

**RECURSOS DEL SOFTWARE LIBRE PARA FAVORECER
PROCESOS INNOVADORES DE ENSEÑANZA – APRENDIZAJE
EN CURSOS DE PROGRAMACIÓN DE COMPUTADORES**

***RESOURCES FREE SOFTWARE PROCESS TO PROMOTE INNOVATIVE
TEACHING - LEARNING COURSES COMPUTER PROGRAMMING***

Miguel Ángel Tovar Cardozo

José Daniel Cabrera Cruz (Director)

*Universidad Autónoma de Bucaramanga
Convenio con la Universidad de Oberta de Cataluña España*

Resumen

Asumiendo la importancia de la innovación para los estudiantes de las diversas ingenierías, se pretende impulsar el acogimiento de las herramientas que ofrece el Software Libre en los procesos de enseñanza aprendizaje. Esto se proyecta desde el curso Programación de Computadores al que acuden no solo estudiantes de Sistemas, sino también todos los que aspiran a obtener su título de Ingeniero. En las ingenierías diferentes a Sistemas, se hace aún más perentorio infundir una concepción diferente, innovadora, de lo que aborda la programación, pues la práctica del docente e investigador, le ha permitido percibir que a muchos estudiantes se les dificulta avanzar en los contenidos curriculares que ofrece el *pensum* con respecto al curso.

Palabras clave: software libre, Innovación, programación de computadores, enseñanza-aprendizaje, ingenierías.

Abstract

Assuming the importance of innovation for students of various engineering, is intended to promote the placement of the tools offered by Free Software in teaching and learning processes. This extends from the Computer Programming course that students come not only systems, but also all those who aspire to obtain his degree in Engineering. In the various engineering to systems, it becomes even more imperative to instill a different conception, innovative, what addresses the programming, for the practice of teaching and research, has enabled him to perceive that many students find it difficult to advance the curriculum offering on the assignment.

Keywords: free software, Innovation, computer programming, teaching-learning engineering.

Introducción: El curso de Programación de Computadores, una oportunidad para aprender a innovar dentro de las Ingenierías

Acudir a las TIC con la firme idea de fortalecer el proceso de educativo, se viene convirtiendo en una oportunidad para que el curso de Programación de Computadores de la facultad de Ingeniería en la Universidad Antonio Nariño de Neiva, replantee las prácticas pedagógicas; por supuesto, sin alejarse de los saberes, los aprendizajes, los valores y los compromisos que impone su Plan de Estudios. Por ello, vale la pena ahondar en los niveles de acercamiento a las herramientas que ofrece el Software Libre, y el aprovechamiento de estas tanto en las clases, como fuera de ellas.

En medio de los vertiginosos cambios a los que nos vemos abocados en la últimas dos décadas, propiciar un horizonte de mirada a los actuales estudiantes y a otros docentes que

participan cotidianamente en la materia Programación de Computadores, amplía el ángulo de visión y agudiza los sentidos, de tal forma que se encuentra para ellos también, un lugar propio dentro del mundo globalizado que toca a las puertas de la Universidad sin duda alguna.

Se aclara, entonces, que la idea es dedicar un espacio académico para escudriñar el tema de la enseñanza aprendizaje del curso Programación de Computadores, enfocándolo en la innovación, y asumiendo que como docente de esta asignatura, la creatividad, el pensamiento propositivo y el razonamiento lógico son habilidades esenciales en las diversas Ingenierías y pueden desarrollarse con mayor amplitud si el curso se apoya en herramientas del Software Libre.

Como docente de Programación de Computadores, reconozco que el profesor de esta asignatura constantemente enfrenta situaciones relacionadas con las dificultades que tienen los estudiantes para identificar un problema, para definirlo u ordenarlo, y luego seguir un procedimiento para la solución del mismo. Si bien estas deficiencias afectan el aprendizaje de otros cursos, es más apremiante en el caso de esta materia, en la cual estas destrezas son fundamentales. Los estudiantes de Carreras diferentes a Ingeniería de Sistemas, no entienden o no le hallan el sentido que tiene dicha materia en su Ingeniería escogida. Esto hace que no se sientan motivados o interesados en esos cursos. Se tiende a limitar el ofrecimiento de materiales, recursos y actividades a lo que se hace en el aula de clase con acompañamiento del docente. No se brinda mayor atención a lo que el estudiante hace por fuera del aula (es decir, su trabajo independiente). El docente considera que el estudiante no siempre cuenta con recursos dispuestos para sacar mejor provecho de su tiempo de trabajo independiente. Adicionalmente, no resulta sencillo para el docente percibir la posibilidad de que el alumno reciba realimentación oportuna y seguimiento de lo que está haciendo por fuera de la clase.

Como si fuera poco, es común apreciar situaciones en las que como docentes, creemos posible enseñar la programación de computadores sin utilizar equipos computarizados; a los estudiantes de hoy en día, que tienen interés y expectativas de utilizar elementos informáticos y digitales, eso les resulta desmotivante, especialmente cuando se trata de un curso vinculado esencialmente con la tecnología.

Una razón más para que en ocasiones los estudiantes de los cursos de Programación de Computadores no descubran la aplicación o uso práctico de lo que están aprendiendo, consiste en que la mayoría de las evaluaciones que se realizan dentro de los cursos, se reducen a pruebas de escritorio que no se vinculan con la vida cotidiana del estudiante, como por ejemplo: el hecho de que los programas que se construyen no resuelven problemas o necesidades cotidianas que puedan ejecutarse desde sus móviles o portátiles para resolver algún problema o necesidad clara para ellos. Se observa, además, que los métodos que usan los estudiantes para registrar u organizar los recursos y contenidos que se brindan en clase no son muy eficientes; de modo que, con frecuencia, no pueden disponer de éstos cuando los necesitan para su estudio en futuras clases y en su tiempo de trabajo independiente.

Se subraya, en estos términos, que en los cursos de Programación de Computadores se desaprovechan recursos e innovaciones tecnológicas educativas existentes, tanto aquellas de carácter general (LMS, CMS, por ejemplo), como aquellas específicamente orientadas a la educación en programación. Respecto de estas últimas, se vienen utilizando herramientas como DFD, PSINT y Zinjai, que si bien resultan un apoyo importante, no facilitan su uso desde Web con los dispositivos móviles de los estudiantes, y a veces no fortalecen el aprendizaje y el ideal seguimiento. Estas herramientas no ofrecen muchas alternativas diferentes y quizás más llamativas para programar, por ejemplo: compilación en línea, formas

gráficas o visuales y variadas de presentación de los programas, la posibilidad de centralizar y disponer los recursos que van encontrando o generando ellos mismos. Existen hoy en día, en cambio, recursos más novedosos que pueden motivar y favorecer la clase. Aprovechar el Software Libre puede ser esa opción para descartar debilidades como las mencionadas.

Tradicionalmente en el aula se enseña a programar con lenguajes y herramientas privativas (no libres) que se limitan a lo que ofrece la casa desarrolladora de estas herramientas. Por desconocimiento o cultura de los docentes dentro del curso de Programación de Computadores, no proponemos recursos de software libre y si se hace, no enfatizamos en los beneficios y en el trabajo en comunidad que es uno de las características fundamentales del mismo.

De esta manera, no se puede seguir pretendiendo que con un plan de estudios que ya cuenta con relativa trayectoria, se logre llegar a un grupo de estudiantes que cursan generalmente su primer año de pregrado (estamos hablando de jóvenes con edades que oscilan entre los 18 y 20 años) y a quienes -desde luego- si algo se les facilita, es precisamente la utilización de dispositivos electrónicos, sin antes como profesor proponernos enfocar el conocimiento curricular, a un campo donde tenga real aplicabilidad y sentido para ellos.

A partir de diversos estudios que se han realizado respecto al tema, se acepta que para los nacidos en esta era digital, y que se les considera “usuarios permanentes de las tecnologías con una habilidad consumada” (Benito, García, Portillo, & Romo, 2007, pág. 23) el currículo debe ser otro: Uno que de alguna manera tenga claro sus necesidades de comunicación, información y, desde luego, de formación.

Los expertos plantean que estos nuevos usuarios enfocan el aprendizaje de nuevas formas: absorben rápidamente la información multimedia de imágenes y videos, igual o

mejor que si fuera texto; consumen datos simultáneamente de múltiples fuentes; esperan respuestas instantáneas; permanecen comunicados permanentemente y crean también sus propios contenidos. Funcionan mejor trabajando en red, puesto que el nativo digital en su niñez ha construido sus conceptos de espacio, tiempo, número, causalidad, identidad, memoria y mente a partir, precisamente, de los objetos digitales que le rodean, pertenecientes a un entorno altamente tecnificado. Hay quien sostiene que el crecimiento en este entorno tecnológico puede haber influido en la evolución del cerebro de aquellos individuos. (Benito, García, Portillo, & Romo, 2007)

Esta necesidad le demanda al docente determinar qué dirección debería tomar la innovación en las ingenierías, así como en todos los niveles formativos, para adecuarse a las características de los nativos digitales. No en vano se ha planteado que los efectos que producen las nuevas tecnologías demandan estudios con una perspectiva pedagógica, debido a que la enseñanza no se origina en la renovación del equipo, sino más bien por la reestructuración de los enfoques pedagógicos. (Maggio, 2002)

En esos términos, se puede considerar que el Software Libre, puede convertirse en la Universidad, en un ingrediente valioso para las Carreras de Ingenierías. El conjunto de las propiedades que este brinda, puede llegar a ser la alternativa de solución del problema, y con todo su ambiente creador, puede contrarrestar las dificultades que los estudiantes manifiestan dentro de la asignatura para aprender a programar, a tal punto que ellos puedan abrir y estudiar un programa, puedan hacerle cambios y aprender viendo lo que pasa con cada cambio; sin impedimentos o restricciones, ellos lograrán mezclarse con otras rutinas o programas libres.

En Colombia, en todos los sectores, hay una preocupación fundamental: es preciso atraer y formar talento que hagan parte de la era de la información y el conocimiento, es innegable que actualmente no se logren llenar todas las vacantes mundiales exigiendo personal con conocimientos en lenguaje de programación; se hace vital, por lo tanto, en estos momentos, incentivar este aprendizaje en las actuales generaciones. Sin embargo, se recuerda que, aunque para algunos Ingenieros la programación es una gran ayuda a la hora de resolver algunos problemas tecnológicos, no necesariamente ellos la estudian para ser programadores.

La descripción del caso que se ha hecho en los párrafos anteriores, así como el método acogido, pero sobre todo sus hallazgos, son los motivos que impulsan el presente artículo. En definitiva, el propósito de este texto es determinar cómo se pueden aprovechar los recursos que brinda el software no privativo para potenciar los procesos de enseñanza aprendizaje e innovación en el marco de los cursos de la Programación de Computadores en programas de Ingeniería. El enfoque se ha fundamentado esencialmente en una perspectiva multivariada que integra, asimismo, la influencia de los factores personales y contextuales.

Método: Aproximaciones a una perspectiva multivariada

El trabajo de investigación titulado precisamente “Recursos del Software Libre para favorecer la enseñanza, aprendizaje e innovación en cursos de programación de computadores en ingenierías”, lo he presentado con algunos elementos conceptuales que lo fortalecen sin lugar a dudas. Y el proceso investigativo que aquí se reporta, se concretó en tres etapas desarrolladas de modo paralelo: el análisis documental, la instrumentación de una entrevista diagnóstica, y la selección de herramientas del software libre para ser llevadas a

dos grupos de estudiantes seleccionados bajo condición experimental¹, precisando que existió un tercer grupo, al que no se le hizo ninguna intervención.

La primera fase consistió en la síntesis de los documentos publicados siguiendo como criterios las temáticas (programación de computadores, innovación, Software Libre, educación, ingenierías) y su tipología (libros, investigaciones, ensayos, postulados); la tarea permitió identificar a los diferentes referentes con los cuales las bases se afirmaron, hasta el punto de construir una radiografía de la investigación propuesta.

Pues de esta manera, y siguiendo las pautas de Sabino (2002) cuando asegura que durante la investigación cualitativa la naturaleza del caso o fenómeno es el que orienta al investigador para que indague con mayor detenimiento ciertos aspectos del mismo, sostuve un estudio exhaustivo del fenómeno que rodea el tema del Software Libre en el proceso de enseñanza aprendizaje del curso Programación de Computadores ofrecido en los diversos pregrados de la facultad de Ingeniería de la Universidad Antonio Nariño, sede Neiva, lo cual permitió ampliar la mirada sobre los procesos implícitos en el caso.

Mientras esta etapa avanzaba, el diagnóstico de posibles problemas que se presentan en la enseñanza aprendizaje de dicha materia con respecto a la innovación, se fue profundizando. Para ello, asumí la tarea de acercarme a cinco estudiantes que en semestres anteriores ya hubieran tomado la asignatura; así como también a cinco docentes de Programación de Computadores, de los cuales uno de ellos no ofrece el curso en este

¹ Los sujetos pertenecientes a la condición experimental, pertenecen a la facultad de Ingeniería de la Universidad Antonio Nariño sede Buganviles-Neiva, y lo conforman 23 estudiantes de un grupo o aula [22 varones, 1 mujer], y 22 del otro curso [21 hombres, 2 mujeres]. El tercer grupo (grupo control) al que no se le haría ninguna intervención, lo forman 27 alumnos [13 hombres, 14 mujeres]. La edad promedio de estos estudiantes participantes, es de 19 años.

momento, pero lo dirigió hasta hace poco tiempo. A los entrevistados le compartí un cuestionario a manera de guía, para apreciar más de cerca la situación problema.

Las respuestas ratificaron, en gran medida, la auto-reflexión: mi propia percepción también fue un ingrediente más que le sumé a esta radiografía, y que no pudo pasar inadvertida ante la lluvia de situaciones descritas en el estado del arte. De todas formas, el diagnóstico se convirtió en el primer paso para descubrir con claridad cuáles eran las principales dificultades e intereses de los estudiantes, así como los requerimientos de los docentes para desarrollar un proceso de enseñanza/aprendizaje apropiado, en un marco de innovación.

Las oportunidades que brindaran dichas dificultades e intereses, mostraban seguramente las posibilidades de uso del Software Libre, y por ello la tercera etapa, paralela en este trasegar metodológico, lo enmarcó la pesquisa de sugerencias que se logra en la Web y los mismos antecedentes estudiados, en cuanto a recursos del software no privativo para favorecer la enseñanza, el aprendizaje y la innovación en el curso de Programación.

El constructivismo le otorgó a esta investigación de corte cualitativo, los énfasis principales que lo caracterizan: (a) El reconocimiento de que el investigador necesita encuadrar en los estudios, los puntos de vista de los participantes; (b) La necesidad de inquirir cuestiones abiertas; (c) En vista de que el contexto cultural es fundamental, la recolección de los datos se asume, es levantada en los espacios donde las personas ejecutan sus actividades diarias; (d) La investigación debe servir para optimizar el desenvolvimiento de los individuos; y (e) la preocupación no es solamente por las variables “exactas”, puesto que lo que se estudia son conceptos, cuya particularidad no solamente se conquista a través de cifras y mediciones (Hernández R. F., 2006).

Proceso investigativos: Tareas y síntesis de las mismas

Durante el proceso investigativo asumí el hecho de elaborar síntesis luego de cierto número de tareas, reconociéndolas como acciones “orientadas a facilitar la mirada reflexiva del investigador”, como diría Salgado (2007, pág. 33) en su documento sobre investigación cualitativa. Con ellas, el proceso llegó a ser un buen ejercicio metodológico.

En cuanto a las síntesis de las que hago mención, y que se iban convirtiendo en antesala a los anhelados resultados, los iré presentando en cuadros y en figuras. Iniciaré por un cuadro creado a partir de los hallazgos del estudio, siguiendo las etapas comprendidas en el trabajo, en relación con la revisión -problemas en aprendizaje e innovación en cursos de Programación de Computadores (Cuadro 1). Posteriormente trazaré otros relacionados con la continuidad de la agenda investigativa, para finalizar definiendo en qué consistió la implementación parcial y mencionando las principales conclusiones del estudio.

Mientras que el Cuadro 1 sintetiza con exclusividad, lo más vigente y significativo de la pesquisa realizada, me propuse identificar las oportunidades de uso del Software Libre en Colombia y en otros lugares, para terminar sintetizando las principales herramientas y los casos de éxito de las mismas, tal como se aprecia en los Cuadros 2 y 3. Allí expongo las herramientas más destacadas. Su criterio de selección para ser llevadas a esta síntesis, consistió en su cercanía con el interés investigativo, según la revisión bibliográfica.

Cuadro 1. Síntesis revisión -problemas en aprendizaje e innovación en cursos de Programación de Computadores

Autor	Año	Caso representativo (Institución)	Síntesis
Malaver, Rey, & Rodríguez	2011 (Colombia)	Enseñanza de la programación de computadores en Colombia Universidad Politécnica Gran Colombiano	<ul style="list-style-type: none"> • A pesar de la inclusión de Pensamiento Algorítmico, los estudiantes llegan a la asignatura Programación de Computadores con deficiencias conceptuales en términos de la aplicación de conceptos matemáticos adquiridos previamente en su educación media. • Para los estudiantes resulta complejo entender cómo una serie estática de instrucciones se comporta en tiempo de ejecución; particularmente las sentencias condicionales y los ciclos resultan confusos en términos de su interpretación en el tiempo. Las pruebas de escritorio ayudan mucho en este sentido, pero en muchos de los casos no son prácticas en términos de tiempo. • La apropiación del conjunto de reglas inherentes a un lenguaje de alto nivel añade otro nivel de dificultad al aprendizaje de programación.
Villalobos Salcedo, Jorge Alberto	2009 (Colombia)	Universidad de los Andes	A los problemas de motivación de los estudiantes se une la falta de un estudio a fondo de las habilidades que deben adquirir, reduciendo muchas veces los cursos a un recorrido de estructuras sintácticas de un lenguaje de programación. El estudio corroboró que es claro que los cursos de programación son fundamentales en la formación de ingenieros.
Hermanos Arocha Cisneros	2008 (Venezuela)	Instituto Superior Politécnico "José Antonio Echeverría"	El alumno asume un rol poco activo y mínimamente participativo; su motivación es mínima y se demuestra desinteresado en ser un constructor de su aprendizaje.
Stallman, Richard	2013	Organización FSF	Frente a la apremiante condición de innovar en la enseñanza, se halla la necesidad de reconocer que los jóvenes estudiantes de hoy están más prestos a programar, pues las computadoras hacen parte de su cotidianidad. Sin embargo, con el software privativo esta información es secreta, por lo tanto los profesores no tienen forma de dejarla a disposición de los alumnos.

Fuente: Elaboración propia

Me atrevo a decir que en cuanto a las LMS Chamilo y Claroline, no se especifica en línea casos precisos de aprovechamiento en los procesos de enseñanza de cursos de Programación, tan solo opiniones o comentarios muy breves en foros, donde son ubicadas como potenciales herramientas para aprender teoría y evaluarla. En cuanto a Moodle, Dokeos y Edx Platform, se perciben experiencias procesos formativos alrededor de la programación, pero sobre todo en cursos masivos en línea. Sin embargo, Chamilo ha de ser mi favorita, y más adelante explicaré el porqué.

Cuadro 2. Herramientas innovadoras y sus casos de éxito

Herramienta	Casos de éxito
Waterbear	En línea se encuentra exclusivamente un texto del 3 de mayo de 2011, día en el cual Dethe Elza, creadora de esta herramienta, la presenta en JSConf en Portland, OR. Allí se asegura que Waterbear trae los conceptos de lenguajes de programación. Ver más en http://readwrite.com/2011/05/03/waterbear-is-like-scratch-but#awesm=~os1TH05D76W15l
Scratch	Entre los diferentes casos de éxito de esta herramienta que en línea se hallan, se destaca la relatada por un docente que asegura que 300 estudiantes mexicanos trabajaron durante cuatro meses con Scratch, en quienes se generó un ambiente de aprendizaje que los motivó a desarrollar habilidades de razonamiento cognitivo y metacognitivo. El docente que relata la experiencia, recuerda que fue en un taller con Telmex años atrás cuando lo llegó a descubrir; llevaba casi dos décadas estaba empleando LogoWriter y Lego Mindstorms, pero al encontrarse con Scratch, reconoce en ella todas las ventajas para fortalecer el aprendizaje de la programación. http://scratched.media.mit.edu/stories/using-scratch-develop-reasoning-skills-mexico mas historias en http://scratched.media.mit.edu/stories
Blockly	La página 'Niños Construyendo' (http://constructingkids.com/2013/05/15/vpl/), en una entrada del 15 de mayo de 2013, se habla de esta herramienta para hacer referencia a lenguajes de programación visuales.
App Inventor	En línea se puede leer un testimonio con respecto a un proyecto de aula dispuesto con el fin de enseñar a programar en un instituto de secundaria. Se logró allí que Los alumnos crearan apps con esta herramienta. Eso exactamente fue en la asignatura Iniciación Profesional a la Electrónica en la Institución Educativa Palomas Vallecas de la capital española. La propuesta consistió en que cada alumno diseñara en lenguaje de programación, al menos una aplicación para el móvil. La ponente, Angeles Aragus, demuestra con este proyecto que la metodología es lo que importa, pues la creación de una app para el móvil es un pretexto perfecto para aprender a programar, a escribir código. Más en: http://www.educacontic.es/blog/inventores-de-apps
Rebeca a través del espejo	Dentro de las Actas de las I Jornadas en Innovación y TIC Educativas –JITICE 2010, existe un interesante texto de Estefanía Martín Barroso, Manuel Rubio Sánchez y Jaime Urquiza Fuentes, quienes describen el surgimiento de "Rebeca a través del espejo", basado en Alice, como aquella idea de la Universidad Carnegie Mellon, dispuesta para apoyar y motivar la enseñanza de la programación orientada a objetos. Los ponentes de esta jornada describen que durante el desarrollo de Rebeca se llevaron a cabo diversos talleres pilotos en Juvenalia y el fesTICval, al igual que concursos de animación en la Comunidad de Madrid, subrayando lo interesante de la experiencia y sus positivos resultados. Más en: http://www.etsii.urjc.es/investigacion/archivos/BoletínETSII-2010-001-JITICE-2010.pdf

Fuente: Elaboración propia

Cuadro 3. Herramientas LMS y casos de éxito

Herramienta LMS	Casos de éxito
Dokeos	La Universidad Santiago de Cali (usc.edu.co) dispone de un portal soportado en Dokeos para todos sus programas incluido el de Ingeniería de Sistemas. La página es http://virtual.usc.edu.co/
Moodle	Apps.co un programa del gobierno colombiano para promover el emprendimiento en TICs, hace uso de Moodle en la etapa llamada <i>bootcamps</i> https://apps.co/inscripciones/fase/bootcamps/ , en ella enseñan diferentes tecnologías para programar en diferentes entornos. La dirección donde se hace eso es http://bootcamps.apps.co/
Edx platform	Esta plataforma es usada por la organización Edx www.edx.org , en la que se ofrecen cursos masivos en línea con el apoyo de prestigiosas universidades. Se pueden encontrar diferentes cursos de programación como por ejemplo: https://www.edx.org/course/mitx/mitx-6-00x-introduction-computer-science-586

Fuente: Elaboración propia

No obstante, se hacía necesario cumplir con una siguiente tarea, consistente en definir los criterios de comparación de herramientas y recursos del Software Libre para su uso en educación para Programación de Computadores. Las fuentes de los criterios seleccionados giraron en torno al diagnóstico realizado, y a criterios de este, junto a lo hallado sobre ingeniería de software y software educativo. De esta manera, se definieron como criterios a tener en cuenta, los siguientes: La facilidad de uso (instalación e interfaz de usuario), las características del producto (tipo de licencia, documentación, que se encuentre en español preferiblemente), el objetivo (ámbito y posible descripción), y las características técnicas (tipo, repositorio, número de personas involucradas en su desarrollo).

En este sentido, surgen los Cuadros 4 y 5, donde los criterios se plantean en la primera columna como ‘Componente’, y en la segunda, sus respectivos ‘elementos’. Decidí independizar el análisis de los criterios, cuando se trataba de las herramientas para aprender teoría y evaluarla (Cuadro 5), de las herramientas cuyo fuerte es la visualización y la animación (Cuadro 4).

Cuadro 4. Herramientas innovadoras

Componente	Elemento	Waterbear	Scratch	Blockly	App Inventor	Rebeca a través del espejo
Facilidad de uso	Instalación	No	Si	Si	Si	Si
	Interfaz de Usuario	Si	Si	Si	Si	No
Características del producto	Licencia	Apache v2	GNU GPL v2 y Scratch Source Code	Apache License 2.0	MIT	GNU v3
	Documentación	aceptable documentación técnica	Completa documentación	Poca Documentación General aceptable documentación técnica	Completa documentación	Buena documentación
	Idioma Español	No	Si	Si	No	Si
Objetivo	Ámbito	Educación	Educación	Educación	Educación / Producción	Educación
	Descripción			Un entorno de programación visual para hacer la programación divertida y accesible		
Características técnicas	Tipo	Web	Escritorio	Web	Web	Escritorio
	Repositorio	https://github.com/waterbear1/ang/waterbear/	http://wiki.scratch.mit.edu/wiki/Scratch_Source_Code	https://code.google.com/p/blockly/	https://github.com/inf-cmi/appinventor-sources	http://code.google.com/p/rebeca/
	Personas Involucradas	12	--	21	36	3

Fuente: Elaboración propia

Cuadro 5. Herramientas LMS para acompañar el proceso de enseñanza-aprendizaje

Componente	Elemento	Chamilo	Claroline	Dokeos	Moodle	Edx platform
Facilidad de uso	Instalación	Si	Si	Si	Si	No
	Interfaz de Usuario	Si	Si	Si	No	Si
Características del producto	Licencia	GPL v3	GNU GPL	GNU GPL	GPL v3	AGPL v3
	Documentación	Completa	Completa	Completa	Completa	Completa
	Idioma Español	Si	Si	Si	Si	Si
Objetivo	Ámbito	Educación	Educación	Educación	Educación	Educación
	Tipo de Herramienta	LMS	LMS	LMS	LMS	LMS
	Descripción					
Características técnicas	Tipo	Web	Web	Web	Web	Web
	Repositorio	https://github.com/chamilo/chamilo-lms	https://github.com/claroline/Claroline	http://sourceforge.net/projects/dokeos	http://git.moodle.org/gw	https://github.com/edx/edx-platform
	Persona involucradas desarrollo	38	12	--	197	110

Fuente: Elaboración propia

Dado que por una parte ya había sintetizado de alguna manera los problemas detectados por el mundo académico, y desde luego en ese momento ya me encontraba con el reflejo palpable de las dificultades presentes en mi contexto laboral, es decir, en mi curso de Programación de Computadores (gracias al diagnóstico), y por el otro ya visualizaba mejor las diversas posibilidades que brindaba el Software Libre, me propuse una matriz que confrontara las diversas herramientas con los problemas identificados en el diagnóstico (Figura 1).

En la Figura 1, la viñeta (✓) indica la posibilidad intrínseca que posee la herramienta para vencer el problema (P). Es entendible la razón del porqué Blockly termina siendo una herramienta predilecta, y por lo tanto, una buena opción en proyectos educativos más aun cuando se desean vencer los problemas descubiertos con el presente estudio. Se trata de una herramienta con la que se aprende jugando, sirve para opinar, y visualizar diversas iniciativas de uso, y lo mejor de todo: el código es libre, es abierto. Por su parte Chamilo LMS, se ubica

como una tendencia lo suficientemente moderna y bastante completa, ideal para acompañar el proceso de aprendizaje en el Curso de Programación de Computadores.

Figura 1. Matriz Herramientas / Problemas diagnosticados

		Problemas Diagnosticados				
		P ₁	P ₂	P ₃	P ₄	P ₅
Herramientas	Waterbear	✓				✓
	Scratch		✓	✓	✓	✓
	Blockly	✓	✓	✓	✓	✓
	App Inventor	✓	✓	✓		✓
	Rebeca a través del espejo	✓				✓
	Chamilo		✓	✓	✓	✓
	Clarine		✓			
	Dokeos		✓	✓	✓	✓
	Moodle		✓			
	Edx Platform		✓	✓	✓	✓
Convenciones: P ₁ : Elevado nivel de complejidad en el aprendizaje de algoritmos P ₂ : Problemas relacionados con la enseñanza de teoría y evaluación de ritmos particulares de aprendizaje P ₃ : Insatisfacción con el proceso enseñanza - aprendizaje P ₄ : Desmotivación en el estudiantado (encuentra complejo o poco llamativo el curso) P ₅ : Poca sentido del curso con respecto a la profesión (Ingenierías diversas a la de Sistemas)						

Fuente: Elaboración propia

El desarrollo de esta experiencia me iba confirmando cómo la relación paralela del diagnóstico, el análisis de antecedentes y de documento dicientes, junto al reconocimiento de oportunidades de uso de Software Libre, iban a ganar invaluable precio antes de la misma intervención curricular.

Pues los resultados arrojados luego de la revisión bibliográfica y del diagnóstico (Cuadro 6), me ratificaban que los estudiantes llegan con problemas de motivación (Villalobos Salcedo, 2009); que es común hallar estudiantes insatisfechos con el aprendizaje por encontrarlo difícil y poco útil para sus estudios y vida profesional (Bermúdez, 2007); que se descuidan muchas veces las habilidades que deben adquirir, reduciendo los cursos a un

Identificación del problema	Descripción del problema	Fuente
Problemas de tipo académico y Problemas en el proceso de enseñanza aprendizaje	<ul style="list-style-type: none"> • Los estudiantes llegan con problemas de motivación. (Villalobos Salcedo, 2009) • Es común hallar estudiantes insatisfechos con el aprendizaje de la programación por encontrarlo difícil y poco útil para sus estudios y vida profesional. (Bermúdez, 2007) • Se descuidan muchas veces las habilidades que deben adquirir, reduciendo los cursos a un recorrido de estructuras sintácticas de un lenguaje de programación. (Villalobos Salcedo, 2009). • El hecho de reescribir los algoritmos hasta ponerlos a punto es operativamente complicada cuando se trabaja con los elementos tradicionales como lápiz y papel, tiza y pizarrón, etc. (Moroni & Señas, 2005) • Entre un grupo es común apreciar diferencias de conocimientos entre los estudiantes (Moroni & Señas, 2005) • Generalmente se espera que quienes aprueben el curso lo hagan porque son capaces de realizar programas con un grado de dificultad mayor al que poseen la mayoría de estudiantes. (Martínez, 2005) 	Revisión Bibliográfica

Fuente: Elaboración propia

Tal como lo había anunciado en el punto referente al método, la tercera etapa también ha sido llevada de manera paralela al diagnóstico y al análisis documental, con la meta de ir definiendo lo más apropiado, innovador y atractivo que se descubriera en el mundo académico con respecto a recursos del Software Libre, y que se puedan sugerir e implementar en el plan de estudios del curso de Programación.

Camino a los resultados: Se gana en la línea de salida, no en la línea de llegada

Con la certeza de que aún sin ver resultados palpables, el acercamiento logrado había sido de hecho, supremamente enriquecedor, la posibilidad de seleccionar los recursos en Software Libre para atender la problemática del Curso Programación para Computadores, ya se acercaba. En estos términos, me permito presentar a manera de hallazgos, las posibilidades de uso de Software no privativo, señalando los recursos y las herramientas interactivas

(Cuadro 7) que iba deduciendo a medida que progresaba en el estudio, como una oportunidad de solución a los problemas identificados y sintetizados en el Cuadro inmediatamente anterior.

Cuadro 7. Paralelo entre problemas y oportunidades de uso del Software Libre

Identificación del problema	Oportunidades de uso del SL
<p>Problemas de tipo académico (cognitivo)</p> <p>Problemas proceso de enseñanza</p> <p>Según docentes, estudiantes y auto-reflexión del investigador</p>	<p>Con Blockly el docente de programación de Computadores puede crear ejercicios interactivos y gráficos animados con los que el estudiante propondrá soluciones mediante lógica; así el estudiante aprenderá programación respondiendo a los principales desafíos.</p> <p>Mediante Chamilo LMS, por su parte, es posible proporcionar toda la teoría de la materia y foros en los cuales los estudiantes y profesores pueden resolver dudas y problemas presentes en los ejercicios.</p>
<p>Problemas de tipo académico y Problemas en el proceso de enseñanza aprendizaje</p> <p>Según:</p> <p>(Bermúdez, 2007)</p> <p>(Villalobos Salcedo, 2009)</p> <p>(Moroni & Señas, 2005)</p> <p>(Martínez, 2005)</p>	<p>Claroline es otra opción de software libre. Ella, frente a otras herramientas como Moodle, le apuesta por la simplicidad, en contraste con un cierto grado de complejidad que se encuentra en el uso de Moodle.</p> <p>La presentación de Moodle, sin embargo, está más en sintonía con la web de hoy.</p> <p>Con Scratch y Rebeca resolver ejercicios resulta entretenido, se puede aprender lógica de programación como si se tratase de un juego. Son herramientas fascinantes por ser valiosamente interactivas.</p>

Fuente: Elaboración propia

Se logra destacar, en el transcurso de la investigación, y más precisamente en su etapa de aprovechamiento del diagnóstico, que las herramientas o métodos identificados en el proceso de enseñanza aprendizaje van desde herramientas libres usadas habitualmente como gestor de asignaturas o cursos (el caso de Moodle), hasta llegar a aplicaciones como Scratch en código abierto. Para alcanzar dicha comparación, realicé previamente una pesquisa de la literatura científica o en línea sobre herramientas o recursos del Software Libre y las maneras o métodos como se han usado para la educación en Programación de Computadores. Esta

pesquisa se logra sintetizar hasta llevarla a un cuadro que expusiera de manera precisa y cercana lo que en realidad se acogería como actividades, frente a las más opcionadas herramientas, así no se contara con casos precisos de éxito dentro de Cursos de Programación de Computadores (Cuadro 8).

Cuadro 8. Métodos o herramientas considerados para implementación

Diagnóstico Implementación Recurso SL	PROBLEMA ATENDIDO SEGÚN DIAGNÓSTICO	ACTIVIDAD DEL PLAN DE CURSO EN EL QUE SE IMPLEMENTARÍA
Blockly	Problemas de tipo académico Problemas de proceso de enseñanza	Socializar con los estudiantes la Herramienta, proponer ejercicios, evaluar resultados y con base a los resultados proponer más ejercicios de tal manera que la dificultad se va haciendo más alta. Cuando ya se haya progresado en un 50% aproximadamente de la asignatura, mostrar al estudiante el código que puede generar Blockly a otros lenguajes como JavaScript o Python. Mostrar las estructuras similares, y de esta manera incentivar los estudiantes a escribir código directamente.
Chamilo LMS	Problemas de tipo académico y Problemas en el proceso de enseñanza aprendizaje	Crear espacios para entregar la teoría de todo el curso, organizado para hacer entregas semanales conforme avanza la materia. Se deberá entregar también material sobre Blockly y otros lenguajes. Es necesario crear canales como foros para resolver dudas y recibir retroalimentación por parte de los estudiantes. Puede considerarse crear una wiki donde se registren las experiencias del docente en el proceso de enseñanza con el fin de compartirlo con sus colegas.

Fuente: Elaboración propia

Todo esto, dado que el Plan del Curso se encuentra delineado por el aprendizaje de un lenguaje como C++, y que este lenguaje presenta una empinada curva de aprendizaje para el estudiante que nunca ha escrito código, considero estar ante una hipotética razón por la que ellos temen a la asignatura, y por la que la deserción académica es más marcada que en otras materias. El Plan de Curso vigente, se ha venido orientando exclusivamente al aprendizaje

de dicho lenguaje, y tal vez se ha dejado a un lado el fomento de las habilidades para que de manera progresiva los estudiantes se apropien de prácticas innovadoras.

Me propongo, entonces, una implementación parcial del curso, que no esté ligado a algún lenguaje de programación específico, mientras se plantea que sea un lenguaje moderno de alto nivel. Se considera dar clases prácticas y teóricas con herramientas innovadoras, de fácil uso y portables, adaptables a cualquier dispositivo y que puedan ser usadas fuera y dentro de la clase. Considerando estas necesidades, las herramientas debían ser en ambiente Web.

Se necesitaba una herramienta o lenguaje gráfico que fuera de fácil uso para los estudiantes, que permitiera explorar todas las estructuras de un lenguaje de programación tradicional. Con esa necesidad diagnosticada, el que más se ajustaba a las necesidades era Blockly; un entorno de programación que permite por medio de estructuras gráficas hacer el camino hacia la programación de computadores algo más natural y fácil al estudiante. Blockly es una herramienta que permite crear un lenguaje de programación con una leve curva de aprendizaje para el alumno que nunca antes ha escrito código.

Como se ha partido de la mirada reflexiva hecha a otras experiencias logradas en ambientes similares, me propuse que al finalizar el curso, se llevara a cabo esa breve implementación. La actividad se dispuso para ocho días aproximadamente, y se segmentó en dos principales etapas.

Etapa Uno: Socialización de la metodología con los estudiantes, presentación de las herramientas y técnicas para resolver los problemas planteados por el docente, quien brinda opciones para su real análisis, desarrollo e implementación, acogiendo el lenguaje de programación.

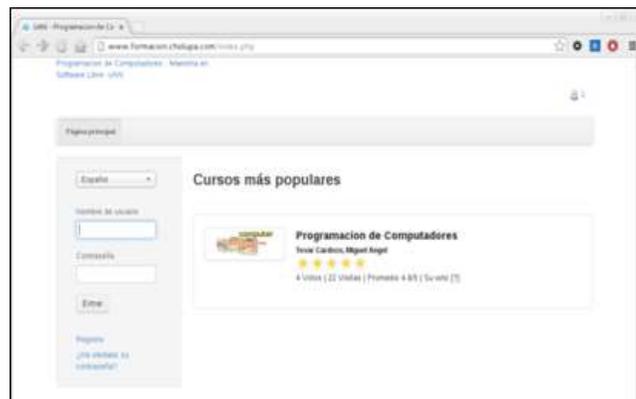
Etapa Dos: Evaluación y retroalimentación; el docente aprecia el avance de los estudiantes, de los temas propuestos y repite la dinámica de la fase uno con los nuevos temas según el plan del curso.

La implementación parcial

A través de algunas capturas de pantalla, apoyo la interesante ejecución de las actividades realizadas con los estudiantes pertenecientes al grupo control. A continuación, su síntesis.

A. Se presentó la plataforma Chamilo a los estudiantes. Por medio de la url www.formacion.cholupa.com, los alumnos ingresaron y se registraron con sus datos. El registro del estudiante es usado para personalizar su intervención activa, a la vez que el docente y los demás miembros pueden identificarse mutuamente; al docente se le facilita la posterior evaluación de los mismos. (Figura 1)

Figura 1. Captura de pantalla Página Principal propuesta del docente investigador



B. La propuesta incluye la interfaz del docente. Desde allí él puede gestionar sus cursos, admitir estudiantes, publicar material, etc. (Figura 2).

Figura 2. Interfaz para el Docente



Además, el docente crea materiales para los estudiantes por medio de sencillas herramientas de edición. Por otra parte, la propuesta incluye la posibilidad de que los estudiantes y docentes participen en los foros de forma activa.

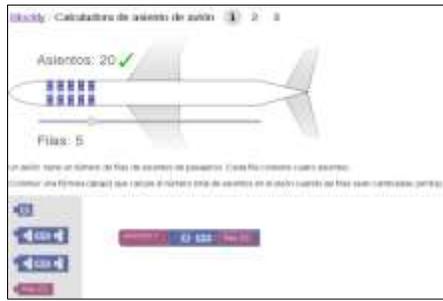
C. Como parte de la ejecución de las actividades, destaco de manera central el aprovechamiento de la herramienta Blockly. Con un ejercicio muy simple, se ha introducido al estudiante en la mecánica de trabajo con un lenguaje visual (Figura 3).

Figura 3. Introducción a Blockly



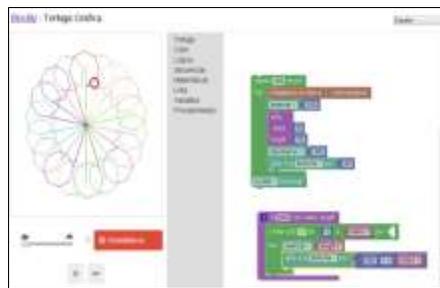
Usando la herramienta Blockly, el estudiante resuelva el problema con los bloques entregados; son 3 ejercicios en este nivel (Figura 4).

Figura 4. Ejercicio de cálculo



En esta etapa de la ejecución de las actividades programadas, los estudiantes han reconocido las características del lenguaje y demostraron poder desarrollar ejercicios propuestos por el docente y por ellos mismos. La Figura 5 evidencia un ejercicio propuesto por un estudiante.

Figura 5. Ejercicio Libre



En este ejercicio, el estudiante escribe un programa con Blockly y lo ha exportado hacia otros lenguajes. En la Figura 6 se puede apreciar la captura de los 3 lenguajes.

Figura 6. Creando Aplicaciones



D. Dentro de la ejecución de las actividades, se destaca la posibilidad para los actores de ejercer roles más activos y significativos. Se puede aseverar que fue creciente el interés y

participación por parte de los estudiantes quienes constantemente acudían al docente para interrogarle por temas que estaban más allá de lo programado en esa tutoría.

E. Para finalizar la descripción de lo que fue en sí la ejecución de las actividades, se destaca que ultimando la programación, los estudiantes iniciaron el ejercicio de empezar a escribir código JavaScript desde cero, ya sin ayuda de la herramienta Blockly.

Al final de las tareas, se instrumentó una nueva guía de preguntas con la cual se observara tanto en el grupo experimental como en el grupo control, la apropiación de los contenidos, el trabajo participativo alcanzado durante el desarrollo de la temática final, la vinculación de los equipos tecnológicos al trabajo del curso, el proceso que más favoreció la comprensión del paradigma de la programación orientada a objetos y, de paso, atender posibles recomendaciones que harían al docente de Programación en Computadores. Con base a dicha entrevista, me permito exponer más firmemente los resultados confrontándoles, de cierta manera con el marco referencial, y las posteriores conclusiones.

Resultados

Con el análisis de la información recolectada a través del instrumento dispuesto, se descubre lo que podría sostenerse y, de cierta manera, lo que podría avanzarse para proponer un definitivo Plan de Curso guiado por la implementación de herramientas innovadoras, basado en el aprovechamiento del Software Libre, en la materia Programación de Computadores. Aquí, desde luego, se expone el análisis cualitativo de lo arrojado por el instrumento del que ya se hacía mención. Para ello, decidí tomar pregunta por pregunta, identificar las respuestas más dicientes de manera literal, y manifestar la precisa reflexión investigativa.

La investigación en estos términos, confirmó que los estudiantes que pretenden aprender programación demuestran más progresos académicos cuando se acude a herramientas de Software Libre (Gómez-Albarrán, 2005). Las ventajas de la materia se constataron al final del curso con la implementación parcial, cuando los estudiantes del grupo experimental expresaron que sus miedos e inseguridades, propias de quien posee incluso débiles conocimientos, se tornaron menos fuertes ante el empleo de los recursos innovadores presentados por el docente. Debilidades que aún persistían en un buen número de quienes pertenecieron al grupo control.

Los alumnos pertenecientes al grupo control no estuvieron seguros de progresos significativos en el curso de Programación de Computadores, lo que me lleva a pensar que un Plan de Curso apoyado en las herramientas en Software no privativo, arroja mayor apropiación de los contenidos brindados. Por otra parte, se hace indiscutible la oportunidad de seguir sosteniendo para todo el currículo de la materia y para todos los grupos que se abren cada semestre en la Universidad Antonio Nariño de Neiva en su facultad de Ingeniería, el aprovechamiento del Software Libre.

Sobre la importancia de fomentar las oportunidades de participación por parte del estudiante, se viene hablando desde hace casi dos décadas (Marqués, 1996), (GNU Operating Systems, s/f), (Shockey, 2005), (Romero, 2006), y estudios más recientes como el de los Hermanos Cisneros Arocha (2008) y el de Vizconde et al. (2012) lo recalcan, ofreciendo mayor evidencia en cuanto a que con el aprovechamiento del Software Libre, esta práctica pedagógica es atendida.

Estudios como el de los Hermanos Cisneros Arocha (2008) anunciaban la efectividad en la implementación del software educativo, por el apoyo, refuerzo y evaluación del

aprendizaje, y este estudio lo ratifica. Al igual, se destaca el acierto en la selección de las herramientas Blockly y Chamilo LMS, aplicadas con criterios de selección y obedeciendo a un diagnóstico que, como se apreció con las respuestas del grupo control, conserva su vigencia. Seguirá siendo clave el acompañamiento del docente que en plena era digital, debe trascender del aula; y no se descarta la necesidad de tener en cuenta los ritmos propios de aprendizaje. Todas estas condiciones, se suplen en el acogimiento de recursos innovadoras en entornos Web.

Por otra parte, con la aplicación del instrumento último, se evidenció además que los equipos como son los celulares que ya cuentan con tecnología de punta y están al alcance de los jóvenes de las Universidades, o las Tablet, y no solo los portátiles y los PC, deben ser aprovechados para el proceso de aprendizaje. Estudios como el de Salinas (2004), Cebrián (2004) y Duque (2009), citados dentro de esta investigación, ya lo hacían ver.

Siguiendo las respuestas de los estudiantes pertenecientes al grupo control, no se descubrió ninguna evidencia concreta de que la problemática detectada con el diagnóstico se hubiera superado. No se encontraron datos brindados desde el instrumento, que permitieran pensar que las dificultades observadas al inicio del presente estudio, se vencieran sin la intervención de estrategias que apoyaran el acogimiento del Software Libre. Por el contrario, varios estudiantes pertenecientes al grupo experimental opinaron sobre la necesidad de un Plan de Curso que de manera amplia (todo el tiempo), acudiera a herramientas como las empleadas al final del semestre. Por su parte, algunos alumnos del grupo control, guardan las esperanzas de una actualización en el currículo, pues expresaban que lo ven “algo anticuado”.

Destacable en estos resultados, el reconocimiento que se dio con la implementación, al aprovechamiento de los equipos que portan los estudiantes. Es definitivo seguir apoyando

el hecho de que los estudiantes usen sus *netbooks* y equipos celulares con tecnología Android, por ejemplo, pues eso se convierte en una ventaja a la hora de efectuar programaciones en la clase, apoyadas en herramientas Web. Muy seguramente en cualquier lugar en el que se encuentren, van a abordar la herramienta innovadora, van a recibir un mensaje o van a poder participar en un foro. Todo esto, reforzará al estudiante, y de paso a sus compañeros en un entorno interactivo y colaborativo, desde luego.

Al igual, aprovechar herramientas como Blockly y Chamilo LMS para definir y ordenar y luego seguir un procedimiento en la solución de un problema, no solo durante el tiempo de la clase sino desde el contexto del hogar, es relevante, dado que las prácticas del proceso de enseñanza aprendizaje de la programación no pueden seguirse abordando de manera tradicional, sino con propuestas innovadoras. La herramienta Blockly puede complementar los contenidos dispuestos en el Plan de Curso de Programación de Computadores, fomentando la acción de programar, gracias a un sistema con bloques, como si se tratara de armar un rompecabezas. Chamilo LMS por su parte, trasciende el contexto del aula, presentándose entre estas dos herramientas, ciertas características con las cuales la materia se fortalece en su proceso de enseñanza aprendizaje:

a. Admitirse en Chamilo LMS diferentes tipos de archivos: Texto, imagen, audio, video, animaciones, enlaces a otros documentos o a sitios diversos de la Web. Es decir, brillante por la inclusión de textos multimediales.

b. Fácil empleo; versión mejorada: A diferencia de Moodle, Chamilo LMS posee una interfaz de usuario sencilla, para ser utilizada de manera que su empleo sea al alcance de todos. Vale la pena recordar que se han elegido herramientas recientes; Chamilo, por su parte, recoge las experiencias de su antecesor inmediato que es Dokeos, y esta a su vez, lo había

hecho sobre Claroline. Igual sucede con Blockly. Blockly se ha inspirado en Lego, en Scratch, y en App Inventor.

c. Organización cronológica y temática de contenidos: La herramienta Chamilo LMS permite a futuro, que los contenidos del curso sean organizados por su docente.

d. Promueve la participación: Acoger herramientas como estas, permiten a mediano plazo la participación creciente de los estudiantes y de sus docentes. Lamentablemente ha imperado por décadas el tradicionalismo, y esta ventaja que es propia de tendencias pedagógicas donde la acción es un ingrediente fuerte, no es fácil motivar, ni lograr ver resultados a corto plazo.

A manera de conclusiones

Se diseñó una propuesta de intervención para el curso Programación de Computadores ofrecido en el primer año de las Carreras dispuestas en la Facultad de Ingeniería de la Universidad Antonio Nariño sede Neiva. La idea clara consistió en llevar a cabo dicha intervención, con el apoyo de recursos de Software Libre. Entre el análisis posterior al estudio incluyendo la implementación parcial de un reformado Plan de Curso, se aprecia que:

Se puede incorporar ejercicios y problemas en forma gradual usando la herramienta Blockly, atendiendo las dificultades manifestadas en el camino, por los estudiantes. Con la implementación de Chamilo LMS, y también de Blockly, se atiende favorablemente la heterogeneidad de los grupos donde se experimentó el método, y se aprecia los grados reales de participación en clase o fuera de ella.

Las prácticas innovadoras favorecen la dialógica entre estudiante y docente, pero también entre pares, de manera que sean atendidas las diferencias, las propuestas y los aportes de todos, no solo en el espacio del aula, sino en otros contextos, generando mejores y mayores

espacios comunicativos y, desde luego, de aprendizaje. Por lo tanto, las formas de enseñar y de aprender en plena era digital no pueden seguir siendo inmutables en la Universidad Antonio Nariño de la ciudad de Neiva, pues de ella toda una comunidad espera mayor adaptabilidad al ritmo acelerado de las TIC.

Referencias Bibliográficas

- Benito, M., García, F., Portillo, J., & Romo, J. (2007). *Descripción y características del concepto Nativo Digital. En M. B. otros, Nativos digitales y modelos de aprendizaje.* (págs. 2-3). Universidad de País Vasco / Euskal Herriko Unibertsitatea (UPV/EHU).
- Bermúdez, C. P. (2007). *Aprendizaje de Programación de Computadores y Resolución de Problemas en un Ambiente de Trabajo en Colaboración.* Bogotá: Universidad de los Andes.
- Cebrián de la Serna, M. (2004). Diseño y producción de materiales didácticos por profesores y estudiantes para la innovación educativa. In *Tecnologías para la educación: diseño, producción y evaluación de medios para la formación docente* (pp. 31-46). Alianza Editorial.
- Cisneros Arocha, J. V., & Cisneros Arocha, E. O. (2008). *Videojuego Educativo para la enseñanza de la Arquitectura del Computador e Introducción a la Programación apoyadas en Software Libre.* Caracas.
- Duque, M. (2009). Ciencia, tecnología e innovación en ingeniería, como aporte a la competitividad del país. *Cómo formar ingenieros para la innovación* (pág. 50). Bogotá: Universidad de los Andes.
- Gómez-Albarrán, M. (2005). La enseñanza y el aprendizaje de la programación: Un estudio de las herramientas de apoyo de software. *The Computer Journal* , 48 (2), 130-

- 144.Hernández, R. F. (2006). *Metodología de la Investigación*. México: Mc Graw Hill.
- Maggio, M. (2002). *El Tutor en la Educación a Distancia*. Buenos Aires: Amorrortu Editores, S. A.
- Malaver, N., Rey, C., & Rodríguez, J. (2011). Enseñanza de programación en el Politécnico Grancolombiano. Situación actual y aplicación de TIC como alternativa de mejora. *open Journal Systems* Vol 1, No 1.
- Martínez, Y. (2005). *En busca de una nueva forma de enseñar a programar*. Obtenido de <http://www.mty.itesm.mx/rectoria/dda/rieee/pdf-05/28%28DTIE%29.YolandaMtz..pdf>
- Marqués, P. (1995). Metodología para la elaboración de software educativo en SoftwareEducativo. Guía de uso y metodología de diseño. Barcelona: Estel.Moroni,
- N., & Señas, P. (2005). Estrategias para la enseñanza de la programación . *JEITICS*, 254-269.
- Romero, T. A. (2006). Moodle, Unimos Mentes, Creamos Conocimiento Libre. Ponencia presentada al VI Congreso Internacional Virtual de Educación. Islas Baleares: CIVE.
- Sabino, C. (2002). *El proceso de investigación*. Caracas: Panapo.
- Salgado Lévano, A. (2007). Investigación cualitativa: diseños, evaluación del rigor metodológico y retos. *LIBERABIT: Lima (Perú)* 13, 71-78.
- Salinas, J. (2004). Los recursos didácticos y la innovación educativa. *Comunicación y Pedagogía*, 36 - 39.
- Shockey, K. &. (2005). Using open source to enhance learning. In *Information Technology Based Higher Education and Training.*, (págs. pp. F2A-7).

- Stallman, R. (1999). El sistema operativo GNU y el movimiento de software libre de fuentes abiertas: voces desde la revolución de código abierto. Obtenido de <http://www.gnu.org/education/edu-why.html>
- Villalobos Salcedo, J. A. (2009). *Proyecto CUPI2 – Una solución integral al problema de enseñar y aprender a programar*. Obtenido de http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-205832_recurso_1.pdf
- Vizconde Veraszto, E., Franco de Camargo , J. T., Barros Filho , J., García García, F., Ferreira do Amaral , S., & Bortoloti , A. A. (2012). Producción de animaciones modeladas por ordenador utilizando software libre. *Icono* No. 14, 198-212.