



La programación de software como competencia profesional del ingeniero en sistemas informáticos

Software programming as a professional competence of computing system engineers

M. Sc. Esthela María San Andrés Laz

esthelasanandres@hotmail.com

Universidad Técnica de Manabí, Ecuador

Dr .C. Milagros de la Caridad Rodríguez Andino

mrodriguezandino@gmail.com

Universidad de la Habana

Dr.C. Yaile Caballero Mota

yaile.caballero@reduc.edu.cu

Universidad de Camagüey “Ignacio Agramonte Loynaz”

Las autoras son profesoras universitarias. **San Andrés Laz** está titulado como Magíster en Informática de Gestión y Nuevas Tecnologías por la Universidad Santa María de Chile (USM) y Magíster en Gerencia Educativa por la Universidad Estatal del Sur de Manabí (UNESUM); ejerce como profesor principal de la Universidad Técnica de Manabí (Ecuador) en la Facultad de Ciencias Informáticas y realiza estudios relacionados con la formación de competencia profesionales; actualmente hace su doctorado en Ciencias Pedagógicas en la Universidad de Camagüey (Cuba). **Rodríguez Andino** es Doctora en Ciencias de la Educación y Profesora Titular de la Facultad de Educación a Distancia de la Universidad de la Habana. **Caballero Mota** es Doctora en Ciencias Técnicas y Profesora Titular de la Facultad de Informática de la Universidad de Camagüey “Ignacio Agramonte Loynaz”, junto a Rodríguez Andino ejerce la tutoría de la autora principal en el doctorado referido.

RESUMEN

En el nuevo milenio con el impetuoso avance de las TIC las organizaciones y empresas demandan el desarrollo de sistemas computacionales para la automatización de diferentes procesos que permitan mejorar e incrementar su productividad y eficiencia, por lo que exigen de las instituciones de educación superior la preparación de profesionales que puedan aportar sus esfuerzos en la construcción de soluciones de negocio utilizando la tecnología computacional de vanguardia en la programación de software. El presente artículo describe los resultados parciales de una investigación que se realiza con el objetivo de formar y desarrollar la competencia profesional programar software en los Ingenieros en Sistemas Informáticos. Se realizó un diagnóstico en la carrera de Ingeniería en Sistemas Informáticos de la UTM donde se aplicaron encuestas, entrevistas y observaciones a clases, tomando una muestra de 179 estudiantes y siete profesores, los resultados identificaron insuficiencias de los estudiantes en la formación teórico-práctica en programación, lo que limita su futuro desempeño profesional. En el desarrollo de la investigación, también se utilizó el análisis y la síntesis para resumir los fundamentos teóricos esenciales que sirven de sustento a los resultados que se presentan, a través de la asignatura Programación I se

muestra la relación entre la competencia y la asignatura y los resultados de aprendizaje que se deben alcanzar con las respectivas evidencias para su evaluación.

Palabras clave: competencias, competencias profesionales, aprendizaje,

ABSTRACT

In the new millennium with the impetuous progress of ICTs the organizations and companies demand the development of computer systems for automatizing processes which allow improving and increasing their productivity and efficiency, so they require from the higher education institutions the training of professionals capable of contributing to develop business solutions using the latest computational technology in software programming. This article describes partial results of an investigation that was carried out with the aim of training and developing the professional competence of programming software on computer systems engineers. A diagnosis was made in systems computing engineering of the Technical University of Manabí by applying surveys, interviews and observations to lessons, several shortcomings in the theoretical and practical training in programming were identified in a sample of 179 students and 7 professors. Analysis and synthesis were carried out to summarize the essential theoretical foundations that support the results presented, through the Programming I Course, shows the relationship between the competence that is related to the subject and learning outcomes to be achieved with the respective evidence for its assessment.

Keywords: competence, professional competencies, learning.

Las universidades del siglo XXI están orientadas a preparar profesionales competentes, capaces de realizar con eficiencia y eficacia las tareas de su profesión y que a la vez contribuyan al desarrollo científico-tecnológico del entorno donde están enclavadas, el cual es un reclamo global de la sociedad actual. Estos criterios se manifiestan en el proceso de evaluación señalado en el Modelo General para la evaluación de carreras con fines de acreditación (CEAACES, 2011) y que se está implementando en el área de la educación ecuatoriana y latinoamericana con el fin de aplicar los principios del Proyecto Tunning para América Latina del cual Ecuador es participante.

En la literatura consultada varios autores coinciden en señalar que la evaluación del aprendizaje de los estudiantes no sólo debería estar basada en los conocimientos sobre la materia en cuestión, sino que debe incluir una evaluación basada en las competencias, capacidades y procesos estrechamente relacionadas con el trabajo y las actividades que conducen al progreso del estudiante y a su articulación con los perfiles profesionales definidos. Esto sitúa, a los profesores, en un escenario distinto, con un enfoque centrado en el estudiante. En este enfoque definir los resultados de aprendizaje facilita la tarea del profesor y orienta el aprendizaje del alumno, además de anticipar qué serán capaces de hacer los estudiantes una vez acabado el aprendizaje, en qué condiciones deberán hacerlo y qué criterios se pueden utilizar para valorar lo aprendido. Asimismo los resultados de aprendizaje, guían la elección de las estrategias y las actividades de aprendizaje, los métodos de enseñanza y el sistema de evaluación.

En la Universidad Técnica de Manabí (UTM), los docentes están enfrascados en la aplicación de estas concepciones novedosas al proceso de enseñanza aprendizaje de las diferentes materias, con miras a los futuros procesos de acreditación. Esto ha llevado a la autora principal de este trabajo a realizar una investigación en el marco de la carrera de Ingeniería en Sistemas Informáticos, con el propósito de

contribuir a una formación adecuada en la materia de programación que les permita a los estudiantes un futuro profesional exitoso.

El Ingeniero en Sistemas Informáticos desempeña un papel importante en la sociedad moderna. Se trata de una sociedad impregnada de tecnología, en la cual esta se ha convertido en el instrumento de todas las actividades sociales, por lo tanto los ingenieros tienen que desarrollar competencias que le permitan un eficiente desempeño ante las funciones que deben realizar.

La formación de la competencia programar software es un tema de gran actualidad. A nivel internacional son diversos los autores, instituciones y organizaciones que se dedican hoy al estudio de la misma y desarrollan experiencias docentes, toda vez que se reconoce la necesidad de que cada ingeniero sea capaz de escribir en código computacional un programa mediante un lenguaje de programación, lo cual es esencial en cualquier carrera relacionada con la informática, siendo importante en la mayoría de las asignaturas de la carrera y en el campo profesional del egresado, lo que constituye una competencia profesional esencial.

El presente artículo describe los resultados parciales de una investigación que se realiza con el objetivo de formar y desarrollar la competencia programar software en los Ingenieros en Sistemas Informáticos. Se muestran los principales resultados del diagnóstico y a través de la asignatura Programación I se presenta un ejemplo de cómo relacionar el eje procesual con la asignatura, los resultados de aprendizaje que se deben alcanzar con las respectivas evidencias y los niveles de dominio con su ponderación para realizar la evaluación de la competencia.

Materiales y métodos

En el desarrollo de la presente investigación se aplicaron encuestas y entrevistas a profesores y estudiantes de la carrera de Ingeniería en Sistemas Informáticos (ISI) de la UTM con el objetivo de conocer su apreciación sobre la enseñanza-aprendizaje de la programación. Se tomó una muestra de 186 sujetos de los cuales, 179 fueron estudiantes que cursan los diferentes niveles de la carrera y 7 profesores.

La encuesta a estudiantes recoge criterios relacionados con aspectos esenciales como: si los conocimientos precedentes que poseen de programación le ayudan para enfrentar la carrera; si en las clases de laboratorio se proponen ejercicios que vinculan los aspectos teóricos con los prácticos en la solución de tareas de programación relacionadas con la actividad práctica de la profesión; si las clases de lenguajes y técnicas de programación que se imparten contribuyen a su formación como ingeniero en sistemas; si creen que el pensum de la carrera de Ingeniería en Sistemas Informáticos se orienta a responder a las demandas de la sociedad, sobre el conocimiento de técnicas de programación, que se espera del egresado de dicha carrera.

La encuesta y la observación a clases realizada a profesores de lenguaje y técnicas de programación que imparten clases en la carrera de Ingeniería en Sistemas Informáticos se aplicó con el objetivo de caracterizar el tratamiento metodológico que recibe la enseñanza de los lenguajes y técnicas de programación desde las asignaturas vinculadas con esta temática. Se valoraron entre otros los siguientes aspectos: si se hace conscientes a los estudiantes sobre los resultados o logros de aprendizaje que deben alcanzar al terminar la clase o la unidad de aprendizaje; si se utilizan métodos de enseñanza-aprendizaje que promuevan la reflexión y la colaboración y ayuden a la comprensión de las técnicas y

lenguajes de programación; si se ejecutan tareas que promuevan la generación y utilización del conocimiento relacionado con los lenguajes y técnicas de programación; si se trabaja por la orientación y motivación del estudiante hacia la necesidad de utilizar los conocimientos relacionados con los lenguajes y las técnicas de programación; si la evaluación se hace a través de tareas integradoras que permiten la vinculación de la teoría con la práctica en relación con los lenguajes y técnicas de programación.

La entrevista a profesores se realizó para caracterizar y diagnosticar el estado actual del tratamiento de la competencia programar software entre los estudiantes de Ingeniería en sistemas informáticos. Las preguntas más importantes versaron sobre: cuáles competencias consideran importantes desarrollar; cómo definen la competencia programar software; qué condiciones pedagógicas deben crearse en el proceso de formación; cómo contribuyen a motivar a sus alumnos hacia el aprendizaje de los lenguajes y técnicas de programación?; qué tipos de tareas o ejercicios utilizan en las clases de lenguajes y técnicas de programación; y cómo se desarrolla la evaluación de la competencia programar software.

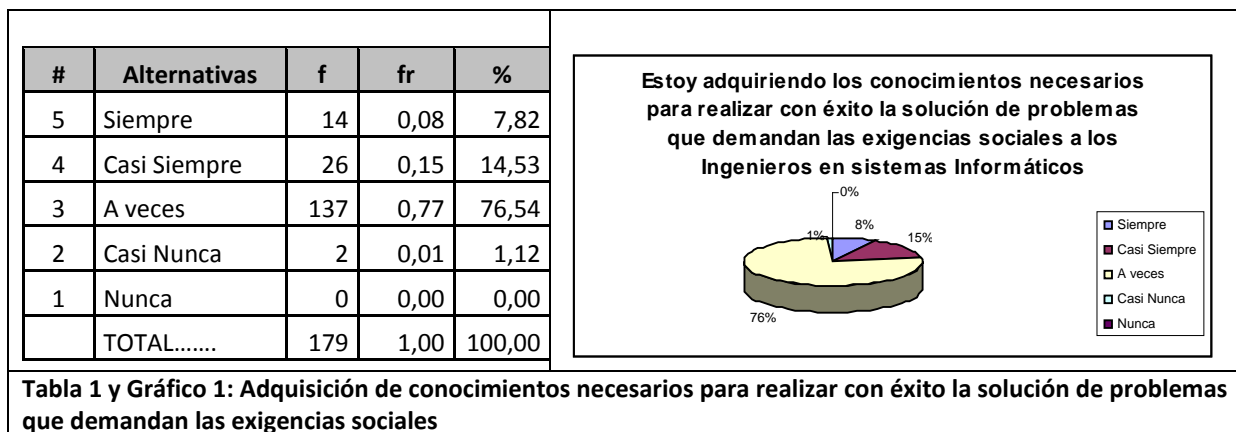
Además se utilizó el análisis y la síntesis para resumir los fundamentos teóricos esenciales que sirven de sustento a los resultados que se presentan.

Resultados y discusión

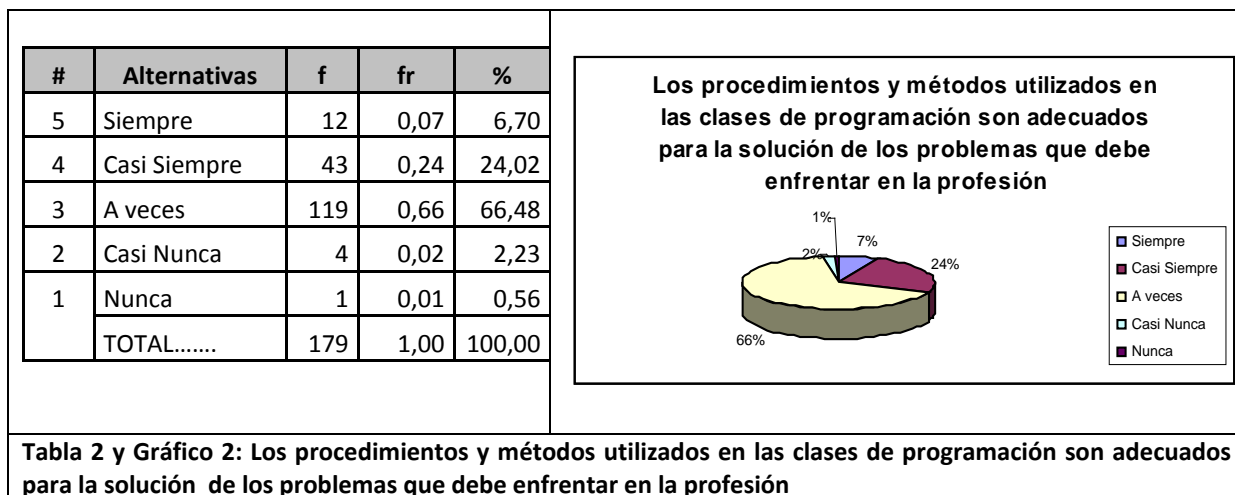
Después de realizar el procesamiento de la información a través de la aplicación informática EXCEL, los resultados del diagnóstico, han permitido conocer que existen insuficiencias relacionadas con la enseñanza aprendizaje de la programación lo que incide en el nivel de preparación que tienen los estudiantes de 8vo, 9no y 10mo nivel de la carrera de Ingeniería en Sistemas Informáticos de la UTM.

Entre las principales insuficiencias están:

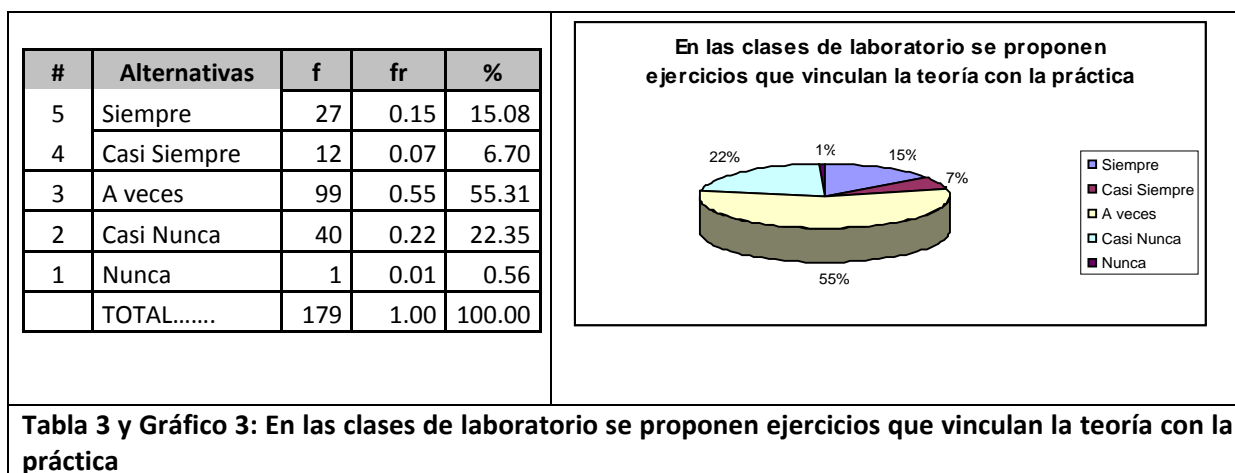
Según se observa en la Tabla 1, gráfico 1, de los estudiantes encuestados la mayoría (76 %) considera que **a veces** los conocimientos que están adquiriendo son los necesarios para realizar con éxito la solución de los problemas que demandan las exigencias sociales a los ingenieros en sistemas informáticos. Un análisis de esta insuficiencia indica que es muy importante revisar si los conocimientos que se enseñan y aprenden los estudiantes son los que realmente requiere el perfil de este profesional en la actualidad atendiendo a los cambios tecnológicos constantes que se producen en la esfera relacionada con la informática.



Más de la mitad de los estudiantes (66 %) considera que solo a **veces** los procedimientos y métodos que se utilizan en las clases de programación son los adecuados para darle solución a los problemas que debe enfrentar en la profesión (Observar Tabla 2 y Gráfico 2)



Entre un 55 y un 22 % consideran que a veces o casi nunca en las clases de laboratorio se proponen ejercicios que vinculan la teoría con la práctica. (Observar Tabla 3 y Gráfico 3)



Los resultados de la encuesta, la entrevista y la observación a clases a los profesores después de aplicar el método de triangulación evidenciaron que:

La mayoría de los docentes (72 %) no tienen suficientemente declarados los resultados o logros de aprendizaje que deben alcanzar los estudiantes en la clase o en la unidad de aprendizaje, por lo que no logran hacer conscientes a los estudiantes de los resultados que deben alcanzar.

La mayoría de los docentes (67 %) utilizan métodos de enseñanza-aprendizaje que no promueven la reflexión y la colaboración y no contribuyen a la comprensión de las técnicas y lenguajes de programación ya que se basan en la repetición o reproducción de trozos de código computacional en la

solución de los ejercicios y no se basan en la algoritmia y la búsqueda de vías para escribir en el lenguaje de programación apropiado la solución de los problemas.

En las clases se ejecutan tareas que no siempre promueven la generación y utilización del conocimiento relacionado con los lenguajes y técnicas de programación; no en todos los casos se trabaja por la orientación y motivación del estudiante hacia la necesidad de utilizar los conocimientos relacionados con los lenguajes y las técnicas de programación.

La evaluación se realizamayormente a partir de ejercicios, que solicitan al estudiante desarrollar en el lenguaje de programación estudiado un programa con nivel de complejidad simple para la solución de un problema dado, no se hace a través de tareas integradoras que permitan la vinculación de la teoría con la práctica en relación con los lenguajes y técnicas de programación.

En los resultados de la entrevista la mayoría de los docentes (90%) sitúan la competencia programar software entre aquellas de mayor importancia para los ingenieros en sistemas informáticos. Lo expresado por los docentes entrevistados y la revisión teórica realizada, permitió a las autoras sistematizar dichas ideas y proponer la siguiente definición de la competencia profesional programación de software:*el resultado de la integración, esencial y generalizada de un complejo conjunto de conocimientos y habilidades relacionadas con el proceso de diseño, codificación, depuración y mantenimiento del código fuente de programas computacionales aplicando técnicas y herramientas informáticas con valores profesionales, que se manifiestan a través de un desempeño profesional eficiente en la solución de los problemas propios de la Ingeniería en Sistemas Informáticos.*

Con relación a lascondiciones pedagógicas que deberían crearse en el proceso de formación del ingeniero en sistemas informáticos para favorecer el desarrollo de esta competencia reconocen la necesidad de aplicar métodos activos y grupales para la solución colaborativa de problemas que presenten un cierto grado de complejidad y requieran la aplicación integrada de técnicas y lenguajes de programación de avanzada acorde con las nuevas tecnologías informáticas, aceptan que existen dificultades en este aspecto que deben ser superadas. También reconocen que se requiere aplicar estrategias didácticas para favorecer la formación de esta competencia. Se sugiere además la necesidad de determinar, sobre la base de una fundamentación teórica adecuada, los resultados o logros de aprendizaje que se deben alcanzar en las diferentes materias y que tributan a la formación de la competencia programar software en este profesional.

Estos resultados señalan que es necesario repensar el trabajo que se realiza, nos indican que se necesita una fundamentación teórica para un perfeccionamiento de la formación de los estudiantes que permita alcanzar por parte de estos lascompetencias y los resultados de aprendizaje declarados para la carrera y en específico las relacionadas con la programación.

En un estudio realizado de los resultados de la reunión de Bruselas de noviembre del 2012 en el Área Informática del Proyecto Tuning para América Latina, en su proyección 2012-2013, presentado por el coordinador de esta área José Contreras se observa que los especialistas que trabajan en esta propuesta señalan que“los perfiles profesionales y competencias que se definan deben estar basados en la esencialidad que posibiliten el constante cambio, adaptabilidad del entorno y un aumento en la formalidad de la profesión y también en la especificidad que posibiliten la asimilación temprana de áreas de aplicación particulares”(2012, pág.5).

El proyecto ha propuesto 13 competencias profesionales generalizadoras para el área de Informática, Sistemas y Computación con orientación internacional. A continuación se mencionan estas competencias:

1. Aplicar el conocimiento de ciencias de la computación, de tecnologías de la información, y de las organizaciones, para desarrollar soluciones informáticas.
2. Concebir, diseñar, desarrollar y operar soluciones informáticas basándose en principios de ingeniería y estándares de calidad.
3. Aplicar el enfoque sistémico en el análisis y resolución de problemas.
4. Aplicar fundamentos matemáticos, principios algorítmicos y de programación y teorías de Ciencias de la Computación en la modelación y diseño de soluciones informáticas.
5. Desempeñar diferentes roles en proyectos informáticos, en contextos multidisciplinares y multiculturales, tanto locales como globalizados.
6. Aplicar su conocimiento en forma independiente e innovadora en la búsqueda de soluciones informáticas, con responsabilidad y compromiso social.
7. Identificar oportunidades para mejorar el desempeño de las organizaciones a través del uso eficiente y eficaz de soluciones informáticas.
8. Liderar procesos de incorporación, adaptación, transferencia y producción de soluciones informáticas para apoyar los objetivos estratégicos de las organizaciones.
9. Aplicar estándares de calidad en el desarrollo y evaluación de soluciones informáticas.
10. Comprender y aplicar los conceptos éticos, legales, económicos y financieros para la toma de decisiones y para la gestión de proyectos informáticos.
11. Liderar emprendimientos en la creación de productos y servicios vinculados con la informática.
12. Aplicar metodologías de investigación en la búsqueda, fundamentación y elaboración de soluciones informáticas.
13. Asimilar los cambios tecnológicos y sociales emergentes (Contreras, 2012, pág. 6).

Es criterio de las autoras que esta propuesta se debe revisar en cada institución y perfeccionar el perfil profesional y las competencias de las carreras vinculadas con el área informática de manera tal que se ajusten a las nuevas exigencias y demandas a nivel mundial y que emanen de los análisis del Proyecto Tuning para América Latina, con el fin de buscar coherencia en la formación de profesionales de las distintas titulaciones que corresponden a esta área, en los diferentes países respetando los aspectos de carácter específico que puedan emanar de los distintos contextos, lo que permitirá prever a los estudiantes de oportunidades para el desarrollo de trabajos en contextos internacionales.

Como se percibe, la competencia objeto de estudio en la investigación que se desarrolla es parte de la competencia profesional generalizadora *“aplicar fundamentos matemáticos, principios algorítmicos y de programación y teorías de Ciencias de la Computación en la modelación y diseño de soluciones informáticas.”*

En la literatura revisada se observa que existe consenso en varios autores, entre estos Tobón, Pimienta & García (2010); Martínez, Cegarra, & Rubio (2012); Abambari (2015) en relación a la formación, desarrollo y evaluación de una competencia. Según estos autores es necesario identificar y definir, lo más exactamente posible, cada una de las competencias de un perfil de egreso, de un programa, de manera que queden bien delimitadas. Por ello se requiere el estudio del contexto y la definición de los problemas actuales y futuros para la profesión objeto de análisis. Además para cada competencia sugieren establecer:

- Los criterios de desempeño¹ o resultados de aprendizaje asociados a la demostración de la competencia.
- Las evidencias que se requieren para evaluar los criterios o resultados de aprendizaje y que son las pruebas concretas y tangibles de la competencia y resultan esenciales para evaluar los criterios y medir el éxito de la formación.
- Nivel de desarrollo o dominio que permiten establecer los logros alcanzados en el aprendizaje.
- Ponderación o puntaje que consiste en asignarle un valor cuantitativo a los criterios respecto a su grado de contribución para valorar la competencia.

Los resultados de aprendizaje apoyan a las competencias, son más detallados y forman la base tanto del aprendizaje como de la apreciación.

Abambari señala que “la evaluación de competencias plantea la evaluación de desempeños (ejecuciones) fácilmente reconocidos y aceptados como desempeños deseados. Asimismo, debe establecerse qué deben saber y qué deben ser capaces de hacer, qué motivaciones, actitudes demuestran, de tal manera que se pueda medir el dominio logrado para un desempeño satisfactorio”(2015, pág.22).

Así en esta investigación se comparte este punto de vista y se asume el término resultados de aprendizaje, debido a que se considera que es el término más adecuado en el contexto del proceso de enseñanza aprendizaje, reconociendo que los mismos bien formulados serán útiles para indicar lo que los estudiantes deben saber, lo que los estudiantes deben comprender, y lo que los estudiantes deben ser capaces de hacer.

La definición de resultados del aprendizaje acordada en el Proyecto Tuning para América Latina (Proyecto Tuning América Latina, 2007, pág.323) es la siguiente: “*formulaciones que el estudiante debe conocer, entender o ser capaz de demostrar una vez concluido el proceso de aprendizaje*”. Los resultados del aprendizaje deben estar acompañados de evidencias de evaluación adecuadas que puedan ser empleadas para juzgar si se han conseguido los resultados previstos.

Otra definición más actual es la emitida por Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior donde se señala que “*los resultados o logros del aprendizaje describen el aprendizaje en términos de comportamientos, conocimientos y actitudes en niveles de cumplimiento específicos, es decir, lo que el alumno será capaz de conocer, comprender y de hacer al término de un proceso de aprendizaje y/o de sus estudios de la carrera*”(2011, pág. 20).

En la literatura especializada referida a lo que se considera el mejor procedimiento para redactar resultados de aprendizaje (Bingham, 1999; Fry, Ketteridge, & Marshall, 2000; Jenkins y Unwin, 2001; Moon, 2002), quienes señalan que cuando se redactan resultados de aprendizaje es útil centrarse en lo que se espera de los estudiantes sean capaces de hacer o demostrar al término del módulo o del programa.

La distinción entre los resultados del aprendizaje y la competencia se hace clara en la recomendación de European Qualifications Framework, aquí los resultados de aprendizaje se definen como “lo que un

¹Tobón, Pimienta & García (2010) en su libro “Secuencias didácticas: aprendizaje y evaluación de competencias señalan que los criterios son las pautas fundamentales que se deben tener en cuenta en la valoración de la competencia y que es el término más usado a nivel internacional, pero que es equivalente a conceptos cercanos como “Resultados de Aprendizaje” (pág.14).

aprendiz reconoce y es capaz de hacer al finalizar un proceso de aprendizaje..." mientras que la competencia significa "la capacidad demostrada para utilizar conocimientos, habilidades y capacidades personales, sociales y/o metodológicas, en el estudio o situaciones de trabajo y en el desarrollo profesional y personal...", competencias, por tanto, son las características que una persona muestran en acción (Marco europeo de cualificaciones para el aprendizaje permanente, 2011).

Otro elemento que es mencionado por Tobón, Pimienta & García en la formulación de una competencia es la necesidad de tener en cuenta los "ejes procesuales" o grandes desempeños de la competencia que dan cuenta de la estructura de la competencia como proceso sistémico, por lo general se explicitan para organizar los criterios de desempeño o resultados de aprendizaje (2010, pág.13). En la investigación que sirve de base a este artículo se asume el trabajo por ejes procesuales para la competencia profesional "Programar Software", debido a que se coincide con Tobón, Pimienta & García (2010) los que señalan que "el término *unidad de competencia*, ya no es pertinente, porque toda competencia es una unidad sistémica integrativa" (2010, pág. 71).

Según el análisis realizado en esta investigación a partir de la malla curricular de la carrera, sobre la base de los elementos teóricos valorados y de las competencias profesionales o específicas del perfil del graduado declaradas en el Proyecto Turning para América Latina, la formación de la competencia programar software debe ser un proceso que transite por alcanzar los resultados de aprendizaje en cuatro grandes ejes procesuales, estos ejes son:

- A. Formación en Programación de Software en Lenguaje Estructurado
- B. Formación en Programación de Software en Lenguaje Orientado a Objeto
- C. Formación en Programación de Software en Lenguaje Avanzado Visual y Web
- D. Desarrollo de Software

A continuación se presenta una muestra relacionada con la asignatura Programación I, que enmarca el Eje procesual A, definiendo, los criterios de desempeño o resultados de aprendizaje que se deben alcanzar, las evidencias para su evaluación, y los niveles de desarrollo o dominio que permiten establecer los logros alcanzados en el aprendizaje, para estos se toman en cuenta los cuatro niveles de dominio definidos por Tobón, Pimienta & García (2010), estos son: Nivel Inicial-Receptivo, Nivel Básico, Nivel Autónomo, Nivel Estratégico. Además se establece el puntaje asignado a cada criterio.

Dominio de Competencia: Programar Software	
Nivel que cursan los estudiantes: Primero.	Asignatura: Programación I
Eje procesual 1: Formación en programación de software en lenguaje estructurado.	
Criterio de desempeño o Resultado de Aprendizaje	
1.1 Describe algoritmos de programación informático mediante el diseño de diagramas de flujos y pseudocódigos verificados a través de la prueba de escritorio.	
Evidencias	

- a) Conceptualiza la importancia que tiene la programación de computadoras. (saber conocer)
- b) Identifica y diferencia eficientemente los tipos de lenguajes utilizados en programación: compilados, interpretados, estructurados y orientados a objetos. (saber hacer y conocer)
- c) Identifica con claridad los principales conceptos relacionados con la programación: algoritmos, diagramas de flujo, pseudocódigos, tipos, secuencias, reglas, prueba de escritorio. (saber conocer)
- d) Define y establece diferencias entre lo que es variables, constante, tipos de datos. (saber conocer)
- e) Reconoce y selecciona correctamente la simbología a utilizar en el diseño del diagrama de flujo. (saber conocer y hacer)
- f) Aplica en forma correcta los interruptores en la elaboración de los diagramas de flujo (saber hacer).
- g) Aplica la lógica de programación adecuada en la resolución de problemas mediante el diagrama de flujo y pseudocódigo. (saber hacer)
- h) Interpreta diagramas de flujo construidos para solucionar problemas específicos sin margen de error. (saber conocer y hacer)
- i) Define con claridad el funcionamiento de cada una de las estructuras de selección y repetición. (saber conocer)
- j) Utiliza las estructuras condicionales y de repetición en una forma correcta. (saber conocer)
- k) Realiza con precisión ejercicios algorítmicos y diagramas de flujo dando solución a un problema del contexto. (saber hacer)
- l) Establece diferencias entre contadores y acumuladores. (saber conocer).
- m) Aplica correctamente la utilización de contadores y acumuladores al elaborar el diagrama de flujo y pseudocódigo. (saber ser)
- n) Aplica la prueba de escritorio a diagramas de flujo y pseudocódigos de forma correcta (saber hacer)
- o) Demuestra puntualidad en la entrega de sus trabajos. (saber ser)

Niveles de dominio:

Inicial-Receptivo:No tiene dominio del criterio de desempeño y una o ninguna evidencia concreta. Manifiesta escaso dominio sobre conceptos básicos para la realización de algoritmos, no aplica técnicas ni estrategias al diseñar diagramas de flujo y pseudocódigos.

Básico:Muestra parcialmente el dominio del criterio de desempeño y hasta dos evidencias, con insuficiencias. Demuestra pobre dominio de conceptos para la realización de algoritmos, aplica parcialmente técnicas y estrategias al diseñar diagramas de flujo y pseudocódigos.

Autónomo:Muestra buen dominio del criterio de desempeño y hasta cuatro evidencias, dos con insuficiencias. Demuestra dominio parcial de conceptos para la realización de algoritmos, aplica parcialmente técnicas y estrategias al diseñar diagramas de flujo y pseudocódigos.

Estratégico:Muestra excelente dominio del criterio de desempeño y evidencias. Demuestra creatividad e innovación, así como estrategias propias a partir de la realidad con amplio dominio de conceptos para la realización de algoritmos, aplica técnicas y herramientas en la resolución de problemas.

Ponderación del criterio: 25 puntos

Criterio de desempeño o Resultado de Aprendizaje

1.2. Identifica generalidades del lenguaje de programación estructurada C/C++ y aplica instrucciones y tipos de estructuras fundamentales en la elaboración de un programa.

Evidencias:

- a) Conoce la historia del lenguaje de programación estructurado C/C++. (saber conocer)
- b) Define la estructura de un programa en C/C++. (saber conocer)
- c) Establece diferencias entre las diferentes directivas del preprocesador
- d) Conoce y aplica las palabras reservadas que no pueden ser utilizadas como identificadores al desarrollar un programa en C/C++. (saber conocer y hacer)
- e) Identifica y selecciona los tipos de datos utilizados en C/C++. (saber conocer y hacer)
- f) Diseña programas utilizando operaciones de manejo de entrada/salida. (saber hacer)

<p>g) Establece diferencias entre las funciones de entradas y salidas con y sin formato. (saber conocer)</p> <p>h) Utiliza modificadores para indicar el tipo de dato que se trata en la entrada y salida de datos en el desarrollo de un programa en lenguaje estructurado. (saber hacer)</p> <p>i) Aplica secuencias de escape para salidas a pantalla, a un archivo en disco o a la impresora. (saber hacer).</p> <p>j) Resuelve problemas utilizando los elementos que intervienen en el desarrollo de un programa. (saber hacer).</p> <p>k) Identifica y selecciona la sentencia de control adecuada al desarrollar un programa en lenguaje estructurado. (saber conocer y hacer)</p> <p>l) Utiliza operadores aritméticos, lógicos y relacionales en forma adecuada en el desarrollo del programa. (saber hacer)</p> <p>m) Desarrolla programas que utilizan bibliotecas de funciones de entrada y salida, de manejo de archivos y manipulación de cadenas. (saber hacer).</p> <p>n) Resuelve problemas utilizando estructuras repetitivas en lenguaje de programación estructurado. (saber hacer)</p> <p>o) Demuestra puntualidad en la entrega de sus trabajos. (saber ser)</p> <p>p) Tiene sentido de reto al desarrollar programas correctamente considerando los requerimientos del contexto. (saber ser).</p>
<p>Niveles de dominio</p> <p>Inicial-Receptivo: No tiene dominio del criterio de desempeño y una o ninguna evidencia concreta. Manifiesta escaso dominio sobre conceptos básicos del lenguaje de programación estructurada C/C++, no aplica instrucciones ni estructuras en la elaboración de un programa.</p> <p>Básico: Muestra parcialmente el dominio del criterio de desempeño y hasta dos evidencias concretas, con alguna insuficiencia. Demuestra dominio parcial de conceptos en el lenguaje de programación estructurada C/C++, aplica parcialmente instrucciones y tipos de estructuras en la elaboración de un programa.</p> <p>Autónomo: Muestra buen dominio del criterio de desempeño y cuatro evidencias, dos con insuficiencias. Demuestra autonomía con dominio de conceptos para la realización de programas en lenguaje C++, aplica instrucciones, tipos de estructuras y herramientas en la elaboración de un programa</p> <p>Estratégico: Muestra excelente dominio del criterio de desempeño y evidencias. Demuestra creatividad e innovación, así como estrategias propias a partir de la realidad con amplio dominio de conceptos para la realización de programas en C/C+, aplica técnicas y herramientas en la resolución de problemas.</p>
<p>Ponderación del Criterio: 20 puntos</p>
<p>Criterio de desempeño o Resultado de Aprendizaje</p>
<p>1.3. Desarrolla programas más complejos utilizando un método de diseño descendente con funciones</p>
<p>EVIDENCIAS</p>
<p>a) Define la necesidad de utilizar funciones al momento de codificar un programa en C/C++. (saber conocer)</p> <p>b) Define los conceptos relacionados con el uso de funciones. (saber conocer)</p> <p>c) Codifica programas utilizando funciones como parte de la solución de problemas específicos. (saber hacer)</p> <p>d) Enumera con precisión las características y usos de las funciones en lenguaje de programación estructurado. (saber conocer)</p> <p>e) Identifica y aplica las reglas y normas para la invocación, uso de variables y parámetros en las funciones al desarrollar un programa en lenguaje estructurado. (saber conocer y hacer)</p> <p>f) Ejemplifica el método para la creación de funciones recursivas. (saber conocer)</p> <p>g) Resuelve problemas utilizando funciones recursivas de forma correcta (saber hacer)</p> <p>h) Produce programas específicos utilizando funciones con precisión en un lenguaje de programación estructurado. (saber hacer)</p> <p>i) Participa interdisciplinariamente en los proyectos de solución de problemas computacionales. (saber ser).</p> <p>j) Demuestra responsabilidad y honestidad ante las tareas asignadas. (saber ser)</p>
<p>Niveles de dominio</p>

<p>Inicial-Receptivo:No tiene dominio del criterio de desempeño y una o ninguna evidencia concreta. Manifiesta escaso dominio sobre conceptos básicos de programación con funciones, no aplica técnicas ni herramientas en la resolución de problemas.</p> <p>Básico:Muestra parcialmente dominio del criterio de desempeño y hasta dos evidencias concretas con insuficiencias. Demuestra dominio parcial de conceptos básicos de la programación con funciones, presenta baja aplicación de técnicas y estrategias en la resolución de problemas.</p> <p>Autónomo: Muestra buen dominio del criterio de desempeño y cuatro evidencias, dos con insuficiencias. Demuestra autonomía con dominio de conceptos para la realización de programas utilizando funciones en programación estructurada, aplica técnicas y herramientas en la resolución de problemas.</p> <p>Estratégico:Muestra excelente dominio del criterio de desempeño y evidencias. Demuestra creatividad e innovación, así como estrategias propias a partir de la realidad con amplio dominio de conceptos para la realización de programas utilizando funciones en lenguaje de programación estructurado, aplica técnicas y herramientas en la resolución de problemas.</p>
Ponderación del Criterio: 27 puntos
Criterio de desempeño o Resultado de Aprendizaje
1.4. Crea y manipula tipos de datos estructurados mediante la utilización de arreglos.
Evidencias
<p>a) Diferencia y define los tipos de datos estructurados: unidimensional, bidimensional y multidimensional utilizados en lenguaje de programación C/C++. (saber conocer y hacer)</p> <p>b) Comprende el uso de arreglos: declara, inicializa y referencia a los elementos de un arreglo de una manera precisa. (saber conocer, hacer).</p> <p>c) Invoca funciones con arreglos en forma adecuada. (saber conocer)</p> <p>d) Utiliza arreglos unidimensionales y bidimensionales para el almacenamiento de datos en la solución de problemas en forma precisa. (saber hacer)</p> <p>e) Produce programas específicos utilizando arreglos en forma adecuada en un lenguaje de programación estructurado. (saber hacer)</p> <p>f) Participa interdisciplinariamente en los proyectos de solución de problemas computacionales. (saber ser).</p> <p>g) Demuestra responsabilidad y honestidad ante las tareas asignadas. (saber ser)</p>
Niveles de dominio
<p>Inicial-Receptivo:No tiene dominio del criterio de desempeño y una o ninguna evidencia concreta. Manifiesta escaso dominio sobre conceptos referentes a los arreglos en C/C++, no aplica técnicas ni estrategias en la resolución de problemas utilizando arreglos.</p> <p>Básico: Muestra parcialmente el dominio del criterio de desempeño y hasta dos evidencias, con insuficiencias. Demuestra dominio parcial de conceptos referentes a la creación de datos estructurados en C/C++, aplica parcialmente técnicas y estrategias en la resolución de problemas utilizando arreglos.</p> <p>Autónomo:Muestra buen dominio del criterio y cuatro evidencias concretas de su desempeño, dos con insuficiencias. Demuestra autonomía con dominio de conceptos referentes a la creación y manipulación de los datos estructurados, aplica técnicas y herramientas en la resolución de problemas utilizando arreglos.</p> <p>Estratégico:Muestra excelente dominio del criterio de desempeño y evidencias. Demuestra creatividad e innovación, así como estrategias propias a partir de la realidad con amplio dominio de conceptos para la realización de programas utilizando arreglos en C/C+, aplica técnicas y herramientas en la resolución de problemas.</p>
Ponderación del Criterio: 28 puntos

Las evidencias se evalúan con base en los criterios o resultados de aprendizaje, se valoran de forma integral y no de manera individual, lo que significa que cada evidencia es valorada considerando las demás evidencias y no de forma separada.

Para precisar el nivel de desarrollo de la competencia se asignan un puntaje a cada uno de los criterios o resultados de aprendizaje que se toman en cuenta en la evaluación de la competencia, a partir de comparar cada criterio entre sí y en cada criterio o resultado de aprendizaje comparar las evidencias para determinar su grado de relevancia en la competencia.

Para determinar la puntuación a alcanzar en cada criterio o resultado de aprendizaje se ha determinado la siguiente escala en función del nivel de dominio que se demuestre: *Inicial- Receptivo*: 50 % del puntaje del criterio, *Básico*: 70 % del puntaje del criterio, *Autónomo*: 80 % del puntaje del criterio, *Estratégico*: 100 % del puntaje del criterio. El puntaje total obtenido a partir de la suma de los puntajes alcanzados en cada criterio se tomará en cuenta para valorar el nivel de dominio de la Competencia en este eje procesual, evaluando la siguiente escala:

Nivel Inicial – Receptivo: Valor obtenido entre 0 y 60

Nivel Básico: Valor obtenido entre 61 y 85

Nivel Autónomo: Valor Obtenido entre 86 y 95

Nivel Estratégico: Valor Obtenido entre 96 y 100

Conclusiones

La formación y desarrollo de la competencia programar software en los estudiantes de la carrera de Ingeniería en Sistemas Informáticos es un proceso que debe transitar por alcanzar los resultados de aprendizaje previstos en las distintas asignaturas que conforman los cuatro grandes ejes procesuales de la competencia.

La evaluación de los criterios de desempeño o resultados de aprendizaje requiere la definición previa de las evidencias que dan cuenta, con pruebas concretas y tangibles, de la formación y desarrollo de la competencia. Este proceso supone la definición de los niveles de dominio y los puntajes asignados a cada nivel.

Recibido: noviembre 2014

Aprobado: julio 2015

Bibliografía

Abambari, M. (2015). Valoración epistemológica y didáctica de la evaluación de competencias profesionales. *Transformación*, 11(1), 16-27. Disponible en <http://transformacion.reduc.edu.cu/index.php/transformacion/art>.

Adam, S. (2004). Using Learning Outcomes: A consideration of the nature, role, application and implications for European education of employing learning outcomes at the local, national and international levels. *United Kingdom Bologna Seminar*. Herriot-Watt University. Disponible en <http://www.gov.scot/resource/doc/25725/0028779.pdf>.

Allan, J. (1996). Learning outcomes in Higher Education. *Studies in Higher Education*, 21(10), 93-108.

Disponible en:

<http://reforma.fen.uchile.cl/Papers/Learning%20outcomes%20in%20higher%20education%20-%20Allan.pdf>.

Anderson, L., & Krathwohl, D. (Edits.). (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's taxonomy of educational objectives*. New York: Longman.

Association for Computing Machinery; Institute for Electrical and Electronic Engineers IEEE. (2001).

Computing Curricula 2001. Recuperado el 3 de mayo de 2014, de www.computer.org/education/cc2001

Bingham, J. (1999). *Guide to developing learning outcomes. The Learning and Teaching Institute Sheffield Hallam University*. Sheffield: Sheffield Hallam University.

Bloom, B. (1956). *Taxonomy of educational objectives, Book 1. Cognitive domain*. New York: Longman Publishing.

Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior. (2011).

Modelo general para la evaluación de carreras con fines de acreditación. Quito: Disponible en: <http://www.uta.edu>.

Contreras, J. (2012). *Tuning América Latina: Social and educative innovation*. Recuperado el 13 de enero de 2013, de http://www.tuningal.org/es/publicaciones/cat_view/44-presentaciones-reunion-bruselas-19-22-nov-2012

Council of Europe. (2002). *Seminar on Recognition Issues in the Bologna Process*. Recuperado el 20 de enero de 2012, de <http://www.coe.int>

Dalerba, A., Karpow, N., & Fissore, M. (2014). *Problemas de Programación: Algunos nexos posibles entre la enseñanza y el aprendizaje*. Recuperado el 5 de agosto de 2014, de http://sedici.unlp.edu.ar/bitstream/handle/10915/38361/Documento_completo.pdf?sequence=1

European Credit Transfer and Accumulation System. (2005). *ECTS Users' Guide Brussels: Directorate-General for Education and Culture*. Recuperado el 13 de octubre de 2013, de http://ec.europa.eu/education/programmes/socrates/ects/doc/guide_en.pdf

(s.f.). Ferris, T. & Aziz, S. (2005). *A psychomotor skills extension to Bloom's Taxonomy of Education Objectives for engineering education. Exploring Innovation in Education and Research*.

Ferris, T., & Aziz, S. (2005). A psychomotor skills extension to Bloom's Taxonomy of Education Objectives for engineering education. *Innovation in Education and Research*, 21(1), 7-14. <http://www.researchgate.net/publication/228372464>.

- Fry, H., Ketteridge, S., & Marshall, A. (2000). *A Handbook for Teaching and Learning in Higher Education*. London: Kogan Page Limited.
- Gonzalez, M., & Ramírez, I. (2011). La formación de competencias profesionales: un reto en los proyectos curriculares universitarios. *Odiseo*, 8(16), <http://www.odiseo.com.mx/2011/8-16/gonzalez-ramirez-formacion-competencias.html>.
- Gosling, D., & Moon, J. (2001). *How to use learning outcomes and assessment criteria*. London: Oficina de la Sociedad Española de Educación Comparada. Disponible: <http://wcces.com/SEEC-Bulletin-1-2012.pdf>.
- Jenkins, A., & Unwin, D. (2001). *How to write learning outcomes*. Recuperado el 13 de enero de 2013, de www.ncgia.ucsb.edu/education/curricula/giscc/units/format/outcomes.html
- Kennedy, D. (2007). *Redactar y utilizar resultados de aprendizaje*. Recuperado el 23 de febrero de 2013, de http://www.uctemuco.cl/cedid/archivos/apoyo/new_resultados_de_aprendizaje_01_dkennedy.pdf.
- Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into Practice*, 41(4), Disponible: http://www.unco.edu/cetl/sir/stating_outcome/documents/Krathwohl.pdf.
- Marco europeo de cualificaciones para el aprendizaje permanente. (2011). *Using Learning outcomes*. Recuperado el 13 de enero de 2013, de http://ec.europa.eu/education/lifelong-learning-policy/doc/eqf/note4_en.pdf.
- Montes, N., Machado, E., & González, C. (2011). Estrategia didáctica para el desarrollo de la competencia gestión del conocimiento matemático en estudiantes universitarios. *Transformación*, 7(2), 1-8. Disponible en <http://transformacion.reduc.edu.cu/index.php/transformacion/article/view/48>.
- Moon, J. (2002). *The Module and Programme Development Handbook*. London: Kogan Page Limited.
- Proyecto Tuning América Latina. (2007). *Reflexiones y perspectivas de la Educación Superior en América Latina, Informe Final*. . Recuperado el 13 de enero de 2013, de Disponible: <http://www.tuning.unideusto.org/tuningal>.
- Tobón, S. (2005). *Formación basada en competencias: Pensamiento complejo, diseño curricular y didáctica* (2da ed.). Bogota: Ecoe Ediciones.
- Tobón, S. (2008). *Vacios, problemas y nuevas perspectivas en el proceso de evaluación y certificación de competencias: propuesta metodológica. Conferencia Magistral* . Recuperado el 13 de marzo de 2014, de <http://ojvf.files.wordpress.com/2008/10>
- Tobón, S., Pimienta, J., & García Fraile, J. A. (2010). *Secuencias didácticas: aprendizaje y evaluación de competencias. 1era. Edición*, (1ra ed.). Ciudad México: Pearson Educación.

Villardón, L. (2006). *Evaluación del aprendizaje para promover el desarrollo de las competencias*. Vol. 24. Recuperado el 18 de mayo de 2014, de <http://revistas.um.es/educatio/article/view/153/136>

Yániz, C., & Villardón, L. (2006). Planificar desde competencias para promover el aprendizaje. El reto de la sociedad del conocimiento para el profesorado universitario. En *Cuadernos monográficos del ICE. Número 12*. Bilbao.