

DEFICIENCIA DE ADQUISICIÓN DE COMPETENCIAS MÍNIMAS EN ESTUDIANTES DE DESARROLLO DE SOFTWARE: HACIA UN NUEVO MODELO DE ENSEÑANZA PEDAGÓGICO PRAXEOLÓGICO

(DEFICIENCY OF ACQUISITION OF MINIMUM SKILLS IN STUDENTS OF SOFTWARE DEVELOPMENT: TOWARDS A NEW MODEL OF PEDAGOGICAL TEACHING)

Helmer Julián Romero Romero

eromero@uniminuto.edu

Corporación Universitaria Minuto de Dios – UNIMINUTO

Bogotá - Colombia

Luis Eduardo Cano Olivera

luis.Cano@uniminuto.edu

Corporación Universitaria Minuto de Dios – UNIMINUTO

Bogotá - Colombia

Carlos Guillermo Charry Mora

ccharry@uniminuto.edu

Corporación Universitaria Minuto de Dios – UNIMINUTO

Bogotá - Colombia

Jorge Rolando Pardo Morales

jorge.Pardo@uniminuto.edu

Corporación Universitaria Minuto de Dios - UNIMINUTO

Bogotá - Colombia

Fotografía de portada y Diagramación

Sindy Catherine Charcas Ibarra

Encuentre este artículo en:
<http://revistas.uniminuto.edu/index.php/IYD>

Fecha de recepción: 8 de febrero de 2019
Fecha de aceptación: 27 de marzo de 2019
Fecha de publicación: 30 de junio de 2019

RESUMEN

Objetivo: Identificar un nuevo modelo de enseñanza pedagógico praxeológico que contribuya a la adquisición de competencias mínimas en los estudiantes de informática para el desarrollo de aplicaciones de software.

Metodología: Se aplicó la metodología propuesta por los doctores Ferreiro & Cummings (2012) llamada MeProb, la cual, contribuyó a determinar y formular un posible problema de investigación para el desarrollo de un proyecto de investigación de una manera ordenada y sistemática, facilitando la aplicabilidad de instrumentos a los estudiantes del programa de Tecnología en Informática (TINF) de los tres(3) semestres, al cuerpo docente, egresados y al sector empresarial de las Sedes Principal Calle 80, Bogotá Sur y Soacha de la Corporación Universitaria Minuto de Dios- UNIMINUTO, logrando identificar problemas, dificultades y/o inconvenientes que tienen los estudiantes del programa para adquirir las competencias mínimas necesarias para el desarrollo de software.

Resultados: Se pudo demostrar que los estudiantes no adquieren las competencias mínimas para la lógica en programación por distintos factores como la dificultad de aprendizaje, falta de equipos en una Institución para reforzar lo aprendido, el idioma en el que está escrito el lenguaje de programación, y la dificultad para analizar y resolver problemas y ejercicios propuestos, siendo necesario aplicar un nuevo modelo de enseñanza pedagógico praxeológico distinto.

Conclusión: Se concluye que existen diferentes causas para que un estudiante de informática no llegue a adquirir las competencias mínimas necesarias para el desarrollo de cualquier software, y que las metodologías de enseñanza existentes empleadas por el docente de acuerdo a los lineamientos de los micro currículos o syllabus, no son siempre del todo convenientes debido a que cada estudiante tiene una forma de aprender distinto, por lo que se requiere aplicar un nuevo modelo de enseñanza pedagógico praxeológico que contribuya a fortalecer y adquirir las competencias en los estudiantes en informática.

Palabras clave: programación, lenguaje de programación, estrategias didácticas, lógica de programación, desarrollo de software, pedagogía, andragogía.

ABSTRACT

Objective: To identify a new model of pedagogical teaching that contributes to the acquisition of minimum competences in computer science students for the development of software applications.

Methodology: The methodology proposed by Doctors Ferreiro & Cummings (2012) called MeProb was applied, this one contributed to determine and formulate a possible research problem for the development of a research project in an orderly and systematic way, facilitating the applicability of instruments to the students of the Computer Technology program (TINF), professors the program, the faculty, graduates and to the business sector of the Sede Principal Calle 80, Bogotá Sur and Soacha of the Minuto de Dios University - UNIMINUTO, managing to identify problems, difficulties and / or inconveniences that the students of the program have to acquire the minimum competences necessary for software development.

Results: It was possible to demonstrate that the students do not acquire the minimum competences for logic in programming due to different factors, such as the difficulty of learning, lack of equipment in an Institution to reinforce what has been learned, the language in which the programming language is written, and the difficulty to analyze and solve problems and proposed exercises; this is the reason why, it is necessary to apply a new teaching model.

Conclusion: It is concluded that, there are different causes when a computer science student does not acquire the minimum competences necessary for the development of any software, and that the existing teaching methodologies used by the teacher according to the guidelines of the micro curricula or syllabus, are not always convenient because each student has a different way of learning, so that, it is necessary to apply a new model of teaching that contributes to strengthen and acquire skills in computer science students.

Keywords: programming, programming language, didactic strategies, programming logic, software development, pedagogy, andragogy.

INTRODUCCIÓN

En la era de la información y las comunicaciones, la humanidad, la sociedad en general y el mundo entero depende cada día más del uso de distintas aplicaciones informáticas para el desarrollo de cualquier actividad en la cual se desenvuelva cada individuo, siendo, además, un factor indispensable y necesario para la economía global, el avance tecnológico y científico, los modelos de enseñanza tanto a las nuevas generaciones como a la andragogía aplicada a las viejas generaciones a través de la usabilidad, accesibilidad y generación de nuevos conocimientos mediante el uso de estas Tecnologías de la Información y las comunicaciones (TICs) que están transformando cada día más a la sociedad en general.

Para que estas aplicaciones puedan ser una realidad y satisfacer las necesidades del mercado en general, se requieren de personas calificadas en el dominio y destreza de cualquier lenguaje de programación que puedan dar solución a cualquier problema y que logren satisfacer cualquier tipo de reto que la sociedad les demande, y esto solo es posible, gracias a los llamados, “programadores de software (o programmers) o “desarrolladores de aplicaciones informáticas” (developers), una noble profesión que se debe, gracias a la academia que contribuye en la formación de personal cualificado para suplir la demanda del mercado, pero que muchas veces es poco valorada y/o incentivada.

El papel de importancia que tienen los programadores de software o desarrolladores de aplicaciones informáticas contribuyen a que día a día se cuenten con aplicaciones tan sencillas como una simple calculadora hasta aplicaciones tan avanzadas que auto aprenden a través de algoritmos de inteligencia artificial o aplicaciones informáticas que facilitan la inteligencia de negocios y permitan la toma de decisiones en una organización, contribuyendo a revolucionar día a día, el mundo a nivel tecnológico y científico. Pero existe un gran inconveniente y es que cada día son menos estos profesionales, surgiendo algunos interrogantes, como ¿Cuál sería el problema y dificultades que puedan tener estos futuros profesionales informáticos en formación? ¿Cuál sería la necesidad de que los estudiantes en informática adquieran las competencias mínimas para el aprendizaje de cualquier lenguaje de programación para resolver cualquier tipo de problema? ¿Qué tipo de estrategias de andragogía serían vitales para que los adultos adquirieran las competencias mínimas necesarias para el dominio y destreza de cualquier lenguaje de programación? ¿Cuál es el impacto que tiene la programación en la sociedad en general?

Para responder a algunos de estos interrogantes, es necesario comprender como la “programación de aplicaciones”, está contribuyendo a que cada día se esté generando nuevo conocimiento e información, la cual es subida diariamente a Internet o a la denominada Nube (Cloud) de una manera exorbitante a través de computadores, portátiles, tablets, phablets, smart phones y smarth watches por millones de usuarios y que estén accesibles a nivel global, contribuyendo a que se realicen estudios de investigación científico y tecnológico, encontrándose hasta infografías que demuestran el uso de la información que es compartida por la mayoría de usuarios en sitios como el encontrado en el portal de

Juan Marquina (2013), que refleja la cantidad de videos, audios, documentos, imágenes, mensajes, publicaciones y un sin número de información digital que es aprovechada para distintos usos.

De igual manera, gracias al “desarrollo de la programación”, se ha logrado transformar distintas profesiones que giran alrededor de la tecnología que de una u otra manera contribuyen a que surjan nuevos perfiles que antes ni existían, como son: *Internautas o cibernautas (personas que consumen contenido de Internet)*, *proconsumers (aquellos que además de consumir contenido, son generadores de contenidos)*, *smart-consumers (aquellos que buscan, preguntan y se informan antes de comprar)*; *community managers (aquellos que se encargan de posicionar un sitio web o una marca de una empresa)*, *youtubers (aquellos que realizan videos a través de YouTube)*, *tuiteros (aquellos que publican mensajes, videos o imágenes de cualquier tema a través de la red de Twitter)*, *instagrammers (Aquellos que publican en la red de Instagram)*, *whatsapperos (aquellos que comparte contenido e interactúan con sus contactos a través del servicio de mensajería Whastapp)*, *Científicos de datos (personas especializadas en matemáticas y estadísticas y que dominan la programación y sus diferentes lenguajes, ciencias de la computación y analítica aprovechando los grandes volúmenes de información como el BigData)*, y la lista puede seguir creciendo exponencialmente. (Ver gráfico No 2)



Gráfico No 2 Tipos de Usuarios – Fuente autores

Gracias al trabajo de los programadores, se puede encontrar una gran riqueza de herramientas informáticas que ellos suben constantemente a plataformas de versionamiento licenciadas ya sea bajo la licencia pública general (GPL, 1984) de la Free Software Foundation, el Open Source o cualquiera de sus variantes o de licenciamiento privativo (copyright), facilitando que se pueda acceder al código fuente de las aplicaciones informáticas desarrollada por múltiples programadores alrededor del mundo como son las “fuentes de código de programación” encontradas en plataformas para desarrolladores como Github(2018), SourceForge(2013), Team Foundation Server (2018), Google Cloud Source (2015), entre otras. Un auge de herramientas informáticas que son utilizadas en distintos tipos de dispositivos desde un laptop hasta un teléfono inteligente e inclusive encontrados en el llamado “Internet de las cosas – IoT (dispositivos inteligentes que se pueden comunicar entre sí a través de internet).

Ahora bien, para entender las competencias mínimas que un “programador” debe adquirir, es necesario entender el proceso que debe realizar el programador para el desarrollo de aplicaciones informáticas o como se le conoce comúnmente como programación de computadores o de software, y que consta de una serie de etapas conocido como el “Ciclo de vida de la Ingeniería de Software”, en donde, se contempla las siguientes partes a seguir: requerimientos (entender el problema), diseño

(elaborar el algoritmo), análisis (entender la funcionalidad del algoritmo), codificación (escribir el código en un lenguaje de programación específico), pruebas (para garantizar la funcionalidad del programa desarrollado) y liberación (entrega del software funcionando) (Sommerville, 2011), logrando al final, un producto de software definitivo para cualquier entidad. (Gráfico No 1)

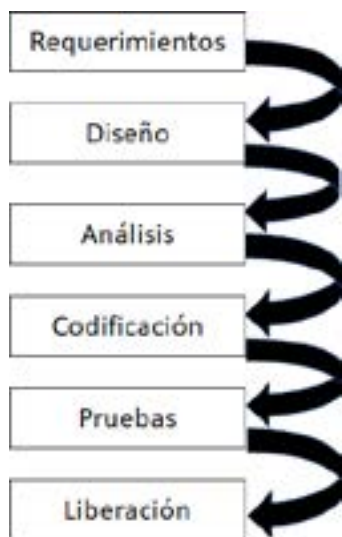


Gráfico No 1 – Ciclo de la Ingeniería de Software según Sommerville. Diseño autores.

Además de realizarse las distintas etapas del Ciclo de vida de la ingeniería de software, un desarrollador de software, y en especial, que salga recién egresado de la academia, se pueda desenvolver en el campo laboral, debe contar mínimo con unas competencias específicas profesionales relacionados con el área del conocimiento en programación, y que como plantea Aguerrondo (2009), no se deben confundir con los “logros de aprendizaje”. Y que según Tobón (2007), “las competencias se pueden contemplar desde cuatro (4) perspectivas:

- 1. Conductual:** Que las define como los comportamientos clave de las personas para ser competitivas en una organización.
- 2. Funcionalista:** que las ve como atributos que se deben tener para satisfacer los propósitos laborales - profesionales a partir de funciones definidas.
- 3. Constructivista:** donde son habilidades, conocimientos y destrezas que permiten resolver dificultades en los procesos laborales-profesionales desde el marco organizacional.
- 4. Compleja:** que relaciona las actividades y problemas con idoneidad y ética a desarrollar, buscando la realización personal, la calidad de vida y el desarrollo social y económico sostenible y en equilibrio con el ambiente.” (Tobón, 2007)

También plantea Tobón (2007), que las competencias se pueden lograr mediante “Procesos complejos de desempeño con idoneidad en determinados contextos, integrando diferentes saberes (saber ser, saber hacer, saber conocer y saber convivir), para realizar actividades y/o resolver problemas con sentido de reto, motivación, flexibilidad, creatividad, comprensión y emprendimiento, dentro de una perspectiva de procesamiento metacognitivo, mejoramiento continuo y compromiso ético, con la meta de contribuir al desarrollo personal, la construcción y afianzamiento del tejido social, la búsqueda continua del desarrollo económico-empresarial sostenible, y el cuidado y protección del

ambiente y de las especies vivas” (Tobón, 2007).

Ahora bien, en cuanto a la “programación de computadores”, plantea el Dr. Mitchel Resnick (2013), hablando acerca de la importancia de “aprender a programar” y cómo adquirir estas competencias desde la niñez hasta la adultez en andragogía mediante el uso de *Cursos En Línea Masivos y Abierto – Massive Online Open Courses (MOOC)* como la plataforma de enseñanza **code.org** (2013), menciona en su artículo que es tal la cantidad de oportunidades de trabajo que salen a diario, que cualquier persona debería aprender a programar sin importar la edad, o programar para aprender y que se puede empezar utilizando una herramienta como “**scratch**” (2011), debido a que una persona puede “comprender las ideas matemáticas y computacionales, tales como variables y condicionales, y simultáneamente están aprendiendo estrategias para solucionar problemas, diseñar proyectos y comunicar ideas. Esas habilidades son útiles no solo para los científicos de la computación sino para todas las personas sin distinción de edad, proveniencia, intereses u ocupación.” (Resnick, M, 2013)

Para poder realizar una investigación que contribuya a identificar los inconvenientes que pueda tener un estudiante de informática, se hace necesario, tomar como punto de referencia al Programa de Tecnología en Informática (TINF) de la Corporación Universitaria Minuto de Dios (UNIMINUTO), la cual, no es ajena al tema de investigación y que contribuya a mejorar y fortalecer los procesos de mejora y calidad del programa, por lo que ha visto la necesidad de estudiar e investigar a través de los estudiantes que cursan “programación básica”, los inconvenientes que tienen ellos para adquirir las bases y competencias mínimas necesarias como “lógica de programación”, indispensable para el desarrollo de cualquier aplicación informática. Investigación que comenzó a partir de reuniones de profesores del Programa de Tecnología en Informática (TINF) de la Corporación Universitaria Minuto de Dios (UNIMINUTO) Sede principal del semestre 2015-10; dando origen al proyecto de investigación denominado “Plataforma Colaborativa para el Aprendizaje de Programación Básica haciendo uso de mapas de calor como herramienta de apoyo a la realimentación didáctica” (ALGORINFO) desarrollado en el año 2016. Sin embargo, el equipo de investigadores determinó la imperiosa necesidad de complementar la herramienta, con el “Diseño de un Modelo de Enseñanza de la Programación Básica” que se incluyera en los cursos de la línea en “programación básica” en donde se ofrece el programa de TINF en UNIMINUTO, vinculando, además de la sede principal en Calle 80, a las sedes de Bogotá Sur y Centro Regional Soacha. Además, que la asignatura de “programación básica” se encuentra en consonancia con el “modelo praxeológico” de UNIMINUTO, en el que se “busca promover el desarrollo de todas las dimensiones de la persona, partiendo de las potencialidades de cada uno y teniendo en cuenta el conjunto de ideas, creencias, valores, actos, palabras y contextos del estudiante o comunidad, para ayudarle(s) a construir su bienestar personal y comunitario. Así mismo, busca formar sujetos capaces de reflexionar críticamente sobre sus prácticas y generar, desde ellas, conocimiento innovador y pertinente, así como articular su trabajo y proyecto de vida al proyecto social de nación.” (Uniminuto, 2018). Situación que ha permitido a que se desarrolle un proyecto de investigación articulado con las distintas sedes de Uniminuto del programa,

aplicando una serie de instrumentos a estudiantes de la línea de programación de los tres (3) semestres, docentes del programa TINF y sector empresarial para identificar los inconvenientes que presentan los estudiantes para adquirir las competencias mínimas necesarias en los estudiantes para fortalecer la *lógica de programación* para desarrollar aplicaciones informáticas de acuerdo a la demanda del mercado laboral.

La necesidad de programadores en el mundo

La globalización, evolución e interconectividad de las TICs a nivel mundial, ha tenido incidencia en las distintas áreas del conocimiento creando una dependencia de aplicaciones informáticas, en donde se requiere contar con profesionales cualificados que cuente con las competencias mínimas necesarias para cumplir con los requerimientos científicos y tecnológicos que la sociedad demande a través del desarrollo de soluciones informáticas. Sin embargo, se ha visto como en los últimos años, *las carreras de formación profesional a nivel técnico y tecnológico como: desarrollo de software, diseño y desarrollo web, desarrollo de dispositivos móviles, tecnología en informática, tecnología en computación y tecnología en arquitectura de software* así como a nivel profesional como: *ingeniería de sistemas, ingeniería de sistemas y computación, ingeniería informática, ingeniería de software, ingeniería electrónica y administración de sistemas informáticos* son cada año menos el número de egresados que salen de las distintas instituciones educativas que las ofrecen y cada vez menor el número de nuevos aspirantes que aplican a dichos programas de formación profesional, fenómeno que ha sido abordado en distintos medios noticiosos y portales especializados desde el año 2008 hasta nuestros días tanto a nivel nacional publicando acerca de la escasez de profesionales en programación como son: La República (2016), El Tiempo (2015), El espectador (2015), IDX (2017), Revista Dinero (2008) y Semana (2016) o a nivel internacional como: El Clarín (2017), la Nación(2017) y Radio mitre(2017) de Argentina; Confidencial (2017) , IT User (2016) y TIC Beat (2016) de España; Mi Ambiente y Expansión de México; Gestión (2015) de Perú, entre otros, hablando no sólo acerca de la escasez de estos profesionales sino también abordando las razones para que exista dicha escasez de profesionales expertos en programación, en donde se analiza los problemas que tienen las empresas para conseguir perfiles especializados como: *administradores de redes, jefes de proyectos, programadores Python, Java y .Net, gerentes de Tecnología y Sistemas, administradores de Seguridad y Networking, jefes de Sistemas, científicos de datos, analistas de Información y Datastage*, entre otros.

Razones de la falta de programadores En El Mundo

Dentro de las razones expresadas por los distintos medios científico-tecnológicos y expertos en el tema, y que se categorizan y definen por los autores soportadas a partir de otros estudios de investigación realizados, se contemplan:

No.	Razón para que un estudiante no adquiera competencias mínimas de programación
1	Salarios no dignos y mal remunerados que las empresas no están dispuestas a pagar por personal cualificado
2	El errado concepto de programador que tienen las empresas
3	Empresas con malas culturas acerca de la ingeniería
4	Pérdida de protagonismo de las profesiones en tic
5	Problemas con la industria del software en Colombia y Fedesoft que definen las habilidades y competencias que requieren de los profesionales
6	Profesionales con título y sin título sin experiencia
7	Certifiquitis y costos de las mismas
8	Ciudades con poca demanda de profesionales
9	Desmotivación o desinterés por las asignaturas de programación
10	Escasez de género en profesionales de tic
11	Programas de tics no orientados a personas con discapacidad
12	Falta de orientación e información de las profesiones en tics
13	Problemas de lógica de programación
14	Mallas curriculares de apagaincendios

Tabla No 1 – Razones – Fuente autores.

I. Salarios no dignos y mal remunerados que las empresas no están dispuestas a pagar por personal cualificado: Para Dans (2007), lo define como la “*cicatería de las empresas*”. Y para el diario República (2010), menciona que las empresas por el afán de incrementar sus utilidades han contribuido al detrimento de la remuneración de los profesionales egresados en las distintas ramas de las TICs. Cuando Torrenegra (2015), realizó una encuesta no oficial sobre lo que devengaba un desarrollador de aplicaciones, se generó polémica, al afirmar que un profesional ganaba en promedio seis (6) millones de pesos, y que en dicho estudio no aclaró que muchos de los desarrolladores que participaron de la misma ganaban dichos sueldos porque trabajaban en empresas multinacionales de EE.UU o que tenían títulos y que fueron pocos los desarrolladores colombianos que participaron de la misma con bajos sueldos o que no tenían título, y que al sacar el promedio salarial de un desarrollador, terminó dando una cifra errada y que no se ajustaba a la realidad del país. Dicho estudio, fue desmontado de su sitio oficial al darse esa polémica que se subsistió. Buritica (2015) y Colombia.dev replanteó el estudio de la investigación sobre los sueldos y la satisfacción de los profesionales a partir de otra encuesta en línea y que resumió Przybilla (2016), la cual, puede descargar el dataset del github(2008) de Colombia.dev(2016), en donde se evidencia que el desarrollador es mejor valorado económicamente a nivel internacional que a nivel nacional, dependiendo de los títulos y experiencia que tenga, pero que en el país son pocos valorados nivel salarial.

2. El errado concepto de programador que tienen las empresas:

Para Dans (2007) expresa que las empresas buscan: *“Un profesional mucho más autosuficiente, con conocimientos de ingeniería del software, teoría de la computación, matemática, algorítmica e incluso nociones de estrategia de negocio, que desempeña una actividad de elevada cualificación y responsabilidad.”* Un error común porque consideran que un programador debe realizar todas las etapas del ciclo de vida de la ingeniería de software (Sommerville,2011), y las empresas no cuentan con equipos de desarrollo que se encarguen de funciones específicas, sino que exigen que el programador sea el todero desde el inicio hasta el final del producto de software.

3. Empresas con malas culturas acerca de la ingeniería:

Para Buriticá (2015), expresa *“que las empresas son responsables tanto de lo que contratan como de lo que exige el mercado local, y se requiere de empresas que sean mejores en la cultura de la ingeniería para los profesionales que deseen trabajar”.* Las empresas al no tener claro el perfil profesional incluyen perfiles equivalentes creyendo que reúnen las competencias mínimas para el desarrollo de software, y que sea difícil de aplicar para cualquier profesional en informática.

4. Pérdida de protagonismo de las profesiones en tic:

Para Ulloa(2008), expresa que dicha pérdida de protagonismo, se debe a *“la pobre preparación de los estudiantes en Matemáticas y Ciencias Naturales (física, química y biología). Los estudiantes no ven estas asignaturas con gusto, ni la enseñanza de ellas es agradable. Todavía existe un fuerte componente de aprendizaje memorístico, sin que se haya encontrado un sentido claro a la enseñanza de estas ciencias. Con esta situación como materia prima, los estudiantes de secundaria y de media no desean continuar profundizando en estas ciencias que son las bases de la Ingeniería; tampoco, según sus propias palabras, se sienten bien formados y quizás los “rajén” en los primeros semestres.”* Para Sheppard et al. (2017), en un estudio que hicieron para The Carnegie Foundation sobre la educación de ingeniería en dicha institución, plantea que *“el modelo aplicado para las escuelas de tecnología e ingeniería están influenciadas por tradiciones académicas provenientes del modelo francés basado en un currículo de ciencias básicas, temas técnicos y de humanidades, empleando un modelo más de tipo teórico que práctico, aplicándose hasta nuestros días”.* Situación en la que los estudiantes no contemplan la importancia ni relación que les puede servir para adquirir y desarrollar de competencias necesarias como la lógica de programación para el desarrollo de software, a partir de la resolución de ejercicios matemáticos y físicos.

5. Problemas con la industria del software en colombia y fedesoft que definen las habilidades y competencias que requieren de los profesionales:

Para Buriticá (2015), menciona que *“la industria está compuesta por dos (2) grandes sectores: software empresarial y consultorías de software. Y varios de los miembros de la junta directiva de Fedesoft son los que se encargan de proveer servicios en cualquiera de esos dos (2) sectores, y son las empresas las que determinan cuales habilidades se necesitan para hacer parte de su población activa. Siendo el mercado laboral quien determina qué habilidades anticuadas se requieren para ser empleado y las escuelas y universidades no pueden escoger enseñar a sus estudiantes sobre tecnologías vanguardistas.”.*

6. Profesionales con título y sin título sin experiencia: En el estudio de Colombia.dev (2016) y resumido por Pzybylla

(2016), se refleja que hay muchos desarrolladores que son freelancers frente a los profesionales con títulos que trabajan para empresas por contratos largos, y que las empresas se inclinan más por desarrolladores seniors de más de 5 años de experiencia que por desarrollares juniors o que no tengan experiencia.

7. Certifiquitis y costos de las mismas: Las empresas exigen para aplicar a cualquier cargo que el profesional esté certificado en las distintas herramientas y tecnologías que existan, pero sus altos costos hacen que el profesional no pueda acceder a las mismas y que las empresas que requieren dichos perfiles no asuman capacitar ni costear a dichos profesionales.

8. Ciudades con poca demanda de profesionales:

Con el estudio de Colombia.dev(2016) y resumido por Pzybylla (2016), se evidencia que las ciudades principales en donde hay más desarrolladores y con mejores condiciones económicas son Medellín y Cali, en donde incluso solicitan que los desarrolladores estén certificados en metodologías de desarrollo ágil como Scrum (Kniberg,2008) por encima de Bogotá.

9. Desmotivación o desinterés por las asignaturas de programación:

En la revista Semana (2013) en una entrevista realizada a Jason Ohler, menciona que *“en los colegios no se enseña Informática sino Ofimática, y que esto ha generado un atraso en el país, mientras que en otros países a un niño se le enseña desde temprana edad a programar software, en Colombia no se hace”.* De los estudiantes que ingresan a una Institución educativa a estudiar una carrera basada en desarrollo de software, sólo el 2% puede tener algo de nociones en programación impartida por lo general en colegios privados o que haya adquirido sus conocimientos de manera autodidacta a través de cursos en línea.

10. Escasez de género en profesionales de tic: Otro aspecto a resaltar, dentro del estudio de Colombia.dev(2016) y resumido por Pzybylla (2016), se evidencia la poca existencia de mujeres desarrolladoras de software, y que hasta en la misma la academia se encuentran pocas estudiantes mujeres que optan por tomar una profesión en TICs.

11. Programas de tics no orientados a personas con discapacidad:

La formación está orientada a personas que no tienen discapacidad, al igual que los laboratorios de tecnología o aulas virtuales e incluso hasta el material bibliográfico, excluyen a personas con discapacidad que estén interesadas en el desarrollo de software, y sólo se les tiene en cuenta en proyectos de investigación en donde se les dé a ellos herramientas tecnológicas como las propuestas realizadas por Romero-Díaz, Rodríguez-Castro & Cortés-Trujillo (2014) en la Universidad Distrital de cubrir alguna necesidad de las personas con limitaciones, pero no incentivarlos a que estas personas con limitaciones sean ellos mismos los creadores y desarrolladores de software.

12. Falta de orientación e información de las profesiones en tics:

En una nota para la revista Semana (2012), ofrecida por la Red de Programas de Ingeniería de Sistemas afines – REDIS (1991), se menciona que en los colegios no brindan información suficiente acerca de las profesiones en TICs distinto a como ocurre con otras profesiones, incidiendo en que los aspirantes no quieran estudiar dichas carreras de formación profesional.

13. Problemas de lógica de programación: Para Fuentes-Rosado & Moo-Medina (2017), mencionan que *“para aprender a programar mediante un lenguaje de programación, existen*

diferentes maneras de solucionarlo y que cada programador tiene una forma diferente para programarlo” lo cual, se logra mediante la habilidad para resolver problemas a través de la lógica de programación.

14. Mallas curriculares de apagincendios: Para Sheppard et al. (2017), “Los informes de The Carnegie Foundation recomiendan que las escuelas profesionales, debido a que son responsables de la preparación de los profesionales, deben apuntar a un enfoque cada vez más integrado para formar en los estudiantes razonamiento analítico, destrezas prácticas y juicio profesional. Aunque algunas escuelas de ingeniería han introducido programas, métodos de enseñanza o estructuras curriculares que tratan de integrar estas metas profesionales, ninguna ofrece un enfoque integral en red” Las empresas al requerir personal cualificado con conocimientos específicos puntuales, obligan a que las instituciones educativas dentro de sus mallas curriculares, opten por el uso de las electivas que no tienen siempre una continuidad y dependen más del docente experto de dicha área específica que del propio programa, haciendo que los egresados salgan con distintas competencias profesionales específicas, es decir, que un estudiante puede profundizar en un determinado lenguaje de programación mientras que otro estudiante, termine profundizando en un sistema de gestión de bases de datos, por ejemplo, y que son conocimientos que un futuro profesional debe saber.

Problema de investigación

El problema de investigación es la falta de competencias mínimas de los estudiantes para la programación de aplicaciones de software.

Propósito de la investigación

Desarrollar un nuevo modelo de enseñanza pedagógico praxeológico para fortalecer el proceso de adquisición de competencias mínimas de los estudiantes de informática como desarrolladores de software.

Objetivos de la investigación

- Identificar problemas de los estudiantes en cursos introductorios de la línea de programación básica
- Definir requerimientos del nuevo modelo de enseñanza pedagógico-praxeológico para la programación básica.
- Diseñar componentes pedagógicos y técnicos que fortalezcan el proceso de adquisición de competencias mínimas en los estudiantes de programación básica.

Preguntas de la investigación

Las preguntas surgidas del proyecto de investigación son:

- ¿Cuáles son las dificultades y problemas de aprendizaje en la adquisición de competencias mínimas en los estudiantes requeridos para el desarrollo de aplicaciones informáticas de los cursos introductorios?
- ¿Cuáles son los modelos de enseñanza pedagógico-praxeológico que contribuyen al fortalecimiento y mejoramiento del desarrollo de las competencias mínimas para la programación de aplicaciones de software?
- ¿Cuáles componentes pedagógicos y técnicos contribuyen y que se pueden implementar para fortalecer el proceso de adquisición de competencias mínimas en los estudiantes para

que desarrollen software?

- ¿Cómo diseñar un modelo de enseñanza pedagógico-praxeológico, que vaya a la par de los lineamientos definidos para un programa de formación en tecnología en informática como el ofertado por Uniminuto?

Resultado esperado

Tener un modelo educativo pedagógico-praxeológico que contribuya al mejoramiento y fortalecimiento de las competencias mínimas en los estudiantes para que puedan desarrollar aplicaciones informáticas.

Caso de estudio: programa de tecnología en informática de uniminuto

El proyecto de investigación es aplicable a cualquier rama de la tecnología en donde se ofrezca un programa académico basado en el desarrollo de aplicaciones informáticas. Para poder realizar la investigación se opta por analizar la situación actual dentro del Programa de Tecnología en Informática de la Corporación Universitaria Minuto de Dios - UNIMINUTO de las sedes principal Calle 80, Bogotá Sur y Centro Regional Soacha para los estudiantes de los tres (3) primeros semestres y vincular dentro del estudio a los docentes y al sector empresarial que tengan o no vinculo de práctica empresarial con la Institución, y así poder establecer las bases y lineamientos para el diseño y aplicabilidad en un futuro de un modelo de enseñanza pedagógico-praxeológico que contribuya al fortalecimiento y mejoramiento de las competencias mínimas para el desarrollo de software en los estudiantes del programa, y suplir profesionales especializados que las empresas requieran.

De una muestra de estudiantes del programa TINF de Uniminuto de las tres(3) sedes y que estuvieran cursando las asignaturas pertenecientes a la línea de programación como Programación Básica, Fundamentos de Programación Orientada a Objetos, Programación Orientada a Objetos (POO), Fundamentos de Estructuras de Datos y Estructuras de Datos, mediante uno de los instrumentos realizados por los autores, se les pregunto, que *¿Cuál era la causa principal que le impedía la comprensión total de las temáticas de programación básica?*, arrojando un resultado evidenciado en la Figura No 1, en donde un 58,8% manifiesta que una de las razones es la metodología de enseñanza-aprendizaje impartida por los docentes, un 27,5% que han perdido interés en las asignaturas de la línea de programación, un 6,3% de los estudiantes manifiestan que los docentes no motivan en la asignatura, un 1,3% que los docentes no dominan la asignatura y un 6,1% todas las anteriores.

A los docentes de cada una de las sedes dentro del instrumento aplicado, se les preguntó que, *¿Cuál era la causa principal que les impide a los estudiantes la comprensión total de las temáticas de programación básica?* En los resultados arrojados de la Figura No 2, se contempla que un 66,7% de los docentes manifiestan que los estudiantes presentan problemas de lógica de programación, un 33,3% además de la falta de lógica de programación, le añaden que a los estudiantes les falta interés en la asignatura, la falta de conocimientos en ciencias básicas y la metodología de enseñanza-aprendizaje.

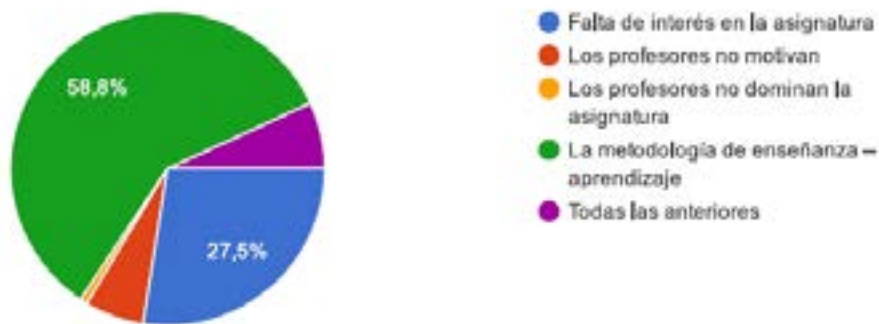


Figura No 1 - Encuesta realizada a estudiantes del programa de Tecnología en Informática de Uniminuto. Fuente Autores

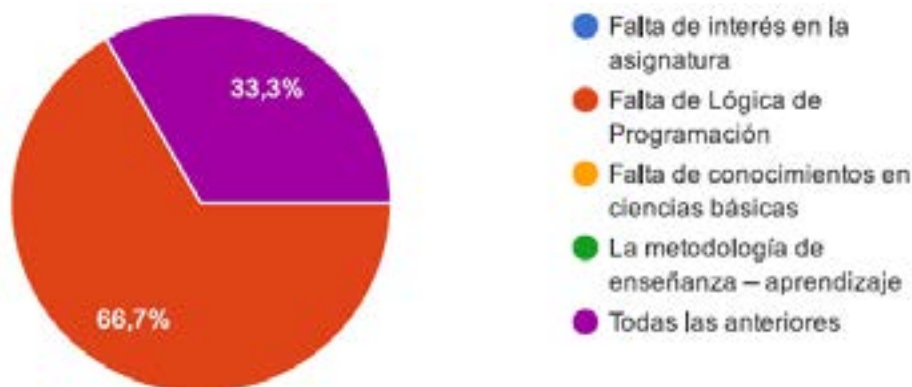


Figura No 2 – Resultados encuesta a docentes del programa de tecnología en informática de Uniminuto – Fuente autores

A las empresas que tiene vinculo de práctica profesional con Uniminuto o no, se les aplicó un instrumento en donde se les preguntó que “Para contratar a un profesional en programación, ¿Qué tipo de conocimientos técnicos y de competencias laborales mínimas considera que debe dominar para desarrollar soluciones informáticas?”

Obteniéndose respuestas muy variadas resaltando algunas como conocimientos técnicos, dominio de un lenguaje de programación, lógica de programación y competencias para aplicar a cualquier etapa del ciclo de vida de la ingeniería de software, y que tengan competencias blandas para poder formarlos desde la misma empresa.

Conclusiones

La investigación demuestra que existen diversas razones para que se evidencie escasez de profesionales que cumplan con las competencias mínimas para el desarrollo de software de acuerdo con las necesidades del mercado, las cuales dependen en parte de los modelos de enseñanzas existentes, en otras de las exigencias impuestas por la demanda del mercado laboral y en otras de las técnicas de estudio y motivación que tenga un estudiante en informática para el aprendizaje de cualquier lenguaje de programación.

Unapartedelproblema,seevidenciaenlasinstituciones educativas de formación profesional, las cuales se rigen por un modelo general de formación profesional establecido en una malla curricular que contempla ciencias básicas, técnicas y de humanidades, definidas en los microcurrículos o syllabus de cada asignatura, en el que se establece de manera general, la metodología de enseñanza que los docentes deben brindar a los estudiantes de informática, a través de clases magistrales, talleres, proyectos de aula, aulas virtuales, aprendizaje autónomo, trabajos colaborativos, mecanismos

evaluativos (quices, parciales, exposiciones, ejercicios), al igual que los libros de consulta bibliográficos especializados en programación, sin que en ello se establezcan estrategias centradas en el aprendizaje autónomo e independiente que ayude a fortalecer la andragogía de los estudiantes para que puedan adquirir las competencias mínimas para el desarrollo de software, y que no estén sujetas únicamente y exclusivamente a las alertas tempranas evidenciadas a través de la pérdida de una asignatura y realizadas en el seguimiento del éxito académico, sino que se establezcan mecanismos que permitan la validación, autoevaluación, retroalimentación y retrospectiva en los procesos de formación profesional que contribuyan a identificar en los estudiantes de informática, independiente de la calificación que obtengan para aprobar una asignatura, comprobar que efectivamente hayan adquirido las competencias mínimas para desarrollar aplicaciones informáticas.

Otra parte del problema de la falta de profesionales relacionadas con el desarrollo de software, tiene que ver con

las estrategias de enseñanza en el saber hacer impartidas por la academia distintas a las empleadas por otras entidades ajenas y que no forma parte de su razón social, las cuales cuentan con un proceso de formación distinto para que los profesionales adquieran las competencias mínimas que el mercado necesita, y que van desde centros de formación especializado o en línea como los MOOCs hasta con políticas e incentivos económicos (becas) y financieros (emprendimiento) para estudiar programas relacionados con el desarrollo de software como los ofrecidos por el Ministerio de las TICs, a través de las iniciativas de Talento TI (2014) y los cursos de formación específica en desarrollo de software ofrecida mediante la plataforma de apps.co (2012) de MinTic en asocio con Platzi, (2017) o con cursos de entrenamiento ofrecidos por el Clúster de Software y TI (2015) de la Cámara de Comercio de Bogotá. Un mercado de oportunidades que día a día, las instituciones educativas están perdiendo cada día, al no transformar ni fortalecer las mallas curriculares ni los procesos de formación pedagógica-praxeológica.

En el ámbito científico-tecnológico, se pueden encontrar publicaciones en revistas indexadas especializadas como la Revista Inclusión & Desarrollo (2014) de Uniminuto, en el que se menciona la importancia de las TICs en los procesos de aprendizaje educativos aplicados a los programas de secundaria y pregrado como la expresada en sus publicaciones por Cortés-Lozano & Vázquez-Rizo (2014) o Tovar-Cardozo (2015) en donde hacen uso de herramientas interactivas como Scratch (2011) para fortalecer la andragogía y creatividad en los jóvenes adultos para la creación de animaciones interactivas así como la investigación realizada por Serna-Agudelo et al (2018), centrada en los niños y jóvenes para que se apropien en el uso de las TICs, y aunque estas estrategias puedan servir también para la adquisición de competencias mínimas como lo es la lógica de programación para fortalecer el desarrollo de programas de software indiferente del lenguaje de programación que se utilice, no son del todo suficientes y se deben buscar otro tipo de herramientas informáticas que contribuyan a fortalecer el aprendizaje en el desarrollo de software.

Las estrategias y casos de éxito que se puedan encontrar en los distintos entornos académico científico tecnológicos, no siempre pueden funcionar, y en muchos casos no son incluyentes sino excluyentes, en donde para un entorno puedan funcionar pero para otro quizás no, por lo que se debe buscar un nuevo modelo de enseñanza pedagógico-praxeológico que contribuya a la adquisición de competencias para el desarrollo de aplicaciones informáticas independiente de la institución educativa que lo ofrezca o del programa de formación en donde se imparta, contemplando estrategias de inclusión para cualquier tipo de personas sin distinción de edad, género, tipo de discapacidad, población diferencial o adultez con andragogía que sea, pueda fortalecer tanto las competencias del saber hacer como de programar aprendiendo y así contribuir a la formación de profesionales cualificados que suplan la demanda de profesionales que el mercado laboral exige y requiere.

Referencias bibliográficas

- Aguerrondo, I. (2009). Conocimiento complejo y competencias educativas. En: IBE Working papers on Currículum Issues No 8.pp 1-13. http://www.ibe.unesco.org/fileadmin/user_upload/Publications/Working_Papers/knowledge_compet_ibewpci_8.pdf
- Anfurrutia, F. I., Álvarez, A., Larrañaga, M., & López-Gil, J.-M. (2016). Incorporación de robots educativos y entornos de programación visuales en asignaturas de programación, 231-234. Recuperado a partir de <http://hdl.handle.net/10366/131533>
- Apps.co. (2012). Ministerio de la TICs <https://apps.co/acerca/appsco/>
- Bernal, L.A. – Aztlek (2013). Una propuesta para afrontar la crisis de la carrera de ingeniería de sistemas <https://aztlek.org/2013/01/08/una-propuesta-para-afrontar-la-crisis-de-la-carrera-de-ingenieria-de-sistemas/>
- Bernal, L.A. - Aztlek (2015). Programación de computadores, liderazgo y formación de emprendedores en el aula. Escuela de administración de negocios, 86-97. Recuperado a partir de <https://journal.universidadean.edu.co/index.php/Revista/article/view/906>
- Buriticá, J.P. (2015). Sobre el déficit de ingenieros en el sector de software colombiano. Recuperado de <https://medium.com/colombia-dev/el-deficit-de-programadores-en-el-sector-de-software-colombiano-b2df0f24b435>
- Carlisle, M., Wilson, F., Humphries, J. & Hadfeld, S. (2005). RAPTOR: A Visual Programming Environment ±or Teaching Algorithmic Problem Solving. USA: Department of Computer Science, United States Air Force Academia. Recuperado de <https://raptor.martincarlisle.com/>
- Clúster de Software y TI (2015). Cámara de Comercio <https://www.ccb.org.co/Clusters/Cluster-de-Software-y-TI/Sobre-el-Cluster/Quienes-somos> Recuperado de <https://www.youtube.com/watch?v=9fOeHXDwRtA>
- Code (2013). Anybody can learn. [Plataforma de cursos online-MOOC] Recuperado de <https://code.org>
- Colby, A., Sheppard, S., Macatangay, K., & Sullovan, W.M.(Autores) & Martínez J.J. (Traductor) (2017) - Educar Ingenieros -Diseño curricular para el futuro de la profesión. Ediciones de la U. ISBN: 9789587626995. Recuperado a partir de <https://edicionesdelau.com/producto/educar-ingenieros-diseno-curricular-para-el-futuro-de-la-profesion/>
- Colombia.Dev (2016). Data Set – Resultados de la encuesta realizada por Buriticá y Colombia.Dev <https://github.com/colombia-dev/data/blob/master/salaries/2016/raw.csv>
- Cortés-Lozano, A.X. & Vázquez-Rizo, F.E. (2014). El uso de la Tecnología en ambientes educativos. Análisis de su inclusión un contexto formativo contemporáneo (The use of technology in educational environments. Analysis of their inclusion in a formative contemporary context) Revista Inclusión & Desarrollo Vol. 1. Núm. 1. Enero-Junio. Pags. 35-44. ISSN Impreso: 2389-7341. ISSN En línea: 2590-7700 <http://revistas.uniminuto.edu/index.php/IYD/article/view/1269>
- Dans, E. (2007). Programadores: cuando falla la base. España. Recuperado de Altia News <https://www.enriquedans.com/2007/10/continando-con-el-tema-de-la-falta-de-programadores.html>
- Díaz Chaparro, L.C., Torres Moreno, M. E., Hurtado Rojas, J. H., Chavarro Flórez, G. A. & Ruiz García, E. E. (2013). Software tangible: metáforas, representaciones visuales y actividades de apoyo didáctico para la enseñanza en construcción de software. Revista lasallista de investigación, vol 10, N° 2, 84-101. Recuperado a partir de <http://repository.lasallista.edu.co:8080/ojs/index.php/rldi/article/view/510>
- El Clarin. (2017) Faltan programadores. Argentina. [Artículo de noticia] https://www.clarin.com/sociedad/falta-programadores-quedan-mil-puestos-vacantes-pais_0_BIFHsMIB-.html
- El Confidencial (2017). España busca programadores: hay 300.000 (pero faltan muchos más). España. https://www.elconfidencial.com/tecnologia/2017-07-21/programadores-desarrolladores-trabajo-espana-paro_1417198/
- El Espectador (2015) A cambiar la ingeniería de sistemas. Entrevista a David Luna. Ministro de las TICs. Colombia. [Artículo de noticia] <https://www.elespectador.com/noticias/economia/cambiar-ingenieria-de-sistemas-articulo-569035>
- El Tiempo.(2015). En el sector TI hay más empleo que profesionales. Colombia. [Artículo de noticia] <http://www.eltiempo.com/archivo/documento/CMS-16462889>

- Expansión (2016), Millennials y falta de programadores. Nuevos retos del e-commerce. México. <https://expansion.mx/tecnologia/2016/10/27/millennials-y-falta-de-programadores-nuevos-retos-del-e-commerce>
- Ferreiro Gravié, R. (2017) ¿Cómo ser maestro investigador? El método Javi - Vademécum. Editorial Uniminuto. ISBN: 978-958-763-220-0. Recuperado a partir de <https://ebooks7-24.ezproxy.uniminuto.edu/book.aspx?i=7330> // <https://www.ebooks7-24.com:443/?il=7330>
- Ferreiro, R. & Cummings, N. (2014). MeProB Determinación y formulación del Problema científico para una Tesis aplicada en Educación. Material en proceso de validación, Escuela de Educación Abraham S. Fischler, Nova Southeastern University, Florida, USA <https://www.unioviado.es/reunido/index.php/RIAICES/article/view/10813>
- Free Software Foundation. (1984) General Public License. GPL <https://www.gnu.org/licenses/licenses.es.html>
- Fuentes-Rosado, J. I. & Moo-Medina, M. (2017). Dificultades de aprender a programar. Artículo publicado en la revista de la Asociación Colombiana de Facultades de Ingeniería (Acofi) <https://www.educacioneningeneria.org/index.php/edi/article/viewFile/728/331>
- Gestión (2015). El 50% de empresas peruanas demoran hasta 45 días para contratar profesionales especializados. Perú [Artículo de noticia] <https://gestion.pe/tendencias/management-empleo/50-empresas-peruanas-demoran-45-dias-contratar-profesionales-especializados-107738>
- GitHub(2008) Plataforma de alojamiento de proyectos de software y versionamiento. <https://github.com>
- Google Cloud Source Repositories. (2015). Repositorio de código fuente para aplicaciones albergados en los servidores de Google <https://cloud.google.com/source-repositories/>
- IDX. Bonilla, J. (2017) ¿Qué podemos hacer ante la escasez de programadores en Colombia?. Colombia <https://idx.com.co/qu%C3%A9-podemos-hacer-ante-la-escasez-de-programadores-en-colombia-fd24239deccd>
- It User.(2016). Crece la escasez de desarrolladores. España [Artículo de noticia] <http://www.ituser.es/cloud/2016/11/crece-la-escasez-de-desarrolladores-cloud>
- Kniberg, H. (2008) Scrum y XP desde las trincheras. Licenciado bajo Creative Commons, y suministrado por InfoQ. <http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf>
- La Nación.(2017). Los programadores, el tesoro que buscan y no encuentran las empresas argentinas. Argentina. [Artículo de noticia] <https://www.lanacion.com.ar/2017760-los-programadores-el-tesoro-que-buscan-y-no-encuentran-las-empresas>
- La República (2010). Cuervo, F. Ingenierías: mucha demanda y poca motivación para estudiarla. [Artículo de noticia]. Recuperado del portal del Ministerio de Educación <https://www.mineducacion.gov.co/observatorio/1722/article-243384.html>
- La República.(2016). Escasez de profesionales beneficia a desarrolladores Web. Colombia. [Artículo de noticia] https://www.larepublica.net/noticia/escasez_de_profesionales_beneficia_a_desarrolladores_web
- Marquina, J. (2013) ¿Qué pasa en Internet en 60 segundos?... ¿y en 24 horas?... Impresionante [Artículo de Internet] <https://www.julianmarquina.es/que-pasa-en-internet-en-60-segundos-y-en-24-horas-impresionante/>
- Martínez Figueroa, L. F. & Molina Sierra, H. O. (2017). Juego de enseñanza de programación para niños, 1-83. Recuperado a partir de <http://eduteka.icesi.edu.co/gp/upload/martinezluisa2017.pdf>
- Mi Ambiente. Machorro, J.C. (2018) Urgente que México subsane la falta de programadores digitales. México. <http://www.miambiente.com.mx/general/urgente-que-mexico-subsane-la-falta-de-programadores-digitales/>
- MinTic-Platzi (2017). MinTIC y Platzi abren cursos virtuales gratuitos para fortalecer habilidades TIC de los emprendedores colombianos <https://www.mintic.gov.co/portal/604/w3-article-61415.html>
- Pi, R. (2012). Raspberry Pi. [Software de computación]. Recuperado de <https://www.raspberrypi.org/>
- Przybilla, D. A. (2016). Análisis : Encuesta Salarios Desarrolladores ColombiaDev 2016 <https://medium.com/colombia-dev/an%C3%A1lisis-encuesta-salarios-desarrolladores-colombiadev-2016-9969a621ec39>

Deficiencia de adquisición de competencias mínimas en estudiantes de desarrollo de software: hacia un nuevo modelo de enseñanza pedagógico praxeológico

Radiomitre (2017). Cada año quedan 5 mil puestos vacantes para programadores en Argentina. Argentina. [Artículo de noticia] <https://radiomitre.cienradios.com/cada-ano-quedan-5-mil-puestos-vacantes-de-programadores-en-argentina/>

Raffini, J.P. (1998) 150 formas de incrementar la motivación en la clase. Editorial Troquel. ISBN: 950-16-3086-2. Recuperado a partir de https://openlibrary.org/books/OL12961694M/150_Formas_de_Incrementar_La_Motivacion_En_Clase

Redis (1991). Red de Programas de Ingeniería de Sistemas y Afines <http://www.acofi.edu.co/redis/>

Resnick, M. (2013). Aprender a programar, programar para aprender. Universidad Icesi. <http://eduteka.icesi.edu.co/articulos/codetolearn>

Revista Dinero.(2008) Escasez de profesionales en informática. Colombia [Artículo de Noticia] <https://www.dinero.com/negocios/articulo/hay-deficit-profesionales-informatica/60456>

Revista Dinero.(2008) No hay ingenieros de sistemas. Colombia [Artículo de Noticia] <https://www.dinero.com/economia/articulo/no-ingenieros-sistemas/68220>

Revista Inclusión & Desarrollo (2014).

Revista Semana (2012). ¿Y dónde están los ingenieros?. Colombia [Artículo de Noticia]. <https://www.semana.com/tecnologia/articulo/y-donde-estan-los-ingenieros/402945-3>

Revista Semana (2013). En TIC hay un cantidad de oportunidades para los jóvenes colombianos [Artículo de noticia]]<https://www.semana.com/nacion/articulo/en-tic-cantidad-oportunidades-para-jovenes-colombianos/347988-3>

Revista Semana (2016). Las carreras tecnológicas que usted podría estudiar gratis. Colombia. [Artículo de Noticia] <https://www.semana.com/tecnologia/articulo/estudiar-gratis-programas-universitarios-de-sistemas-informatica-electronica/506395>

Romero-Díaz, X., Rodríguez-Castro, L.S. & Cortés-Trujillo, J. (2014). Tic en la vida escolar de estudiantes que presentan discapacidad visual (ICT in school life of students that present visually impaired). Vol. 2 Núm. 2 (2015): Julio - Diciembre. Pags.88-95. ISSN Impreso: 2389-7341. ISSN En línea: 2590-7700 <http://biblioteca.uniminuto.edu/ojs/index.php/IYD/article/view/1073>

Rodríguez, J. (2015). Integridad académica en la docencia universitaria actual con énfasis en el plagio de código fuente: modelo, propuesta de intervención y herramientas. Trabajo de grado. Universidad de las palmas de gran Canaria, 1-276. Recuperado a partir de https://acceda.ulpgc.es:8443/bitstream/10553/17412/4/0724522_00000_0000.pdf

Scratch. (2011). Grupo Lifelong Kindergarten del MIT Media Lab. [Software de computación]. Recuperado de <http://scratch.mit.edu/>

Serna-Agudelo, B; Recalde-España, E; Adolfo-Beltrán, G; Cañón-Recalde, C (2018). El Scratch como estrategia didáctica para desarrollar la exploración del medio en la educación inicial Fase I y II. (The Scratch as a didactic strategy to develop the exploration of the environment in early education-Phase I and II-). Revista Inclusión y Desarrollo, 5 (2) 2018, 19-33 <http://biblioteca.uniminuto.edu/ojs/index.php/IYD/article/view/1604>

Sommerville, Ian (2011). Ingeniería de Software. Editorial Pearson Education. Novena Edición. ISBN: 9786073206044 <https://www.pearsoneducacion.net/ecuador/tienda-online/ingenieria-software-sommerville-9ed-ebook1>

SourceForge(2013) Plataforma de colaboración de proyectos de software. <https://sourceforge.net/>

Team Foundation Server (2018). Plataforma de administración de código fuente para soluciones de Microsoft <https://visualstudio.microsoft.com/es/tfs/>

Talento TI (2014). Ministerio de las TICs <https://www.talentoti.gov.co/635/w3-article-14158.html>

TIC BEAT.(2016). El sector TIC sufre por la falta de ingenieros de software <http://www.ticbeat.com/educacion/branded-el-sector-tic-sufre-por-la-falta-de-ingenieros-de-software-cualificados/>

Tobón, S. (2007). El enfoque complejo de las competencias y el diseño curricular por ciclos propedéuticos. Acción Pedagógica 16: 14-28 <https://dialnet.unirioja.es/descarga/articulo/2968540.pdf>

Toledo, J. A. J., Ordoñez, C. A. C., Alegría, J. A. H., & Yépez, W. L. P. (2015). Estrategia colaborativa en entornos tridimensionales como estrategia didáctica de aprendizaje de estructuras iterativas en programación computacional. InvestigiumIRE, 6(2), 80-92.

Recuperado a partir de <http://investigumire.iucesmag.edu.co/index.php/ire/article/view/112>

Torrenegra, A. (2015). Desarrolladores de software ganan en promedio \$6 millones. Encuesta de Bunny Inc. Recuperado de Portafolio, Colombia <http://www.portafolio.co/negocios/empresas/desarrolladores-software-ganan-promedio-millones-31318> y de Pulzo, Colombia <https://pulsosocial.com/2015/02/17/infografia-cuanto-ganan-realmente-los-desarrolladores-colombianos-de-software/>

Tovar-Cardozo, M.A. (2015). Factores asociados al uso de las TIC en estudiantes de programas de pregrado, modalidad presencial, en universidades de Neiva, Colombia (Factors Associated to the Use of ICT in Students of University Programs) Vol. 2 Núm. 1 : Enero - Junio. ISSN Impreso: 2389-7341. ISSN En línea: 2590-7700 <http://revistas.uniminuto.edu/index.php/IYD/article/view/1059>

Trejos Buriticá, O. I. (2014). Relaciones de aprendizaje significativo entre dos paradigmas de programación a partir de dos lenguajes de programación. *Tecnura*, 18(41), 91-102. Recuperado a partir de <http://www.scielo.org.co/pdf/tecn/v18n41/v18n41a08.pdf>

Ulloa, G. (2008). ¿Qué pasa con la Ingeniería en Colombia? Universidad de Antioquia <http://aprendeonline.udea.edu.co/revistas/index.php/ingeso/article/viewFile/7303/6742>

Uniminuto. (2018) Proyecto Educativo Institucional – PEI. Colombia. <https://www.uniminuto.edu/proyecto-educativo-institucional-pei>

Vargas, C., Reyes, C. J., Massé Palermo, M. L., Ramírez Morales, J., Espinoza, C. N., & Tapia, C. R. (2015). Desarrollo de un módulo para Moodle como soporte para el aprendizaje colaborativo de la programación en el nivel universitario inicial. Paper presented at the X Congreso sobre Tecnología en Educación & Educación en Tecnología (TE & ET) (Corrientes, 2015), 66-74. Recuperado a partir de <http://sedici.unlp.edu.ar/handle/10915/49123>