grupo con mis compañeros de curso), ítem 13 (Puedo elegir la manera de desarrollar los trabajos de programación) En ambos ítems 26 participantes (74.2%) respondieron las alternativas CDA o DA.

En contraste, respecto a los ítems con mayor número de respuestas en las categorías ED y CED. Se destaca el ítem 4 en el cual 23 participantes (65.7%) indicaron estar ED o CED en (Esforzarse por las calificaciones para complacer a sus padres). De forma similar, el ítem 7 (Esforzarse por las calificaciones para complacer a los profesores), fue seleccionado en las alternativas CED o ED por 21 participantes (60%).

Por otra parte, con relación a los ítems 8, 9, 11, 12, 15 y 17, con enunciados negativos. Se destaca el ítem 8 (Siento que la relación con mis compañeros de curso es superficial) con 27 respuestas (77.1%) en las categorías ED o CED. Algo semejante ocurre con los ítems 12, 15 y 17 que tratan sobre “sentirse inseguro de las habilidades para programar, sentirse fracasado por los errores cometidos al intentar resolver un problema de programación y sentir que desagrado a mis compañeros de curso” los tres ítems obtuvieron 26 respuestas (74.2%) en las alternativas ED o CED. Por último, los participantes seleccionaron 25 respuestas en las categorías ED o CED (71.43%) para el ítem 9. Este ítem expresa “Tengo serias dudas acerca de si puedo hacer las cosas bien”. Es conveniente resaltar que obtener un alto porcentaje en los ítems negativos en las alternativas ED o CDE se interpreta como favorable con respecto a la motivación para el aprendizaje de programación.

Tabla 12.

*Aspectos relacionados a la motivación para el aprendizaje de la programación*

Ítem	CDA <i>f(P)</i>	DA <i>f(P)</i>	N <i>f(P)</i>	ED <i>f(P)</i>	CED <i>f(P)</i>
Item 1	17(48.6)	14(40)	1(2.9)	2(5.7)	1(2.9)
Item 2	11(31.4)	17(48.6)		6(17.1)	1(2.9)
Item 3	9(25.7)	19(54.3)	2(5.7)	4(11.4)	1(2.9)
Item 4	3(8.6)	8(22.9)	1(2.9)	18(51.4)	5(14.3)
Item 5	8(22.9)	17(48.6)	3(8.6)	5(14.3)	2(5.7)

*Notas.* *f* = frecuencia. *P* = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo. Los ítems que aparecen en negrita tienen enunciado negativo.

Tabla 12 (continuación).

*Aspectos relacionados a la motivación para el aprendizaje de la programación*

Ítem	CDA <i>f(P)</i>	DA <i>f(P)</i>	N <i>f(P)</i>	ED <i>f(P)</i>	CED <i>f(P)</i>
Item 6	12(34.3)	16(45.7)	3(8.6)	4(11.4)	
Item 7	1(2.9)	10(28.6)	3(8.6)	20(57.1)	1(2.9)
<b>Item 8</b>	1(2.9)	6(17.1)	1(2.9)	20(57.1)	7(20)
<b>Item 9</b>	1(2.9)	6(17.1)	2(5.7)	17(48.6)	8(22.9)
Item 10	15(42.9)	16(45.7)		3(8.6)	1(2.9)
<b>Item 11</b>	5(14.3)	6(17.1)	2(5.7)	12(34.3)	10(28.6)
<b>Item 12</b>	2(5.7)	3(8.6)	4(11.4)	16(45.7)	10(28.6)
Item 13	8(22.9)	17(48.6)	3(8.6)	6(17.1)	1(2.9)
Item 14	10(28.6)	13(37.1)	1(2.9)	8(22.9)	2(5.7)
<b>Item 15</b>	3(8.6)	4(11.4)	2(5.7)	14(40)	12(34.3)
Item 16	12(34.3)	15(42.9)	3(8.6)	3(8.6)	2(5.7)
<b>Item 17</b>	3(8.6)	3(8.6)	3(8.6)	17(48.6)	9(25.7)
Item 18	7(20)	19(54.3)	2(5.7)	6(17.1)	1(2.9)

*Notas.* *f* = frecuencia. *P* = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo. Los ítems que aparecen en negrita tienen enunciado negativo.

### **Autoeficacia para el aprendizaje de la programación**

Las premisas de la sección V del cuestionario buscan conocer la percepción de los estudiantes con respecto a la autoeficacia para el aprendizaje de la programación. Esto en respuesta a la quinta pregunta de investigación: ¿Cuál es la percepción que tienen los

estudiantes subgraduados del programa de Ciencia de Cómputos con relación a su nivel de autoeficacia para el aprendizaje de la programación?

En esta sección del cuestionario, al igual que la anterior, los participantes debían seleccionar la respuesta que mejor reflejara su opinión utilizando una escala tipo Likert con las categorías de respuesta: Completamente de acuerdo (CDA), De acuerdo (DA), Neutral (N), En desacuerdo (ED) y Completamente en desacuerdo (CED).

En la Tabla 13 se informan la frecuencia y porcentaje de las respuestas de los participantes a cada uno de los ítems de esta sección. A continuación, se indican los ítems en los cuales la mayoría de los participantes seleccionaron las categorías CDA o DA. En este sentido, se destaca el ítem 9: “Académicamente me siento una persona competente”, con un 82.8% (29 participantes) en estas categorías. En esta misma tendencia, en los ítems: 1, 2, 3, 4, 6, 15 y 13, 28 participantes (80%) señalaron las categorías CDA o DA.

De forma general, se puede indicar que el porcentaje de respuestas de los participantes en las categorías CDA o DA en esta subescala del cuestionario osciló entre un 60% a un 82.8%. Por lo tanto, se puede indicar que la mayoría de los participantes encontró gran aceptación hacia las premisas que abordaron el tema de autoeficacia.

Tabla 13.

*Aspectos relacionados a la autoeficacia para el aprendizaje de la programación*

Ítem	CDA <i>f(P)</i>	DA <i>f(P)</i>	N <i>f(P)</i>	ED <i>f(P)</i>	CED <i>f(P)</i>
Item 1	11(31.4)	17(48.6)	2(5.7)	4(11.4)	1(2.9)
Item 2	10(28.6)	18(51.4)	4(11.4)	3(8.6)	

*Notas.* *f* = frecuencia. *P* = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo.

Tabla 13 (continuación).

*Aspectos relacionados a la autoeficacia para el aprendizaje de la programación*

Ítem	CDA <i>f(P)</i>	DA <i>f(P)</i>	N <i>f(P)</i>	ED <i>f(P)</i>	CED <i>f(P)</i>
Item 3	12(34.3)	16(45.7)	2(5.7)	4(11.4)	
Item 4	7(20)	21(60)	3(8.6)	3(8.6)	1(2.9)
Item 5	9(25.7)	17(48.6)	2(5.7)	5(14.3)	2(5.7)
Item 6	5(14.3)	23(65.7)	1(2.9)	6(17.1)	
Item 7	11(31.4)	13(37.1)	3(8.6)	8(22.9)	
Item 8	7(20)	16(45.7)	3(8.6)	6(17.1)	3(8.6)
Item 9	14(40)	15(42.9)	1(2.9)	5(14.3)	
Item 10	7(20)	18(51.4)	1(2.9)	9(25.7)	
Item 11	7(20)	19(54.3)	1(2.9)	5(14.3)	2(5.7)
Item 12	13(37.1)	15(42.9)	2(5.7)	3(8.6)	1(2.9)
Item 13	15(42.9)	13(37.1)	1(2.9)	6(17.1)	
Item 14	8(22.9)	19(54.3)		8(22.9)	
Item 15	5(14.3)	16(45.7)	7(20)	4(11.4)	3(8.6)

*Notas.*  $f$  = frecuencia.  $P$  = porcentaje. Las instancias en que los porcentajes no totalizan 100% se deben al error de redondeo.

### **Correlación entre la motivación y el rendimiento académico**

Con el propósito de atender la sexta pregunta de investigación: ¿Cuál es la correlación entre la motivación de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de

computadoras? Se presentan a continuación, los hallazgos obtenidos con el análisis de correlación de Spearman entre las respuestas de los participantes en la subescala de motivación y el rendimiento académico (calificación obtenida por el participante en el curso de introducción a la programación). Es importante señalar que el rendimiento académico fue informado por los participantes en la siguiente escala valorativa: A, B, C, D y F.

En la Tabla 14 se presentan los resultados obtenidos al realizar el análisis de correlación de Spearman ( $r_s$ ) entre los ítems de la subescala IV (motivación) y el rendimiento académico. Se destacan aquellas correlaciones en las cuales “el valor de  $p$  es menor que 0.05, se puede concluir que la correlación es significativa, lo que indica una relación real, no debida al azar” (Mondragón, 2014, p. 101). En este sentido, en la Tabla 12, se clasificaron los resultados en aquellos coeficientes de correlación de Spearman con nivel significancia  $<0.01$  (señalados con \*\*) y los coeficientes con nivel de significancia  $<0.05$  (señalados con \*).

Se hallaron correlaciones positivas significativas  $<0.01$  entre los ítems: 2, 3, 8, 9, 11, 13, 16, 18 y el rendimiento académico de los participantes. De igual forma, se encontraron correlaciones positivas significativas  $<0.05$  entre los ítems: 6, 7, 14 y el rendimiento académico. Lo que podría indicar que los tipos de motivación abordados en estos ítems incidieron en buenos resultados académicos de los participantes en el curso introducción a la programación de computadoras.

Tabla 14.

*Análisis de correlación de Spearman ítems subescala IV (Motivación) y el rendimiento académico en el curso introducción a la programación de computadoras.*

	Rendimiento académico
Item 1	$r_s = .203$ $p = .242$
Item 2	<b><math>r_s = .647^{**}</math></b> <b><math>p = .000</math></b>
Item 3	<b><math>r_s = .519^{**}</math></b> <b><math>p = .001</math></b>
Item 4	$r_s = .211$ $p = .223$
Item 5	$r_s = .316$ $p = .064$
Item 6	<b><math>r_s = .408^*</math></b> <b><math>p = .015</math></b>
Item 7	<b><math>r_s = .362^*</math></b> $p = .032$
<b>Item 8</b>	<b><math>r_s = .648^{**}</math></b> <b><math>p = .000</math></b>
<b>Item 9</b>	<b><math>r_s = .664^{**}</math></b> <b><math>p = .000</math></b>
Item 10	$r_s = .140$ $p = .421$
<b>Item 11</b>	<b><math>r_s = .621^{**}</math></b> <b><math>p = .000</math></b>
<b>Item 12</b>	$r_s = .321$ $p = .060$

*Notas.* \*\* Coeficientes de correlación significativos (<0.01); \* coeficientes de correlación significativos (<0.05).

Tabla 14 (continuación).

*Análisis de correlación de Spearman ítems subescala IV (Motivación) y el rendimiento académico en el curso introducción a la programación de computadoras.*

	Rendimiento académico
Item 13	$r_s = .462^{**}$ $p = .005$
<b>Item 14</b>	$r_s = .392^*$ $p = .020$
<b>Item 15</b>	$r_s = .088$ $p = .613$
Item 16	$r_s = .470^{**}$ $p = .004$
<b>Item 17</b>	$r_s = .247$ $p = .152$
Item 18	$r_s = .637^{**}$ $p = .000$

### **Correlación entre la autoeficacia y el rendimiento académico**

En esta sección se presentan las correlaciones de Spearman obtenidas entre los ítems de la subescala V (Autoeficacia) y el rendimiento académico (informado por los participantes) en el curso introducción a la programación de computadoras. Esto en respuesta a la séptima pregunta de investigación: ¿Cuál es la correlación entre la autoeficacia de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras?

La Tabla 15 presenta estos resultados, distinguiendo con coeficientes de correlación significativos ( $<0.01$ ) los ítems: 1, 2, 4, 5, 6, 8, 10, 12 y 15. De igual manera, con coeficientes de correlación significativos ( $<0.05$ ) los ítems: 3, 7, 9 y 13. Por lo

anterior, se puede afirmar que, de forma general, la mayoría de los ítems de esta subescala presentan una correlación positiva significativa con el rendimiento académico. De ahí que se señale el ítem 11 (Tengo la convicción de poder resolver muy bien los exámenes relacionados a programación) y al ítem 14 (Puedo aportar buenas ideas en clase cuando se requiere solucionar un problema de programación) como los únicos ítems que no presentaron correlación significativa con el rendimiento académico.

Por lo tanto, los resultados obtenidos en los análisis de correlación podrían indicar que la autoeficacia incide en los buenos resultados académicos de los participantes en el curso introducción a la programación de computadoras.

Tabla 15.

*Análisis de correlación de Spearman ítems subescala V (Autoeficacia) y el rendimiento académico en el curso introducción a la programación de computadoras.*

	Rendimiento académico
<b>Item 1</b>	<b><math>r_s = .485^{**}</math></b> <b><math>p = .003</math></b>
<b>Item 2</b>	<b><math>r_s = .605^{**}</math></b> <b><math>p = .000</math></b>
<b>Item 3</b>	<b><math>r_s = .417^*</math></b> <b><math>p = .013</math></b>
<b>Item 4</b>	<b><math>r_s = .675^{**}</math></b> <b><math>p = .000</math></b>
<b>Item 5</b>	<b><math>r_s = .616^{**}</math></b> <b><math>p = .000</math></b>

*Notas.* **\*\*** Coeficientes de correlación significativos (<0.01); **\*** coeficientes de correlación significativos (<0.05).

Tabla 15 (continuación).

*Análisis de correlación de Spearman ítems subescala V (Autoeficacia) y el rendimiento académico en el curso introducción a la programación de computadoras.*

	Rendimiento académico
<b>Item 6</b>	<b><math>r_s = .731^{**}</math></b> <b><math>p = .000</math></b>
<b>Item 7</b>	<b><math>r_s = .662^*</math></b> <b><math>p = .000</math></b>
<b>Item 8</b>	<b><math>r_s = .585^{**}</math></b> <b><math>p = .000</math></b>
<b>Item 9</b>	<b><math>r_s = .351^*</math></b> <b><math>p = .039</math></b>
<b>Item 10</b>	<b><math>r_s = .543^{**}</math></b> <b><math>p = .001</math></b>
Item 11	$r_s = .207$ $p = .233$
<b>Item 12</b>	<b><math>r_s = .471^{**}</math></b> <b><math>p = .004</math></b>
<b>Item 13</b>	<b><math>r_s = .425^*</math></b> <b><math>p = .011</math></b>
Item 14	$r_s = .250$ $p = .147$
<b>Item 15</b>	<b><math>r_s = .594^{**}</math></b> <b><math>p = .000</math></b>

*Notas.* **\*\*** Coeficientes de correlación significativos (<0.01); **\*** coeficientes de correlación significativos (<0.05).

## **CAPÍTULO V**

### **ANÁLISIS Y CONCLUSIONES**

El presente estudio se realizó con el propósito de investigar, los factores determinantes que inciden en el aprendizaje de la programación de computadoras. Respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras. Además, se investigó sobre factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos. En este capítulo se presenta la discusión sobre los hallazgos, conclusiones, limitaciones del estudio y sus implicaciones, y sugerencias para futuros estudios.

#### **Discusión de los hallazgos**

La discusión de los resultados se realizará de acuerdo con las preguntas de investigación realizadas en este estudio. Con respecto a la primera pregunta de investigación ¿Cuál es la percepción de los estudiantes subgraduados del programa de Ciencia de Cómputos respecto al nivel de dificultad de los contenidos del curso introducción a la programación de computadoras? De acuerdo con las respuestas de los participantes, los temas de mayor nivel de dificultad, en el aprendizaje inicial de la programación de computadoras son: representar soluciones mediante abstracciones, dividir problemas extensos de programación en subproblemas, y búsqueda de errores en un programa. Estos temas tienen en común que requieren altos niveles de abstracción y la resolución de problemas. De allí que algunos autores consideren que programar computadoras es una tarea difícil, debido a la complejidad del lenguaje y lo abstracto que son los conceptos de programación (Insuasti, 2016). De igual manera, la resolución de problemas mediante programación “precisa una serie de conocimientos y habilidades en

campos como el modelado (lógica matemática y procedimental), la ingeniería con sus arquitecturas y procesos de software y la computación con sus algoritmos, herramientas, técnicas y metodologías de programación” (Villalobos & Casallas, 2006, citados en Jiménez-Toledo, Collazos & Revelo-Sánchez, 2019, p. 86). Esto implica que el aprendiz de programación cuando se enfrenta a contenidos complejos que quizás no tiene símil con sus conocimientos previos, los percibe con un mayor nivel de dificultad.

Por lo anterior, desde el modelo de la autoeficacia de Bandura (1997), se debe promover la exposición a experiencias directas de aprendizaje de la programación desde los primeros años de escolaridad (Soykan & Kanbul, 2018). Así el estudiante, de forma natural puede ir desarrollando habilidades que le permitan ir desde un menor a un mayor nivel de dificultad y le ayuden a sentirse más competente para la realización de una tarea.

En contraste, los temas: reconocer la estructura de un programa, definición de variables y uso de condicional *if*, fueron señalados por la mayoría de los participantes como contenidos con menor nivel de dificultad. Posiblemente esto se deba a que estos contenidos no eran completamente nuevos para los participantes del estudio. Debido a que estos temas son abordados en cursos de matemáticas.

Respecto a la segunda pregunta de investigación: ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los ambientes de aprendizaje más efectivos para la programación? Los participantes de este estudio mostraron una tendencia alta de aceptación a los distintos ambientes para el aprendizaje de programación planteados por esta subescala: Aprendizaje de programación en clases prácticas (laboratorios), clases teóricas, cursos virtuales de

programación, estudio en grupo, mentorías personalizadas, formando parte de grupos relacionados a programación en redes sociales (Por ejemplo: Facebook, Twitter, WhatsApp). Se puede observar que la mayoría de estos ambientes de aprendizaje, implican relacionarse con otras personas. Esto constituye una de las tres necesidades básicas mencionadas por Ryan & Deci (2000), en su teoría de la autodeterminación para desarrollar la motivación intrínseca. De igual forma, en el aprendizaje de los fundamentos de programación los ambientes colaborativos permiten al estudiante “conocer su propio ritmo de aprendizaje y con ello, compararlo con el resto. Este conocimiento le permite aplicar las estrategias metacognitivas para mejorar su aprendizaje, incrementando su motivación” (Carrio, 2007, p. 4).

La pregunta de investigación: ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a los recursos educativos más efectivos para el aprendizaje de la programación? Los resultados obtenidos señalaron que la mayoría de los participantes percibieron los recursos educativos mencionados en la subescala como Muy útiles o útiles. Siendo estos recursos: Ver videos de programación (ejemplo: *Youtube*), Participar en foros en línea de programación de computadoras, leer blogs con temas de programación de computadoras, usar aplicaciones gratuitas en línea para probar códigos de programación, estudiar ejemplos de programas desarrollados, revisar tutoriales en línea sobre programación, usar organizadores gráficos interactivos (diagramas de flujo), revisar mis notas (apuntes) de clase, leer el material de estudio entregado por el profesor, consultar libros de fundamentos de programación y buscar información en la biblioteca.

Sin embargo, revisar tutoriales en línea sobre programación, fue el recurso educativo más efectivo para el aprendizaje de la programación, percibido por los participantes. Esto coincide con una de las fuentes principales que constituyen las creencias de autoeficacia, el aprendizaje vicario (Bandura, 1997). Este tipo de aprendizaje se da por observación directa e imitación.

Por otra parte, la pregunta de investigación: ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos con relación a su nivel de motivación para el aprendizaje de la programación? los participantes mostraron gran aceptación por las premisas que conformaron esta subescala. Exceptuando los ítems: Esforzarse por las calificaciones para complacer a sus padres y esforzarse por las calificaciones para complacer a los profesores. En estos ítem la mayoría de los participantes indicaron estar en ED o CED. Se puede observar que estos ítems hacen alusión a aspectos relacionados con motivación extrínseca. Lo que puede indicar que los participantes se identifican con las premisas asociadas con la motivación intrínseca. De acuerdo con Ryan & Deci (2000) la motivación intrínseca que se refiere al hacer una actividad por la satisfacción inherente que ocasiona la actividad por sí misma.

Respecto a la pregunta de investigación: ¿Cuál es la percepción que tienen los estudiantes subgraduados del programa de Ciencia de Cómputos de su nivel de autoeficacia para el aprendizaje de la programación de computadoras? se puede indicar que la mayoría de los participantes encontró gran aceptación hacia las premisas que abordaron el tema de autoeficacia. Destacándose el ítem 9: “Académicamente me siento una persona competente”. Estos resultados sugieren que los participantes muestran un alto nivel de autoeficacia percibida.

Es importante destacar que de acuerdo con Bandura (1994) un fuerte sentido de eficacia mejora los logros humanos y el bienestar personal de muchas maneras. Las personas con gran seguridad en sus capacidades abordan las tareas difíciles como desafíos que deben dominarse en lugar de amenazas que deben evitarse. Por lo tanto, es favorable que los participantes muestren un alto nivel de autoeficacia percibida, porque esto les permite enfrentar retos complejos con la certeza y creencia de poder superarlos.

Por otra parte, respecto a la pregunta de investigación: ¿Cuál es la correlación entre la motivación de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras? Se hallaron correlaciones positivas significativas entre los ítems de la subescala de motivación: 2, 3, 6, 7, 8, 9, 11, 13, 14, 16, 18 y el rendimiento académico, informado por los participantes, en el curso introducción a la programación de computadoras. Estos ítems se asocian a las tres necesidades básicas que expone la TDA (Autonomía, Relación con los demás y Competencia) como factores determinantes para el desarrollo de la motivación intrínseca. Distribuidos de la siguiente forma: Autonomía (7, 13, 16), Relación con los demás (2, 8, 11, 14), Competencia (3, 6, 9, 18).

Por lo tanto, aunque no todos los ítems que conformaban la subescala de motivación presentaron una correlación significativa positiva con el rendimiento académico en el curso introducción a la programación de computadoras, sí pudo constatar que los ítems que si correlacionaron correspondían a cada una de las tres necesidades básicas sugeridas en la teoría de la autodeterminación de Ryan & Deci (2000). La atención de estas necesidades se constituye en un factor determinante para que un estudiante permanezca en sus estudios (Ryan & Deci, 2000).

Estos resultados son cónsonos con lo expuesto por Park, Kim, Robertson & Kim (2019) quienes indican que el nivel de motivación del estudiante está positivamente relacionado con su proceso de aprendizaje de programación de computadora y los resultados obtenidos por Baser (2013). De igual manera, se cree que la motivación es un facilitador para el aprendizaje y el éxito académico (Lin-nenbrink & Pintrich, 2002; Lynch, 2006, citados por Law, Lee & Yu, 2010).

Por otra parte, respecto a la pregunta de investigación: ¿Cuál es la correlación entre la autoeficacia de los estudiantes para el aprendizaje de la programación y su rendimiento académico en el curso introducción a la programación de computadoras? Se pudo observar que, de forma general, la mayoría de los ítems de esta subescala presentan una correlación positiva significativa con el rendimiento académico. Por lo tanto, los resultados obtenidos en los análisis de correlación podrían indicar que la autoeficacia incide en los buenos resultados académicos de los participantes en el curso introducción a la programación de computadoras. Estos resultados son consistentes con estudios previos como el realizado por Ramalingam, LaBelle, & Wiedenbeck (2004) y Mercado, Andrade & Reynoso, (2018). En estos estudios se comprueba la relación entre los niveles de autoeficacia percibida y el rendimiento académico. Por lo tanto, se recomienda incrementar la autoeficacia “buscando que la teoría y la práctica en el salón de clases no sean demasiado avanzadas para la percepción de capacidad del estudiante, el cual debe ser retado, quien es probable que se rinda si se siente abrumado” (Mercado, Andrade & Reynoso, 2018, p. 79).

## Conclusiones

“La enseñanza de los Fundamentos de Programación es una parte esencial de un curriculum en Ciencias de la Computación” (Muñoz, Barría, Noël, Providel & Quiroz, 2012 p. 120). En los últimos años el aprendizaje de programación de computadoras ha tomado gran importancia a nivel mundial. Esto ha traído consigo el surgimiento de diferentes iniciativas educativas que promueven el aprendizaje de programación de computadoras, incluso desde la educación elemental. Ejemplo de ello son las “200 millones de personas de todo el mundo que han utilizado los recursos y participado en las actividades impulsadas por iniciativas como La Hora del Código o *YouthSpark* que promueven la enseñanza de la computación en niños y jóvenes (Jara & Hepp, 2016). De igual forma, la fundación *Coder Dojo* que promueve un movimiento global de clubes de programación dirigidos por voluntarios que enseñan programación a los más jóvenes de forma gratuita. Desde Italia se distinguen dos iniciativas: *Code’s Cool* y *CodeWeek.it*<sup>6</sup>. Ambas promovidas por la Universidad de Urbino y la asociación NeuNet. En esta misma línea, muchos países como Estonia, Francia o Reino Unido han incorporado a sus currículums la enseñanza de la programación de computadoras desde las edades más tempranas (Arranz, 2017).

En el aprendizaje inicial de la programación de computadoras, los estudiantes se enfrentan a contenidos nuevos que pueden resultar desafiantes. Que requieren el desarrollo de habilidades superiores de pensamiento. En este sentido, los participantes informaron que en el aprendizaje inicial de la programación de computadoras los temas de mayor nivel de dificultad son aquellos que requieren altos niveles de abstracción y la resolución de problemas. Sin embargo, “se ha evidenciado que un buen desempeño

académico no puede ser garantizado solo por los conocimientos y habilidad de los individuos. Las creencias de eficacia pueden determinar un desempeño diferente en dos personas con el mismo grado de habilidad” (Ruiz, 2005, p. 10).

Por otra parte, en cuanto a los ambientes de aprendizaje más efectivos para el aprendizaje de la programación de computadoras, los estudiantes destacan el aprendizaje colaborativo mediante comunidades de aprendizaje. De allí que, “los espacios construidos en la interacción con profesores y compañeros de estudio serían en los que tendrían lugar los procesos de construcción de significados y de atribución de sentido a los contenidos” (Cenich, 2014, p. 5).

De forma similar, revisar tutoriales en línea sobre programación, fue el recurso educativo más efectivo para el aprendizaje de la programación, percibido por los participantes en el presente estudio. Esto coincide con una de las fuentes principales que constituyen las creencias de autoeficacia según Bandura (1997).

Otro aspecto por resaltar es que la autoeficacia y la motivación de los aprendices de programación se constituyen en factores que pueden incidir favorablemente en el rendimiento académico de los estudiantes. En el presente estudio se pudo encontrar una correlación de Spearman positiva significativa entre los ítems de las subescalas de autoeficacia y motivación con el rendimiento académico informado por los participantes en el curso de fundamentos de la programación.

Por lo tanto, la presente investigación busca servir de base para gestar iniciativas en el contexto de Puerto Rico para la enseñanza de la programación de computadoras desde la edad escolar. Esto debido a que: “para la lectura, escritura y aritmética, hay que añadir pensamiento computacional a la capacidad analítica de cada niño” (Wing, 2006, p.

33). En este sentido, aprender a programar contribuye al desarrollo de habilidades de pensamiento como la resolución de problemas y desarrollo de creatividad. Ambas se constituyen en habilidades de pensamiento de orden superior. Un meta-análisis de 440 estudios de estudiantes de los grados K-12 realizado por Scherer, Siddiq & Viveros (2019) confirma que el aprendizaje de la programación de computadoras mejora la creatividad, las habilidades matemáticas, la metacognición, las habilidades espaciales y las habilidades de razonamiento de los estudiantes.

Por tanto, es indispensable incluir dentro del currículo en la escuela una asignatura de programación de computadoras. Para esto, se deben adoptar metodologías de aprendizaje adecuadas con la edad e intereses de los estudiantes. De igual manera, como se pudo evidenciar en la presente investigación la motivación y la autoeficacia del estudiante son fundamentales en el aprendizaje inicial de la programación de computadoras. De allí que existan proyectos exitosos en el que se incluye la lúdica como eje principal en el aprendizaje de programación. “En esta línea se encuentran resultados de proyectos muy interesantes como Alice, Greenfoot y Scratch” (Rodríguez, 2014, p. 60) y organizaciones como code.org. que promueven el aprendizaje de la programación de computadoras de forma sencilla y divertida desde la edad escolar.

Enseñemos a nuestros niños a programar para que las máquinas del futuro hagan lo que ellos quieran y no dejemos que ocurra justo lo contrario. Programar o ser programado (Rushkoff, 2010).

### **Limitaciones del estudio y sus implicaciones**

1. En la revisión de literatura realizada en esta investigación no se encontró estudios en los que se explorara la percepción de los estudiantes de Ciencia de Cómputos en Puerto

Rico respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras y factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos.

2. Inicialmente en este estudio se consideró la recolección de datos de forma presencial mediante un cuestionario impreso. Por motivos de la pandemia por COVID-19 las clases en la Universidad de Puerto Rico pasaron a la modalidad virtual. Por lo tanto, la forma de administración del cuestionario se ajustó a esta modalidad. Esta situación pudo ser un factor para la baja tasa 23% de respuesta al cuestionario.

3. Debido a que en este estudio no se seleccionó la muestra de forma aleatoria, esta no es representativa de la población total. La selección de las participantes de este estudio será por conveniencia y consistirá en todas los estudiantes del programa subgraduado de Ciencia de Cómputos de la UPRRP, que hayan tomado el curso de Introducción a la Programación de Computadoras que deseen voluntariamente participar. Por lo tanto, los resultados que se obtenga de esta investigación no podrán ser generalizados a todas los estudiantes del programa de Ciencia de Cómputos.

### **Sugerencias para futuros estudios**

Los resultados obtenidos en la presente investigación sirven de insumo para continuar futuros estudios en la exploración de factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras y factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos. Para esto se sugiere ampliar la población objeto de estudio a todos los estudiantes de Ciencia de Cómputos, de todas las universidades de Puerto

Rico, que hayan tomado el curso de Introducción a la programación. De igual manera que el tipo de muestreo sea aleatorio para que los resultados pueden ser generalizados a la población.

Se sugiere ampliar este estudio, mediante la implementación de un método mixto. De esta forma, con la fase cualitativa mediante entrevista a profundidad se podrán auscultar los factores que los estudiantes consideran que influyen en el aprendizaje inicial de la programación de computadoras.

También se sugiere indagar la percepción de los(as) profesores(as) del curso Introducción a la programación de computadoras respecto a los factores internos y externos determinantes en el aprendizaje de la programación de computadoras.

## REFERENCIAS

- ACM, IEEE & AIS (2005). Computing Curricula: The Overview Report 2005. Retrieved from: [http://www.acm.org/education/education/curric\\_vols/CC2005-March06Final.pdf](http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf)
- AERA, APA & NCME. (2014). *Standards for educational and psychological testing*. Washington, DC: AERA.
- AERA, APA. (1999). *Standards for educational and psychological testing*. New York: AERA.
- Ala-Mutka, K. (2003). Problems in Learning and Teaching Programming – a literature study for developing visualizations in the Codewitz-Minerva project. Codewitz Needs Analysis. Available at: <https://moam.info/queue/1-problems-in-learning-and-teaching-programming-a-597aa0f21723dd91e8e52e27.html>
- Álvarez, M. (2018). *Significado de la experiencia de educadores con la integración del pensamiento computacional: un estudio fenomenológico* (Disertación doctoral). Universidad de Puerto Rico, Recinto de Río Piedras.
- Anderson, N., & Gegg-Harrison, T. (2013, March). Learning computer science in the comfort zone of proximal development. In *Proceeding of the 44th ACM technical symposium on Computer science education*, ACM, 495-500.
- Arenas París, R. J. (2014). *Modelo para la Motivación del Aprendizaje de la Programación utilizando Gamification* (Tesis). Pontificia Universidad Javeriana, Bogotá-Colombia.
- Arranz, H. (2017). *La enseñanza de la programación informática en Educación Primaria: situación actual, análisis y presentación de experiencias de aula en España* [Trabajo de grado, Universitat de les Illes Balears] [https://dspace.uib.es/xmlui/bitstream/handle/11201/3892/Arranz de la Fuente Hector.pdf?sequence=1&isAllowed=y](https://dspace.uib.es/xmlui/bitstream/handle/11201/3892/Arranz_de_la_Fuente_Hector.pdf?sequence=1&isAllowed=y)
- Askar, P. & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *Online Submission, TOJET: The Turkish Online Journal of Educational Technology*, 8(1). <http://eric.ed.gov/?id=ED503900>
- Association for Computing Machinery, IEEE Computer Society (2004). *Software Engineering 2004: Curriculum guidelines for undergraduate degree programs in Software Engineering*. ACM.

- Astudillo, G. J., Bast, S. G., & Willging, P. A. (2016). Enfoque basado en gamificación para el aprendizaje de un lenguaje de programación. *Virtualidad, Educación y Ciencia*, 7(12), 125-142.
- Bandura, A. (1994). Self-efficacy. In V. S. Ramachaudran (Ed.), *Encyclopedia of human behavior* (Vol. 4, pp. 71-81). New York: Academic Press. (Reprinted in H. Friedman [Ed.], *Encyclopedia of mental health*. San Diego: Academic Press, 1998).
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. Nueva York: Freeman.
- Baser, M. (2013). Attitude, gender and achievement in computer programming. *Online Submission*, 14(2), 248-255.
- Beltrán, J., Sánchez, H., & Rico, M. (2015). Análisis cuantitativo y cualitativo del aprendizaje de Programación I en la Universidad Central del Ecuador. *Revista Tecnológica ESPOL – RTE*, 28 (5), 194-210.
- Bernal, C. A. (2011). *Metodología de la investigación* (3rd ed.). Naucalpan, México: Pearson Educación de México, S. A. de C. V.
- Blanco, Á. B. (2010). Creencias de autoeficacia de estudiantes universitarios: un estudio empírico sobre la especificidad del constructo. *RELIEVE-Revista Electrónica de Investigación y Evaluación Educativa*, 16(1).  
<https://doi.org/10.7203/relieve.16.1.4149>
- Bonetto, V. A., Paoloni, P. V., & Donolo, D. S. (2017). Creencias de autoeficacia y contextos de evaluación. Un estudio con alumnos universitarios. *Actualidades Investigativas en Educación*, 17(2). [10.15517/aie.v17i1.28144](https://doi.org/10.15517/aie.v17i1.28144)
- Bosch N., D’Mello S., Mills C. (2013) *What Emotions Do Novices Experience during Their First Computer Programming Learning Session?*. In: Lane H.C., Yacef K., Mostow J., Pavlik P. (eds) *Artificial Intelligence in Education. AIED 2013. Lecture Notes in Computer Science*, vol 7926. Springer, Berlin, Heidelberg  
[https://doi.org/10.1007/978-3-642-39112-5\\_2](https://doi.org/10.1007/978-3-642-39112-5_2)
- Bruno, O. R. (2005). Análisis de la percepción de los alumnos y de los docentes para la incorporación de un sistema tutor inteligente como facilitador del aprendizaje de algoritmia. *Revista de Informática Educativa y Medios Audiovisuales*, 2 (4), 1-31.
- Carbone, A., Hurst, J., Mitchell, I., & Gunstone, D. (2009, January). An exploration of internal factors influencing student learning of programming. In *Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95* (pp. 25-34).
- Carrancio, L.M. (2013). *Percepciones de la Generación Z y la escuela en el Siglo XXI*. [Tesis de maestría, Universidad Abierta Interamericana].  
<http://imgbiblio.vaneduc.edu.ar/fulltext/files/TC113932.pdf>

- Carrio, M. (2007). Ventajas del uso de la tecnología en el aprendizaje colaborativo. *Revista Iberoamericana de educación*, 41(4), 1-10.
- Casares, J. P. C. (1999). *Ambiente para la instrucción visual de algoritmos* [Tesis inédita, Instituto Tecnológico Autónomo de México]. <http://usablehack.com/amiva/AMIVA.Pdf>
- Casas, S., & Vanoli, V. (2007). Programación y algoritmos: análisis y evaluación de cursos introductorios. In *IX Workshop de Investigadores en Ciencias de la Computación* (pp. 760–764). Comodoro Rivadavia <http://sedici.unlp.edu.ar/handle/10915/20502>
- Cenich, G. (2014). Una propuesta para la enseñanza de programación en la Escuela Secundaria. *I Encuentro Internacional de Educación: espacios de investigación y divulgación*, 1a ed. Tandil, Universidad Nacional del Centro de la Provincia de Buenos Aires. <https://digital.cic.gba.gob.ar/handle/11746/6238>
- CODE. (2015). Leaders and trend-setters all agree on one thing. Retrieved from: 10 septiembre, 2019, de <https://code.org/quotes>
- Zweben, S., & Bizot, B. (2016). Generation CS Continues to Produce Record Undergrad Enrollment; Graduate Degree Production Rises at both Master's and Doctoral Levels. *Computing Research News*, 29(5). <http://cra.org/crn/wp-content/uploads/sites/7/2017/05/2016-Taulbee-Survey.pdf>
- Creswell, J. W. (2008). *Research design: Qualitative, quantitative and mixed methods approaches* (3rd ed.). Thousand Oaks, CA: SAGE Publications, Inc.
- Creswell, J. W. (2012). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research* (4th ed.). Upper Saddle River, NJ: Merrill Prentice Hall.
- Davidson, K., Larzon, L., & Ljunggren, K. (2010). Self-efficacy in programming among STS students. Retrieved <http://www.it.uu.se/edu/course/homepage/datadidaktik/ht10/reports/Self-Efficacy.pdf>.
- Elbelman, A. (2018). Collaborative Learning: Why coding is the best field for kids to learn from each other. *TekkieUni - Coding for Kids*. <https://tekkieuni.com/blog/collaborative-code-learning/>
- Elboj, C. E., Puigdellivol, I. P., Soler Gallart, M. S., & Valls Carol, M. R. (2003). *Comunidades de aprendizaje: transformar la educación*. España: Graó.
- Enríquez, C. (2017, 21 diciembre). La importancia de Aprender a Programar. *Milenio*. Retrieved from <https://www.milenio.com/opinion/varios-autores/universidad-politecnica-de-tulancingo/la-importancia-de-aprender-a-programar>

- Ferreira, A., & Rojo, G. A. (2006). Enseñanza de la programación. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, 1 (1), 1-8.
- Figueiredo, F., & García-Peñalvo, F. J. (2018). Estrategias de enseñanza y aprendizaje de la programación en cursos universitarios. Retrieved from <http://repositorio.grial.eu/handle/grial/1285>
- Fowler, F., Jr. (2009). *Survey research methods* (4a ed.). Thousand Oaks, CA: SAGE Publications, Inc.
- Fracchia, C. C., Alonso de Armiño, A. C., & Martins, A. (2014). Enseñanza de la programación: un tema en la agenda académica para repensar año a año. In *XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014)*.
- Fuentes-Rosado, J.I. & Moo-Medina, M. (2017). Dificultades de aprender a programar. *Revista Educación en Ingeniería*, 12(24), 76-82.
- García-Chato, G. I. (2014). Ambiente de aprendizaje: Su significado en educación preescolar. *Revista de Educación y Desarrollo*, 29, 63-72.
- Gliem, J., & A. Gliem, R. (october, 2003). *Calculating, interpreting and reporting Cronbach's Alpha reliability coefficient for Likert-Type scales*. Presented at the Midwest Research to Practice Conference in Adult, Continuing, and Community Education, Columbus, OH. Retrieved from: <https://scholarworks.iupui.edu/bitstream/handle/1805/344/Gliem%20%26%20Gliem.pdf?sequence=1&isAllowed=y>
- Govender, I., Govender, D. W., Havemga, M., Mentz, E., Breed, B., Dignum, F., & Dignum, V. (2014). Increasing self-efficacy in learning to program: exploring the benefits of explicit instruction for problem solving. *TD: The Journal for Transdisciplinary Research in Southern Africa*, 10(1), 187-200.
- Gries, G. (2002). Where is programming methodology these days?," *ACM SIGCSE Bull.*, 34(4), 1-7.
- Groves, R. M., Fowler, F. J., Jr., Couper, M. P., Lepkowski, J. M., Singer, E., & Tourangeau, R. (2009). *Survey methodology* (2nd ed.). Hoboken, NJ: John Wiley & Sons, Inc.
- Gulatee, Y., & Combes, B. (2006, november). *Identifying the Challenges in Teaching Computer Science Topics Online*. In International Conference. Engagement and Empowerment: New Opportunities for Growth in Higher Education, Edith Cowan University, Perth Western Australia. Retrieved from: <https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1076&context=ceducom>

- Guo, P. J. (2017, May). Older adults learning computer programming: motivations, frustrations, and design opportunities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 7070-7083). ACM.
- Hernández, B. & Velasco-Mondragón, H. E. (2000). Encuestas transversales. *Salud pública de México*, 42(5), 447-455.
- Hernández, G. A. (2013). Creencias docentes y didáctica de la programación de computadoras, *Docencia, investigación e Innovación*, 2(2), 87 -103.
- Herrera, F., Ramírez, M. I., Roa, J. M., y Herrera, I. (2004). Tratamiento de las creencias motivacionales en contextos educativos pluriculturales. *Revista Iberoamericana de Educación, Sección de Investigación*, 34(1), 1-21.  
<https://dialnet.unirioja.es/servlet/articulo?codigo=6429143>
- Insuasti, J. (2016) Problemas de enseñanza y aprendizaje de los fundamentos de programación. *Revista educación y desarrollo social*, 10 (2), 234-246. DOI: org/10/18359/reds.1701
- International Society for Technology in Education (ISTE). (2008). *National educational technology standards for teachers*. Washington, DC: International Society for Technology in Education. Retrieved from: <http://www.iste.org>
- Jara, I., & Hepp, P. (2016). Enseñar Ciencias de la Computación: creando oportunidades para los jóvenes de América Latina. Retrieved from <http://repositorio.minedu.gob.pe/bitstream/handle/MINEDU/5936/Ense%20Ciencias%20de%20la%20Computaci%20creando%20oportunidades%20para%20los%20j%20venes%20de%20Am%20rica%20Latina.pdf?sequence=1&isAllowed=y>
- Jenkins, T. (2001). The motivation of students of programming, *ACM SIGCSE Bull.*, 33(3), 53–56.
- Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, pp. 53–58).  
<http://www.ics.heacademy.ac.uk/Events/conf2002/tjenkins.pdf>
- Jiménez-Toledo, J., Collazos, C., & Revelo-Sánchez, O. (2019). Consideraciones en los procesos de enseñanza-aprendizaje para un primer curso de programación de computadores: una revisión sistemática de la literatura. *TecnoLógicas*, (22), 83-117.
- Kasto, N. (2016). *Learning to Program: The development of knowledge in Novice Programmers* [Doctoral dissertation, Auckland University of Technology].  
<http://hdl.handle.net/10292/10377>

- Kelleher, C., & Pausch, R. (2003). *Lowering the Barriers to Programming: a survey of programming environments and languages for novice programmers*. Human Factors. <https://www.cs.cmu.edu/~caitlin/papers/NoviceProgSurvey.pdf>
- Kori, K., Pedaste, M., Leijen, Ä., & Tõnisson E. (2016). The Role of Programming Experience in ICT Students' Learning Motivation and Academic Achievement, *International Journal of Information and Education Technology*, 6 (5), 331-337.
- Korkmaz, Ö., & Altun, H. (2014). Adapting Computer Programming Self-Efficacy Scale and Engineering Students' Self-Efficacy Perceptions. *Online Submission*, 1(1), 20-31.
- Lacoa, R. F., Lacoa, J., & Blai, A. (2016). La enseñanza de Lenguajes de Programación en la Escuela: ¿ Por qué hay que prestarle atención. *Fundación Telefónica*.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *Acm Sigcse Bulletin*, 37(3), 14-18.
- Law, K. M., Lee, V. C., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, 55(1), 218-228.
- Leyva, J. (1999). Las generaciones de computadoras. *Ciencia y desarrollo*, 25(149), 44-51.
- López, J. (2009). *Educación básica algoritmos y programación*. Colombia. Eduteka.org.
- Lovos, E. (2015). Ambiente de desarrollo virtual para el aprendizaje de la programación: un estudio de caso en la Lic. de Sistemas de la Universidad Nacional de Río Negro, Patagonia Argentina/Virtual Development Environment for Learning Programming: a Case Study at Bachelor Systems of Rio Negro University, Patagonia Argentina. *Revista Internacional de Aprendizaje en Ciencia, Matemáticas y Tecnología*, 2(1).
- Lumsden, L. (1994). Student Motivation to Learn. ERIC Digest: 92. (ERIC Document Reproduction Service No. ED370200)
- Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. *Instructional Science*, 26, 49–63.
- McMillan, J. H. (2004). *Educational research: Fundamentals for the consumer* (4a ed.). Boston, MA: Allyn and Bacon.
- McMillan, J. H., & Schumacher, S. (2005). *Investigación educativa* (5th ed.). Madrid, España: Pearson Educación, S. A.
- Medina-Díaz, M. del R. (2010). *Construcción de cuestionarios para la investigación educativa*. San Juan, PR: ExPERTS Consultants.

- Medina-Díaz, M. del R. (2013). ¿“Validez” o “validación” de un instrumento? *INEVA en acción*, 9(1),1-4. Retrieved from <http://ineva.uprrp.edu/boletin/v0009n0001.pdf>.
- Mercado, C. A. A., Andrade, E. L. M., & Reynoso, J. M. G. (2018). El efecto de la autoeficacia y el trabajo colaborativo en estudiantes novatos de programación. *Investigación y Ciencia: de la Universidad Autónoma de Aguascalientes*, (74), 73-80.
- Mohanarajah, S. (2018). Increasing intrinsic motivation of programming students: Towards fix and play educational games. *Issues in Informing Science and Information Technology*, 15, 69-77. <https://doi.org/10.28945/4027>
- Mondragón, M. (2014). Uso de la correlación de Spearman en un estudio de intervención en fisioterapia. *Movimiento científico*, 8(1), 98-104.
- Morales, L., Chaclán, C., Maldonado, S., Sontay, G., Montenegro, R., & Magzul, J. (2013). *Comunidades de aprendizaje y círculos de lectura*. Guatemala: Juarez & Associates, Inc.
- Moravec, J. (2013). *Knowmad society*. Minneapolis: Education Futures.
- Moroni, N. (2001). Entornos para el Aprendizaje de la Programación. In *I Jornadas de Educación en Informática y TICs en Argentina*. Retrieved from: [https://pdfs.semanticscholar.org/1bab/06fd5b2f6a2a3c42e5a69ae3ab1c86870767.pdf?\\_ga=2.253808008.1663788651.1570843284-1147640258.1552007680](https://pdfs.semanticscholar.org/1bab/06fd5b2f6a2a3c42e5a69ae3ab1c86870767.pdf?_ga=2.253808008.1663788651.1570843284-1147640258.1552007680)
- Muñoz, R.; Barría, M.; Noël, R.; Providel, E. & Quiroz, P. (Agosto de 2012). Determinando las Dificultades en el Aprendizaje de la Primera Asignatura de Programación en Estudiantes de Ingeniería Civil Informática. XVII Congreso Internacional de Informática Educativa, TISE. Santiago, Chile.
- Muñoz, R., Barcelos, T. S., Villarroel, R., Barría, M., Becerra, C., Noel, R., & Frango Silveira, I. (2015). Uso de Scratch y Lego Mindstorms como apoyo a la docencia en Fundamentos de programación. In *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática* (pp. 248-254). Universitat Oberta La Salle.
- Nieto, J. H. (2005). Resolución de problemas, Matemática y Computación. *Enl@ ce: Revista Venezolana de Información, Tecnología y Conocimiento*, 2(2), 37-45.
- Ocre (2016). Todo el mundo debería aprender a programar un ordenador, porque te enseña a pensar. Retrieved from: <http://ocrebcn.com/mundo-deberia-saber-programar/>
- Ortega, Y. (2016). *Motivación en el aprendizaje de la programación a nivel bachillerato utilizando un lenguaje de programación educativo*, [Tesis de maestría, Universidad Autónoma de Aguascalientes México].

- Ossa, G. C. (2002). Tendencias educativas para el siglo XXI: Educación virtual, Online y@ Learning. Elementos para la discusión. *Eduotec: Revista electrónica de tecnología educativa*, (15), 2.
- Pajares, F. (2002). *Self-efficacy beliefs in academic contexts: An outline*.
- Park, H., Kim, K., Robertson, C., & Kim, D. (2019). The Effect of Memorization on the Retention and Learning Acquisition of Programming Practice. *Journal of Strategic Innovation and Sustainability*, 14(2).
- Pérez, R. (2008). Una Herramienta y Técnica para la Enseñanza de la Programación. *Universidad Politécnica del Valle de México*, 11.
- Robins A., Rountree J., & Rountree N. (2003). “Learning and teaching programming: A review and discussion”, *Computer Science Education*, 13(2), 137-172.
- Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004, June). Self-efficacy and mental models in learning to program. In *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 171-175).
- Rodríguez G. M. (2014). Enseñanza de la programación de computadoras para principiantes: Un contexto histórico. *Revista Inventum*, 9(17), 51-61. <http://dx.doi.org.uprrp.idm.oclc.org/10.26620/uniminuto.inventum.9.17.2014.51-61>
- Rushkoff, D. (2010). *Program or Be Programmed: Ten Commandments for a Digital Age*. New York: OR Books. doi:10.2307/j.ctt207g7rj
- Ruiz, F. (2005). Influencia de la autoeficacia en el ámbito académico. *RIDU*, 1(1), 2.
- Ryan, R. & Deci, E. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25, 54-67.
- Sanabria, J. & Romero, M. (2018). Competencias del siglo XXI en proyectos tecnocreativos. *Revista Mexicana de bachillerato a distancia*, (9), 1-8.
- Santimateo, D., Nuñez, G., & González, E. (2018). Estudio de dificultades en la enseñanza y aprendizaje en los cursos básicos de programación de computadoras en Panamá. *Revista de Investigación en Tecnologías de la Información*, 6(11).
- Scherer, R., Siddiq, F., & Viveros, B.S. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, 111, 764-792.
- Soloway, E. (1986). Learning to Program = Learning to Construct Mechanisms and Explanations, *Communications of the ACM*, 29 (9), 850-857.
- Soykan, F., & Kanbul, S. (2018). Analysing K12 Students’ Self-Efficacy Regarding Coding Education. *TEM Journal*, 7(1), 182.

- Strousser, D. (1995). Applications of self-efficacy theory in rehabilitation counseling. *The Journal of Rehabilitation*, 61 (1), 7-12.
- Tavares, P. C., Henriques, P. R., & Gomes, E. F. (2017). A Computer Platform to Increase Motivation in Programming Students-PEP. *In CSEDU (1)*, 284-291
- Villalobos, J. & Casallas, R. (2006). *Fundamentos de programación, aprendizaje activo basado en casos*. Bogotá, Colombia: Pearson Education.
- Villalobos, J., Casallas, R. & Marcos, K. (2005). El reto de diseñar un primer curso de programación de computadores. En *XIII Congreso Iberoamericano de Educación Superior en Computación. Congreso llevado a cabo en Cali, Colombia*.
- Vizcaíno, A., Valencia, D., Soto, J. P., García-Mundo, L., & Piattini, M. (2016). ¿Qué desafíos presenta el desarrollo global del software? Aprende jugando. In *XXI Jornadas de Ingeniería del Software y Bases de Datos*, 605-608.
- Vygotsky L. S. (1978) *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, Massachusetts.

## APÉNDICES

**APÉNDICE A**

**Certificado de *CITI Program***

## COLLABORATIVE INSTITUTIONAL TRAINING INITIATIVE (CITI PROGRAM)

### COMPLETION REPORT - PART 1 OF 2

#### COURSEWORK REQUIREMENTS\*

\* NOTE: Scores on this [Requirements Report](#) reflect quiz completions at the time all requirements for the course were met. See list below for details. See separate [Transcript Report](#) for more recent quiz scores, including those on optional (supplemental) course elements.

- **Name:** Roberto Martínez Mejía (ID: 5883976)
- **Email:** roberto.martinez@upr.edu
- **Institution Affiliation:** Universidad de Puerto Rico, Recinto de Río Piedras (ID: 2377)
- **Institution Unit:** Educacion
- **Phone:** 7874847815
  
- **Curriculum Group:** Investigaciones psicológicas, sociales o educativas
- **Course Learner Group:** Investigaciones psicológicas, sociales o educativas con seres humanos
- **Stage:** Stage 1 - Stage 1
  
- **Report ID:** 21083013
- **Completion Date:** 14-Oct-2016
- **Expiration Date:** 14-Oct-2019
- **Minimum Passing:** 80
- **Reported Score\*:** 95

REQUIRED AND ELECTIVE MODULES ONLY	DATE COMPLETED	SCORE
Historia y principios éticos (ID: 1478)	13-Oct-2016	5/5 (100%)
Investigación con poblaciones protegidas. Sujetos vulnerables: una definición (ID: 1483)	13-Oct-2016	4/4 (100%)
Investigación con datos de archivo (ID: 1490)	13-Oct-2016	2/2 (100%)
Conflictos de interés en investigaciones con sujetos humanos. (ID: 1689)	13-Oct-2016	3/3 (100%)
Daño a grupos. Investigación con poblaciones cultural o médicamente vulnerables (ID: 1719)	14-Oct-2016	3/3 (100%)
Evaluación de riesgo en investigaciones en ciencias sociales y del comportamiento (ID: 1716)	14-Oct-2016	5/5 (100%)
Privacidad y confidencialidad (ID: 1677)	14-Oct-2016	4/5 (80%)
Ética en la publicación de los resultados de la investigación (ID: 1717)	14-Oct-2016	6/6 (100%)
Investigación con mujeres (ID: 1755)	14-Oct-2016	2/3 (67%)
Comités de ética de investigación: autoridad y rol en la revisión de los protocolos de investigación. (ID: 1675)	14-Oct-2016	5/5 (100%)

For this Report to be valid, the learner identified above must have had a valid affiliation with the CITI Program subscribing institution identified above or have been a paid Independent Learner.

Verify at: <https://www.citiprogram.org/verify/?a78a5282-f219-4b47-9642-38529e58e942>

#### CITI Program

Email: [support@citiprogram.org](mailto:support@citiprogram.org)

Phone: 888-529-5929

Web: <https://www.citiprogram.org>

## COLLABORATIVE INSTITUTIONAL TRAINING INITIATIVE (CITI PROGRAM)

### COMPLETION REPORT - PART 2 OF 2 COURSEWORK TRANSCRIPT\*\*

\*\* NOTE: Scores on this Transcript Report reflect the most current quiz completions, including quizzes on optional (supplemental) elements of the course. See list below for details. See separate Requirements Report for the reported scores at the time all requirements for the course were met.

- **Name:** Roberto Martínez Mejía (ID: 5883976)
- **Email:** roberto.martinez6@upr.edu
- **Institution Affiliation:** Universidad de Puerto Rico, Recinto de Río Piedras (ID: 2377)
- **Institution Unit:** Educacion
- **Phone:** 7874847615
- **Curriculum Group:** Investigaciones psicológicas, sociales o educativas
- **Course Learner Group:** Investigaciones psicológicas, sociales o educativas con seres humanos
- **Stage:** Stage 1 - Stage 1
- **Report ID:** 21063913
- **Report Date:** 14-Oct-2016
- **Current Score\*\*:** 95

REQUIRED, ELECTIVE, AND SUPPLEMENTAL MODULES	MOST RECENT	SCORE
Historia y principios éticos (ID: 1478)	13-Oct-2016	5/5 (100%)
Comités de ética de investigación: autoridad y rol en la revisión de los protocolos de investigación. (ID: 1675)	14-Oct-2016	5/5 (100%)
Evaluación de riesgo en investigaciones en ciencias sociales y del comportamiento (ID: 1718)	14-Oct-2016	5/5 (100%)
Privacidad y confidencialidad (ID: 1677)	14-Oct-2016	4/5 (80%)
Investigación con datos de archivo (ID: 1490)	13-Oct-2016	2/2 (100%)
Investigación con poblaciones protegidas. Sujetos vulnerables: una definición (ID: 1483)	13-Oct-2016	4/4 (100%)
Daño a grupos. Investigación con poblaciones cultural o médicamente vulnerables (ID: 1719)	14-Oct-2016	3/3 (100%)
Conflictos de interés en investigaciones con sujetos humanos. (ID: 1689)	13-Oct-2016	3/3 (100%)
Ética en la publicación de los resultados de la investigación (ID: 1717)	14-Oct-2016	6/6 (100%)
Investigación con mujeres (ID: 1755)	14-Oct-2016	2/3 (67%)

For this Report to be valid, the learner identified above must have had a valid affiliation with the CITI Program subscribing institution identified above or have been a paid Independent Learner.

Verify at: <https://www.citiprogram.org/verify/?a76a5282-f219-4b47-9942-38529e58e042>

Collaborative Institutional Training Initiative (CITI Program)

Email: [support@citiprogram.org](mailto:support@citiprogram.org)

Phone: 888-529-5929

Web: <https://www.citiprogram.org>

## **APÉNDICE B**

### **Autorización del CIPSHI del protocolo de investigación**

Universidad de  
Puerto Rico

COMITÉ INSTITUCIONAL PARA LA PROTECCIÓN DE LOS SERES HUMANOS  
EN LA INVESTIGACIÓN (CIPSHI)  
IRB 00000944

cipshi.degi@upr.edu - <http://graduados.uprrp.edu/cipshi>

**AUTORIZACIÓN DEL PROTOCOLO**

**Número del protocolo:** 1920-096

**Título del protocolo:** Factores determinantes para el aprendizaje de programación de computadoras

**Investigador:** Roberto del C. Martínez Mejía

**Tipo de revisión:**  Inicial  Renovación  Modificación

**Evaluación:**  Comité en pleno  
 Revisión expedita:  
Categoría(s) expedita 45 CFR §46.110: 7

**Fecha de la autorización:** 27 de febrero de 2020

Además, el CIPSHI:

- Concedió la **dispensa** solicitada para modificar el procedimiento estándar de toma de consentimiento informado.

*Cualquier modificación posterior a esta autorización requerirá la consideración y reautorización del CIPSHI. Además, debe notificar cualquier incidente adverso o no anticipado que implique a los sujetos o participantes. Al finalizar la investigación, envíe el formulario de Notificación de Terminación de Protocolo.*

Decanato de  
Estudios Graduados  
e Investigación

18 Ave. Universidad STE 1801  
San Juan PR 00925-2512

787-764-0000  
Ext. 86700  
Fax 787-763-6011

Página electrónica:  
<http://graduados.uprrp.edu>



Margarita Moscoso Álvarez, Ph.D.  
Presidenta del CIPSHI o  
representante autorizado

## **APÉNDICE C**

### **Hoja de consentimiento informado**

## HOJA DE CONSENTIMIENTO INFORMADO

### Factores determinantes en el aprendizaje de la programación

(Versión: 20 de marzo de 2020)

#### Descripción

Usted ha sido invitado/a para participar en una investigación sobre factores que dificultan el aprendizaje de la programación en Ciencias Computacionales. Esta investigación es realizada por Roberto D. Martínez Mejía, estudiante del Programa doctoral en Educación en Currículo y enseñanza en la Subespecialidad de Tecnología del Aprendizaje de la Universidad de Puerto Rico, Recinto de Río Piedras. El propósito de esta investigación es indagar, los factores determinantes en el aprendizaje de la programación de computadoras. Respecto a factores internos tales como nivel de motivación y autoeficacia percibida en el aprendizaje de la programación de computadoras, y factores externos tales como contenidos del curso, ambientes y recursos de aprendizaje más efectivos.

Usted fue invitado/a para participar en esta investigación dado que es estudiante del programa de Ciencia de Cómputos y ha tomado el curso CCOM 3033: Introducción a la programación de computadoras ofertado en el Departamento de Ciencia de Cómputos de la Universidad de Puerto Rico, Recinto de Río Piedras. Se espera que en esta investigación participen aproximadamente 100 personas como voluntarios. Si acepta participar en esta investigación se le solicitará completar un cuestionario, el cual está conformado por seis partes que incluyen preguntas abiertas (escribir la respuesta en los espacios en blanco) y cerradas (seleccionar la respuesta entre un grupo de alternativas).

#### Riesgos y beneficios

Los riesgos asociados con la participación en este estudio son mínimos, dado a que sólo se explorarán datos acerca de temas relacionados con situaciones comunes en el proceso de aprendizaje de la programación. Los riesgos asociados son el cansancio debido a la extensión del cuestionario o la incomodidad ante una pregunta. En caso de sentir incomodidad ante una pregunta, puede dejarla sin contestar e incluso no continuar con el proceso de contestar el cuestionario si así lo decide. Participar en esta investigación no conlleva beneficios directos para los participantes.

Los beneficios esperados de esta investigación son contribuir a mejorar el proceso de aprendizaje y enseñanza de la programación de computadoras.

#### Confidencialidad

La identidad del participante será protegida, ya que su nombre y el curso que está tomando no figurarán en el cuestionario ni en la hoja de consentimiento. Los resultados de la investigación se presentarán como datos agrupados. El investigador Roberto D.

Martínez Mejía, y el Dr. Edwin Martínez Hernández, director de disertación, serán los únicos que tendrán acceso a los datos crudos, incluyendo esta hoja de consentimiento informado.

Para acceder a las respuestas del cuestionario se necesitará una contraseña a la que solo tendrá acceso el investigador. Las respuestas del cuestionario serán conservadas permanentemente, por el investigador, en un banco de datos sin información que pueda identificar a una persona.

Las respuestas del cuestionario serán conservadas permanentemente, por el investigador, en un banco de datos sin información que pueda identificar a una persona. Este banco de datos puede ser utilizado por el investigador para comparar los resultados con los de estudios posteriores y compartirse con otros investigadores interesados en llevar a cabo investigaciones relacionadas.

Dado que este estudio utiliza la computadora como medio para recopilar datos es preciso indicarle que: la información que maneje en la computadora que utilice puede ser intervenida o revisada por terceras personas. Estas personas podrían tener acceso legítimo o ilegítimo a la computadora y su contenido como un familiar, patrono, intrusos o piratas informáticos ("*hackers*"), etc. Además, en la computadora que utilice puede quedar registro de la información que acceda o envíe por Internet.

Oficiales del Recinto de Río Piedras de la Universidad de Puerto Rico o de agencias federales responsables de velar por la integridad en la investigación podrían requerirle al investigador los datos crudos obtenidos en este estudio, incluyendo este documento.

### **Incentivos**

Usted no recibirá ningún tipo de incentivo por su participación.

### **Derechos**

Si ha leído este documento y ha decidido participar, por favor entienda que su participación es completamente voluntaria y que usted tiene derecho a abstenerse de participar o retirarse del estudio en cualquier momento, sin ninguna penalidad. También tiene derecho a no contestar alguna pregunta en particular. Además, tiene derecho a recibir una copia de este documento.

Si tiene alguna pregunta o desea más información sobre esta investigación, por favor comuníquese con el investigador principal, Roberto D. Martínez Mejía a [roberto.martinez6@upr.edu](mailto:roberto.martinez6@upr.edu) o llamando al teléfono (787) 484-7615. Además, puede comunicarse con el director del Comité de Disertación Dr. Edwin Martínez Hernández al teléfono (787) 764-0000, ext. 89068 o al correo electrónico: [edwinjose.martinez@upr.edu](mailto:edwinjose.martinez@upr.edu)

De tener alguna pregunta sobre sus derechos como participante o reclamación o queja relacionada con su participación en este estudio puede comunicarse con la Oficial de Cumplimiento del Recinto de Río Piedras de la Universidad de Puerto Rico, al teléfono (787) 764-0000, ext. 86773 o a [cipshi.degi@upr.edu](mailto:cipshi.degi@upr.edu).

Marcar el óvalo de "**Acepto**" significa que ha decidido participar en este estudio después de haber leído la información presentada en esta hoja de consentimiento y que podrá guardar o imprimir la misma

Acepto

No acepto

## RESUMEN BIOGRÁFICO DEL AUTOR

Roberto D. Martínez Mejía, nació en Barranquilla-Colombia. Completó de manera simultánea estudios de ingeniería de sistemas en la Universidad Simón Bolívar y de Ciencias de la Educación especialidad lenguas modernas Español-Inglés en la Universidad del Atlántico, obteniendo la máxima distinción en su tesis de grado: Diseño e implementación de Material Educativo Computarizado “MEC *Learning English*” sobre gramática inglesa en secundaria. Posteriormente realizó una especialización en pedagogía de las ciencias en la universidad Simón Bolívar. Seguidamente una maestría en Educación con especialidad en Educación superior. En la cual desarrollo como tesis de investigación el tema: "Estrategias de comprensión lectora mediadas por TIC. Una alternativa para mejorar las capacidades lectoras en secundaria". En el año 2016 comenzó sus estudios doctorales en la Universidad de Puerto Rico Recinto de Rio Piedras. Roberto cuenta con diez años de experiencia en la enseñanza, desde nivel elemental a secundario, en el área de tecnología. En particular, la enseñanza de la programación de computadoras. Paralelamente se ha desempeñado como desarrollador Full Stack aplicando sus conocimientos en diseño web, lenguajes de programación, base de datos y servidores. Como estudiante PEAf, desarrolló la plataforma del CEA (Centro para la Excelencia Académica), la cual permite matricular talleres, webinars y módulos en línea. También permite descargar certificados en línea y generar informes estadísticos del número de participaciones por escuela facultades o departamento del Recinto. De igual manera pudo desarrollar el sistema de registro para admisiones al Programa de Estudios de Honor y es Sistema de Matrículas de cursos de la DECEP. También se ha destacado

en el campo investigativo siendo autor de varios artículos de investigación y ponente en congresos educativos. Sus líneas de investigación son: Aplicación de tecnologías de la información y la comunicación en la enseñanza y estrategias para el aprendizaje inicial de la programación de computadoras.